

PeopleTools 8.4: PeopleCode Developer's Guide

PeopleSoft®

PeopleTools 8.4: PeopleCode Developer's Guide

SKU Tr84PCD-B 0302

PeopleBooks Contributors: Teams from PeopleSoft Product Documentation and Development.

Copyright © 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, the PeopleSoft logo, PeopleTools, PS/nVision, PeopleCode, PeopleBooks, *PeopleTalk*, and Vantive are registered trademarks, and "People power the internet." and Pure Internet Architecture 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.

Contents

PeopleCode Developer's Guide Preface

About This PeopleBook.....	xiii
Before You Begin.....	xiii
<i>PeopleSoft Application Fundamentals</i>	xiv
Related Documentation	xiv
Hard-copy Documentation.....	xv
PeopleBooks Standard Field Definitions.....	xv
Typographical Conventions and Visual Cues.....	xvii
Page and Panel Introductory Table.....	xviii
Comments and Suggestions.....	xviii

Chapter 1

Introducing What's New

Understanding What's New.....	1-1
New Classes.....	1-1
Writing Your Own Classes.....	1-2
Considerations Using Application Classes	1-2
Using Binary Sorts	1-3
Additional Changes	1-4
PeopleCode Debugger	1-7
Database Changes in the FieldChange Event	1-7

Chapter 2

Understanding PeopleCode and Events

Understanding PeopleCode Programs and Events	2-1
Accessing PeopleCode in Application Designer	2-2
Using Record Field PeopleCode.....	2-4
Accessing Record Field PeopleCode From a Record Definition.....	2-4
Accessing Record Field PeopleCode From a Page Definition.....	2-5
Distinguishing Between Fields and Record Fields	2-7
Record Field Event Set	2-7
Using Component Record Field PeopleCode	2-8
Component Record Field Event Set.....	2-9
Using Component Record PeopleCode	2-9

Component Record Event Set	2-10
Using Component PeopleCode	2-11
Component Event Set.....	2-12
Using Page PeopleCode.....	2-12
Page Activate Event	2-13
Using Menu Item PeopleCode.....	2-13
Menu Item ItemSelected Event	2-15
Using Message PeopleCode.....	2-15
Accessing Message PeopleCode	2-15
Message Event Sets.....	2-17
Storing and Saving PeopleCode Programs	2-17
Automatic Backup of PeopleCode	2-18
Copying PeopleCode With a Parent Definition.....	2-18
Upgrading PeopleCode Programs.....	2-19

Chapter 3

Using the PeopleCode Editor

Navigating Between PeopleCode Programs	3-1
Understanding the PeopleCode Editor Window	3-1
Using the Drop-down Definition List	3-2
Using the Drop-down Event List	3-6
Using the PeopleCode Editor.....	3-7
Editing Functions	3-8
Auto Formatting	3-10
Drag-and-Drop Editing	3-11
Accessing PeopleCode External Functions.....	3-11
Accessing Definitions and Associated PeopleCode	3-12
Context-Sensitive Help	3-12
Selecting a Font for the PeopleCode Editor	3-13
Generating PeopleCode Using Drag-and-Drop	3-13
Generating Definition References	3-14
Generating PeopleCode for a Business Interlink	3-14
Generating PeopleCode for a Component Interface.....	3-15
Generating PeopleCode for a File Layout.....	3-17

Chapter 4

Introducing the SQL Editor

Understanding the SQL Editor Window.....	4-1
Understanding SQL Properties	4-2
Accessing the SQL Editor.....	4-3

SQL Definitions	4-3
Dynamic View or SQL View Records.....	4-4
Application Engine Programs	4-5
Using the SQL Editor	4-7

Chapter 5

Introducing Application Packages

Understanding Application Packages.....	5-1
Using the Application Class Editor	5-3
Considerations Editing Application Classes	5-4
Editing Application Package Classes	5-5

Chapter 6

PeopleCode Language

Understanding the PeopleCode Language.....	6-1
Data Types.....	6-1
Conventional Data Types.....	6-2
Object Data Types.....	6-2
Comments.....	6-5
Statements.....	6-6
Separators.....	6-6
Assignment Statements	6-6
Language Constructs.....	6-7
Control Statements.....	6-8
Conditional Loops.....	6-10
Functions	6-11
Defining Functions.....	6-12
Declaring Functions	6-12
Calling Functions	6-12
Function Return Values	6-13
Avoiding Function Naming Conflicts.....	6-13
Expressions.....	6-14
Constants.....	6-15
Functions as Expressions	6-16
Variables	6-16
System Variables.....	6-24
Metastrings.....	6-24
Record Field References	6-25
Definition Name References	6-26
Operators	6-28

Math Operators.....	6-28
Operations on Dates and Times	6-29
String Concatenation.....	6-29
@ Operator.....	6-30
Comparison Operators.....	6-30
Boolean Operators.....	6-31

Chapter 7

Understanding Objects and Classes in PeopleCode

Understanding Classes and Objects	7-1
What is a Class?	7-1
What is an Object?	7-2
What is Instantiating Objects?.....	7-2
Working With Objects	7-2
Object Properties	7-3
Object Methods	7-4
Assigning Objects.....	7-5
Passing Objects.....	7-7

Chapter 8

Using Methods and Built-In Functions

Working With Restrictions on Method and Function Use.....	8-1
Think-Time Functions.....	8-2
WinMessage and MessageBox.....	8-3
Program Execution With Fields not in the Data Buffer	8-5
Errors and Warnings.....	8-5
DoSave	8-5
Record Object Database Methods	8-6
SQL Object Methods and Functions	8-6
Component Interface Restricted Functions	8-7
SearchInit PeopleCode Function Restrictions	8-7
CallAppEngine.....	8-7
ReturnToServer	8-8
GetPage	8-8
GetGrid.....	8-8
Publish Method	8-9
SyncRequest.....	8-9
Implementing Modal Transfers	8-9
Considerations Before Implementing a Modal Transfer	8-10
Implementing the Multi-Row Insert Feature	8-12

Additional Multi-Row Insert Considerations	8-13
Using the ImageReference Field	8-13
Using PeopleCode With PeopleSoft Internet Architecture	8-14
Using Internet Scripts.....	8-14
Using the Field Object Style Property	8-15
Using the HTML Area	8-16
Using HTML Definitions and the GetHTMLText Function.....	8-17
Using HTML Definitions and the GetJavaScriptURL Method	8-18
Inserting Rows Using PeopleCode	8-18
Using the GenerateTree Function.....	8-19
Building Your HTML Tree Page	8-21
The HTML Tree Rowset Records.....	8-22
Using HTML Tree End-User Actions (Events)	8-24
Customizing PeopleCode for the HTML Tree (Part One)	8-25
PostBuild PeopleCode Example	8-27
Customizing PeopleCode for the HTML Tree (Part Two)	8-29
FieldChange PeopleCode Example.....	8-30
Adding Mouse-Over Ability to the HTML Tree.....	8-34
Adding Visual Selection Node Indicators.....	8-35
Understanding File Attachment Architecture.....	8-36
Debugging File Attachment Problems	8-37
Configuring Multiple Application Servers to Support File Attachments	8-39
Viewing File Attachments	8-39
Chunking Considerations.....	8-41
Using Microsoft Windows NT 4 ftp Server.....	8-42
Using Non-DNS URLs With Unix	8-43
Using the Attachment Functions.....	8-43
Coding Considerations Using the Attachment Functions	8-45
Naming Considerations for URLs and File Attachments.....	8-46
User Filenames for PutAttachment	8-46
Using OLE Functions.....	8-47
Data Types	8-48
Sharing a Single Object Instance	8-48
OLE vs. WinExec	8-48
Using the Select and SelectNew Methods.....	8-49
What Select Does.....	8-50
Select Syntax.....	8-50
Using Standalone Rowsets	8-53
The Fill Method	8-53
The CopyTo Method.....	8-54

Adding Child Rowset	8-54
Using Standalone Rowsets to Write a File	8-55
Using Standalone Rowsets to Read a File.....	8-57
Using Errors and Warnings.....	8-59
Syntax of Errors and Warnings	8-59
Errors, Warnings, and Edits	8-59
Errors and Warnings in FieldEdit.....	8-60
Errors and Warnings in SaveEdit	8-60
Errors and Warnings in RowSelect	8-61
Errors and Warnings in RowDelete.....	8-61
Errors and Warnings in Other Events.....	8-61
Using RemoteCall.....	8-62
RemoteCall Components.....	8-63
RemoteCall and PeopleSoft Process Scheduler	8-65
RemoteCall Modifications for PeopleSoft Process Scheduler	8-65
RemoteCall Programming Guidelines.....	8-66

Chapter 9

Referencing Data in the Component Buffer

Understanding Component Buffer Structure and Contents	9-1
Comparing Rowsets to Scrolls	9-3
What Record Fields Are in the Component Buffer?	9-3
Specifying Data With Contextual References	9-4
Understanding Current Context.....	9-5
Contextual Reference Processing Order.....	9-6
Contextual Row References	9-6
Contextual Buffer Field References	9-7
Specifying Data With References Using Scroll Path Syntax and Dot Notation	9-9
Scroll Path Syntax in PeopleTools 7.5	9-10
Scroll Level, Row, and Buffer Field References.....	9-12
Scroll Path Syntax Before PeopleTools 7.5	9-18

Chapter 10

Data Buffer Access

Understanding Data Buffer Access.....	10-1
Access Classes	10-1
Data Buffer Model and Data Access Objects.....	10-2
Understanding Data Buffer Classes Examples	10-3
Object Creation Examples.....	10-6
Traversing the Data Buffer Hierarchy Example.....	10-14

Traversing a Rowset Example	10-17
Using a Hidden Work Scroll Example.....	10-19
Understanding Current Context.....	10-21
Creating Records or Rowsets and Current Context	10-22
Accessing Secondary Component Buffer Data	10-22
Instantiating Rowsets Using Non-Component Buffer Data	10-23

Chapter 11

PeopleCode and the Component Processor

Understanding the Component Processor.....	11-1
Events Outside the Component Processor Flow.....	11-1
How PeopleCode Programs Are Triggered	11-2
Accessing PeopleCode Programs.....	11-3
Understanding Execution Order of Events and PeopleCode	11-4
Component Processor Behavior	11-7
From Page Start to Page Display	11-8
End-User Actions in the Component	11-9
Processing Sequences.....	11-10
Default Processing	11-11
Search Processing in Update Modes.....	11-13
Search Processing in Add Modesx.....	11-16
Component Build Processing in Update Modes	11-18
Row Select Processing.....	11-20
Component Build Processing in Add Modes.....	11-21
Field Modification.....	11-22
Row Insert Processing.....	11-24
Row Delete Processing	11-25
Pushbuttons.....	11-27
Prompts	11-27
Pop-Up Menu Display	11-27
ItemSelected Processing	11-28
Save Processing.....	11-28
Exit Component	11-30
PeopleSoft Internet Architecture Processing Considerations	11-30
Deferred Processing Mode	11-30
PeopleCode Events.....	11-33
Activate Event.....	11-34
FieldChange Event.....	11-34
FieldDefault Event.....	11-35
FieldEdit Event	11-36

FieldFormula Event.....	11-36
ItemSelected Event.....	11-37
PostBuild Event.....	11-37
PreBuild Event	11-37
PrePopup Event.....	11-38
RowDelete Event.....	11-38
RowInit Event	11-39
RowInsert Event.....	11-40
RowSelect Event	11-41
SaveEdit Event.....	11-42
SavePostChange Event.....	11-43
SavePreChange Event	11-43
SearchInit Event.....	11-44
SearchSave Event.....	11-45
Workflow Event.....	11-45
PeopleCode Execution in Multiple Scroll Pages	11-46

Chapter 12

PeopleCode and PeopleSoft Internet Architecture

Using PeopleCode in the PeopleSoft Internet Architecture.....	12-1
Using PeopleCode to Populate Search Dialog Key Fields.....	12-2
Calling Dynamic Link Library Functions on the Application Server.....	12-2
Sample Cross-Platform External Test Function.....	12-3
Updating the Installation and PSOPTIONS Tables	12-4

Chapter 13

Debugging Your Application

Understanding the PeopleCode Debugger	13-1
Accessing the PeopleCode Debugger	13-1
Understanding the PeopleCode Debugger Features	13-2
Visible Current Line of Execution	13-2
Visible Breakpoints.....	13-3
Hover Inspect	13-4
Single Debugger.....	13-5
Variables Panes	13-6
General Debugging Tips	13-9
PeopleCode Debugger Options.....	13-10
Setting Up the Debugging Environment.....	13-14
Debugging Subscription PeopleCode	13-14
Compiling All PeopleCode Programs.....	13-15

Setting PeopleCode Debugger Log Options..... 13-16

Interpreting the PeopleCode Debugger Log File..... 13-18

 Log File Contents..... 13-19

 Other Items in the Log File..... 13-19

Find In 13-20

Cross Reference Reports 13-23

Glossary

Index

PeopleCode Developer's Guide Preface

This PeopleBook covers the concepts of PeopleCode, the proprietary language used in the development of PeopleSoft applications. Its chapters describe techniques for adding PeopleCode to applications, tips for using PeopleCode, the interaction of PeopleCode and the Component Processor, and a number of other specialized topics, such as the use of the PeopleCode debugger and referencing data in the component buffer.

The accompanying book, *PeopleCode Reference*, is a complete reference of the PeopleCode language. Its chapters describe the syntax and fundamental elements of the PeopleCode language.

As a licensee of PeopleTools, you are licensed to use the base portal technology, which is limited to navigation to licensed PeopleSoft applications. If you want to register additional non-PeopleSoft content, customize your homepage, or create any pagelets, you must license PeopleSoft Enterprise Portal.

The “About This PeopleBook” section contains general product line information, such as related documentation, common page elements, and typographical conventions. This book also contains a glossary with useful terms that are used in PeopleBooks.

See **PeopleSoft Glossary**.

See also *PeopleCode Reference*

About This PeopleBook

This book provides you with the information that you need for implementing and using *PeopleTools 8.4* applications. Complete documentation for this release is provided on the CD-ROM PT84PBR0.

Note. Your access to PeopleSoft PeopleBooks depends on which PeopleSoft applications you've licensed. You may not have access to all of the PeopleBooks.

This section contains information that you should know before you begin working with PeopleSoft products and documentation, including PeopleSoft-specific documentation conventions, information specific to each PeopleSoft product line, and information on ordering additional copies of our documentation.

Before You Begin

To benefit fully from the information covered in this book, you should have a basic understanding of how to use PeopleSoft applications. We recommend that you 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.

Because we assume that you already know how to navigate the PeopleSoft system, much of the information in these books is not procedural. That is, these books do not typically provide step-by-step instructions on using tables, pages, and menus. Instead, we provide you with the information that you need to use the system most effectively and to implement your PeopleSoft application according to your organizational or departmental needs. PeopleBooks expand on the material covered in PeopleSoft training classes.

PeopleSoft Application Fundamentals

Each PeopleSoft application PeopleBook provides implementation and processing information for your PeopleSoft database. However, there is additional, essential information describing the setup and design of your database contained in a companion volume of documentation called *PeopleSoft Application Fundamentals*.

PeopleSoft Application Fundamentals contains important topics that apply to many or all PeopleSoft applications across each product line. Whether you are implementing only one PeopleSoft application, some combination of products within a product line, or an entire PeopleSoft system, you should be familiar with the contents of this central PeopleBook. It contains fundamental information such as setting up control tables and administering security.

The PeopleSoft Applications Fundamentals PeopleBook contains common information pertinent to all applications in each product line, such as defining general options. If you're upgrading from a previous PeopleSoft release, you may notice that we've removed some topics or topic headings from the individual application PeopleBooks and consolidated them in this single reference book. You'll now find only application-specific information in your individual application PeopleBooks. This makes the documentation as a whole less redundant. Throughout each PeopleBook, we provide cross-references to *PeopleSoft Application Fundamentals* and other PeopleBooks.

Related Documentation

You can order printed, bound versions of the complete PeopleSoft documentation delivered on your PeopleBooks CD-ROM and additional copies of the PeopleBooks CDs through the Documentation section of the PeopleSoft Customer Connection website:
<http://www.peoplesoft.com/corp/en/login.asp>

You can find updates and additional documentation for this release, as well as previous releases, on PeopleSoft Customer Connection (<http://www.peoplesoft.com/corp/en/login.asp>). Through the Documentation section of 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 delivered on your PeopleBooks CD.

Important! Before you upgrade, it is *imperative* that you check PeopleSoft Customer Connection for updates to the upgrade instructions. We continually post updates as we refine the upgrade process.

Hard-copy Documentation

To order printed, bound volumes of the complete PeopleSoft documentation delivered on your PeopleBooks CD-ROM, visit the PeopleSoft Press website from the Documentation section of PeopleSoft Customer Connection. The PeopleSoft Press website is a joint venture between PeopleSoft and Consolidated Publications Incorporated (CPI), our book print vendor.

We make 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 the following methods:

Internet	From the main PeopleSoft Internet site, go to the Documentation section of Customer Connection. You can find order information under the Ordering PeopleBooks topic. Use a Customer Connection ID, credit card, or purchase order to place your order. PeopleSoft Internet site: http://www.peoplesoft.com/ .
Telephone	Contact Consolidated Publishing Incorporated (CPI) at 800 888 3559 .
Email	Send email to CPI at callcenter@conpub.com .

PeopleBooks Standard Field Definitions

Throughout our product documentation, you will encounter fields and buttons that are used on many application pages or panels. This section lists the most common fields and buttons and provides standard definitions.

Field	Definition
As of Date	The last date for which a report or process includes data.
Business Unit	An identification code 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	Freeflow text up to 30 characters.

Field	Definition
Effective Date	<p>Date on which a table row becomes effective; the date that an action begins. For example, if you want 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.</p> <hr/> <p>For more information about effective dates, see <u>Understanding Effective Dates in Using PeopleSoft Applications.</u></p> <hr/>
EmplID (employee ID)	Unique identification code for an individual associated with your organization.
Language or Language Code	<p>The language in which you want the field labels and report headings of your reports to print. The field values appear as you enter them.</p> <p>Language also refers to the language spoken by an employee, applicant, or non-employee.</p>
Process Frequency group box	<p>Designates the appropriate frequency in the Process Frequency group box:</p> <p>Once executes the request the next time the batch process runs. After the batch process runs, the process frequency is automatically set to Don't Run.</p> <p>Always executes the request every time the batch process runs.</p> <p>Don't Run ignores the request when the batch process runs.</p>
Report ID	The report identifier.
Report Manager	This button takes you to 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	This button takes you to the Process List page, where you can view the status of submitted process requests.
Run	This button takes you to the Process Scheduler request page, where you can specify the location where a process or job runs and the process output format.
	<hr/> <p>For more information about the Report List page, the Process List page, and the Process Scheduler, see <u>Process Scheduler Basics in the PeopleTools documentation.</u></p> <hr/>
Request ID	A request identification that represents a set of selection criteria for a report or process.

Field	Definition
User ID	The system identifier for the individual who generates a transaction.
SetID	An identification code that represents a set of control table information or TableSets. A TableSet is a group of tables (records) necessary to define your company's structure and processing options.
Short Description	Freeflow text up to 15 characters.

Typographical Conventions and Visual Cues

We use a number of standard conventions and visual cues in our online documentation.

The following list contains our typographical conventions and visual cues:

`(monospace font)`

Indicates a PeopleCode program or other program example.

Bold

Indicates field names and other page elements, such as buttons and group box labels, when these elements are documented below the page on which they appear. When we refer to these elements elsewhere in the documentation, we set them in Normal style (not in bold).

We also use boldface when we refer to navigational paths, menu names, or process actions (such as **Save** and **Run**).

Italics

Indicates a PeopleSoft or other book-length publication. We also use italics for *emphasis* and to indicate specific field values. When we cite a field value under the page on which it appears, we use this style: ***field value***.

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.

Cross-references

The phrase **For more information** indicates where you can find additional documentation on the topic at hand. We include the navigational path to the referenced topic, separated by colons (:). Capitalized titles in *italics* indicate the title of a PeopleBook; capitalized titles in normal font refer to sections and specific topics within the PeopleBook. Here's an example:

For more information, see **Documentation on CD-ROM** in *About These PeopleBooks*: Additional

Resources.

Note. Text in this bar indicates information that you should pay particular attention to as you work with your PeopleSoft system. If the note is preceded by **Important!**, the note is crucial and includes information that concerns what you need to do for the system to function properly.

Text in this bar indicates cross-references to related or additional information.

Warning! Text within this bar indicates a crucial configuration consideration. Pay very close attention to these warning messages.

Page and Panel Introductory Table

In the documentation, each page or panel description in the application includes an introductory table with pertinent information about the page. Not all of the information will be available for all pages or panels.

Usage	Describes how you would use the page or process.
Object Name	Gives the system name of the panel or process as specified in the PeopleTools Application Designer. For example, the Object Name of the Detail Calendar panel is <code>DETAIL_CALENDAR1</code> .
Navigation	Provides the path for accessing the page or process.
Prerequisites	Specifies which objects must have been defined before you use the page or process.
Access Requirements	Specifies the keys and other information necessary to access the page. For example, SetID and Calendar ID are required to open the Detail Calendar page.

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 our documentation, 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 comments by email to the authors of the PeopleSoft documentation at:

DOC@PEOPLESOFT.COM

While we cannot guarantee to answer every email message, we will pay careful attention to your comments and suggestions. We are always improving our product communications for you.

CHAPTER 1

Introducing What's New

This chapter provides an overview of the new features for PeopleTools PeopleCode and discusses the following topics:

- New classes.
- Writing your own classes.
- Using Binary sorts.
- Additional changes.
- PeopleCode Debugger.
- Database changes in the FieldChange event.

Understanding What's New

What's new for PeopleTools PeopleCode? Plenty! There are new classes, new properties and methods, enhancements to the debugger, as well as the ability to create your own classes, to name a few.

This section provides an overview of the new features. This section is written for developers who have a background in PeopleCode. If you're new to PeopleCode, you should probably start with the next chapter. See "Understanding PeopleCode and Events."

As a licensee of PeopleTools, you are licensed to use the base portal technology, which is limited to navigation to licensed PeopleSoft applications. If you want to register additional non-PeopleSoft content, customize your homepage, or create any pagelets, you must license PeopleSoft Enterprise Portal.

New Classes

The following new classes have been added:

- Application classes
- Chart class
- Exception class

- Notification classes
- SOAPDoc class
- XmlDoc class

See the individual class documentation for more details.

Writing Your Own Classes

PeopleSoft now enables you to write your own classes, or extend the provided PeopleCode classes, using the Application Classes. For example, you could create your own class that inherited the rowset class, then create your own property or method that you could use with every rowset in your component.

Use Application classes to help you structure your existing PeopleCode functions. You can replace your function libraries with application packages. Application classes provide more structure. Using the Application Packages, you have a clear definition of each class, as well as its listed properties and methods. This makes it easier to create a complex program that uses many functions.

Considerations Using Application Classes

PeopleTools has added the following words as “reserved words” for the PeopleCode syntax:

- catch
- class
- create
- end-class
- end-method
- end-try
- extends
- get
- import
- instance
- method
- out
- private

- property
- set
read-only
- throw
- try

These words are used in the new Application class. The Application class functionality expands PeopleCode into a full object-oriented language: developers can now create their own classes, inherit from the classes Peoplesoft delivers, and so on.

If there are any existing database fields that use these reserved names *and* are referenced in PeopleCode, you may now get an error when you try to execute this code.

All developers should run the "Compile All PeopleCode" utility on the database that contains their PeopleCode programs. This utility goes through all PeopleCode programs in a database and lists all invalid statements. Developers should then clean up any reported errors.

See Also

PeopleTools PeopleBook: PeopleCode Reference, "Application Classes"

Using Binary Sorts

PeopleTools now supports the use of databases that sort character data using language-sensitive sorting, instead of a purely binary order. As a result, developer can no longer expect the database sort order is always binary. This could impact binary searches and the results of binary comparisons.

Developers should review their code as this may impact COBOL, PeopleCode and SQR programs.

- Keep all non-SQL sorting in binary order, and provide a wrapper around the SQL ORDER BY functions that force the database into sorting data in binary order. As long as any SELECT ... ORDER BY statement whose results are expected to match a memory-ordered list use this wrapper, the list's sequence matches (except in the case of a Japanese Shift-JIS server and Unicode application server which is considered a fringe case). Any SELECT ... ORDER BY statements not expected to match a memory-ordered list (as in, the vast majority of them) are now sorted in the linguistic order of the database. Search dialog list boxes and scroll areas are now sorted correctly.
- The following ways of doing sorts now use linguistic sorts:
 - SortScroll built-in function
 - Sort method of Rowset class
 - Sort method of Array class

- Any in-memory sorting performed using COBOL language functions will be performed as a binary sort in the current character set used for COBOL processing, and may not necessarily match the sort order returned by the database in response to an ORDER BY clause. Should you require the database to return data sorted using a binary sort of its encoding rather than the default linguistically-correct sort, you must use the %BINARYSORT MetaSQL function around each column in the WHERE or ORDER BY clause where binary ordering is important.
- For OS/390 implementations, you must keep in mind that this binary sorting is only equivalent when the COBOL program is run OS/390 server. For example, the binary sort produced in COBOL differ from the binary sort produced by the database, as the database is encoded in EBCDIC and the client is in an ASCII-based encoding. Therefore, the %BINARYSORT Meta-SQL function should only be used in COBOL programs that are not run using RemoteCall where the OS/390 platform is not supported as a RemoteCall server.
- When running against non-OS/390 systems, the %BINARYSORT function can be used in both RemoteCall and non-RemoteCall programs. For example:

```
SELECT RECNAME FROM PSRECDEFN WHERE %BINARYSORT(RECNAME) < %BINARYSORT('xxx')
SELECT RECNAME FROM PSRECDEFN ORDER BY %BINARYSORT(RECNAME)
```

Note. Using the %BINARYSORT token in your WHERE and ORDER BY clauses negates the use of any indexes, as most databases can't use indexes for functional comparisons (For example, `where %BINARYSORT(column) > 'X'`). Use this syntax only when sorting equivalence of SQL statement results and COBOL memory order is absolutely required.

See Also

PeopleTools PeopleBook: Peoplesoft Global Technology, "COBOL Globalization"

Additional Changes

Several additional changes have been made to the PeopleCode editor and runtime environment.

In order to validate your PeopleCode, you should run the **Compile All PeopleCode** utility on the database containing your PeopleCode. This utility goes through all PeopleCode programs in a database and lists all invalid statements. You should clean up any reported errors.

- If you don't declare a variable, it is automatically declared as Any. A warning messages is displayed in the Validate tab.
- You can also assign a variable a value in the declaration statement. For example, the following is now valid:

```
Local Boolean &Bool=True;
```

- You can now declare local variables in the PeopleCode program itself (not just at the beginning). The following code snippet is now valid:

```

Local TransformData &incomingData = %TransformData;
Local XmlDocument &tempDoc = &incomingData.XmlDoc;

If (&tempDoc.ParseXmlString("<?xml version=\"1.0\"?><xml/>")) Then

    /* Create an array to hold the name value pairs */
    Local array of array of string &inNameValuePairsAry;

    /* Load the array with some values */
    &inNameValuePairsAry = CreateArray(CreateArray("locale", "en_us"),
    CreateArray("uom", "box"));

```

- If you declare a local variable inside a function it is a *new* variable each time the function is called. Its scope is only from the declaration to the end of the function.
- If you try to assign an invalid value in a declaration statement, you'll receive a warning in the Validate output window about the type mismatch. For example, the following produces a warning.

```
Local Boolean &bool = "yes";
```

- If a function is defined with a return value, but no return value is specified, the PeopleCode program is invalid. This means that PeopleCode programs that use to run no longer run. You'll receive an error either at design time or at runtime.
- The Exit built-in function accepts only numeric values. Boolean parameters are no longer allowed. PeopleCode programs that use to run no longer run. You'll receive an error either at design time or at runtime.
- For this release, several new built-in functions have been added:
 - IsAlpha
 - IsAlphaNumeric
 - IsDate
 - IsDateTime
 - IsDigits
 - IsNumber
 - IsTime
 - Max
 - Min
 - NumberToString

If you have already created a function to do one of these operations, you need to look at whether or not these new built-ins give you the same functionality. You tend to get better performance by using a PeopleCode built-in instead of a user-defined one.

If you have created a function *with the same name* as one of the built-ins, the user-defined function takes -precedence over the built-in one. However, you will get a warning message at design time.

- You can no longer use the following without committing all database changes:
 - StoreSQL built-in function
 - DeleteSQL built-in function
 - AERSection Save method

This is to avoid locking critical Tools tables and hence freezing all other users. You'll receive a runtime error message for StoreSQL, DeleteSQL, and AERSection using the Save method, in cases where there are pending database updates. The PeopleCode program terminates. You need to commit any database updates prior to using StoreSQL, DeleteSQL, and the AERSection Save method. The CommitWork PeopleCode function has been enhanced to allow this.

- The following charting tools are not supported in PeopleTools 8.4:
 - Pinnacle Chart Server
 - SimpleGraph

PeopleTools now enables you to create charts using PeopleCode. Any chart created using SimpleGraph or using the Pinnacle chart server no longer work. Developers need to rework any charts to use the new Peoplesoft charts.

- PeopleTools has added the ability to store file attachments in the database as BLOBs or externally on a file server. For any transactions that upload file attachments stored in the database, developers need to create a database record for PeopleTools to store attachments in the database and use this new framework. This allows PeopleTools to manage where the file attachment is stored (that is, in the database as a BLOB or on a separate file server).
- The ChangeOnInit rowset property has been changed. Before this release, ChangeOnInit did *not* propagate to child rowsets. Now, it automatically propagates to all child rowsets.

This means that if this property is set for a parent rowset before doing an insert row, it's automatically set on any and all child rowsets.

In addition, parent rows are marked as changed if this property is only set for the child rowset. That is, if you've changed a level 3 child of a level 2 row, the level 2 and level 1 rows are marked as changed to maintain data integrity.

See Also

"PeopleCode Language"

PeopleCode Debugger

You can use the PeopleCode debugger for two-tier and three-tier debugging. The database and application can reside on remote servers; they do not need to reside on the local machine.

Two-tier debugging works "out of the box." Setting up three-tier debugging requires you to make a few modifications in PSADMIN (PSAPPSRV.CFG) to enable debugging.

The following list includes the changes from the previous PeopleCode debugger:

- The Debugger no longer uses COM for its communication. The Unregister COM components have been removed from Configuration Manager. It is no longer required.
- The Debugger Account Setup and DCOMCNFG steps are no longer necessary.
- The psdbgbrkr.exe has been replaced with PSDBGSRV.EXE. In three-tier, TUXEDO launches this server, and In two-tier, the PeopleTool processes launch it.
- You can connect to an NT Server domain that is not on your local machine. Configuring a local domain is not required.
- Having the PeopleTools software installed locally for three-tier debugging is not required.
- The PeopleCode debugger runs on UNIX.

Database Changes in the FieldChange Event

Prior to this release, you could make changes to the database in the FieldChange event by using the **CallAppEngine** function. This functionality has been expanded, and you can now make other changes to the database in the FieldChange event. To support this, the **CommitWork** built-in function has also been enhanced to enable it to be used in FieldChange as well as SavePreChange.

An appropriate use for this functionality is saving user application preferences. For example, suppose you want to provide in your application a way for the end-user to personalize a list or preference, but you don't want to send the user to a different component. With this new functionality, you could display a simple secondary page where the new preferences are saved to the database immediately.

There are several important considerations with this feature.

- All updates done in FieldChange (including those using **CallAppEngine**) should be considered a *single database transaction*. This is a fundamental change: previously, a single transaction was represented by a page or a component.

- A consequence of this is that a message requiring a reply, or any other think-time action, causes a *fatal* error if located in FieldChange *after* a database update that has *not* been committed to the database using the **CommitWork** function. So it is possible for an application to update the database in FieldChange, then do a think-time action, by preceding the think-time action with a call to CommitWork.

Note. If you use DoSaveNow to commit the changes to the database, you will still receive this fatal error. You must use CommitWork.

- CommitWork commits the updates and closes the database transaction (that is, the unit of work). The consequence of using CommitWork is that because it closes the database transaction, any subsequent rollback calls will *not* rollback the committed updates.
- Just as any database updates in FieldChange required careful application design to ensure that the transaction model is appropriate, so too does the use of CommitWork.
- When using CommitWork in the Component Processor environment (as opposed to using it in a PeopleSoft Application Engine program) CommitWork produces an error if there are any open cursors, such as any open PeopleCode SQL objects.

Considerations for Existing PeopleCode Programs

Due to these changes, in a couple of special cases, your code may now produce errors.

- If you open a Component Interface in a FieldChange PeopleCode program, *and* you use that Component Interface to update the database, *and* after you change the database, you issue a message requiring a reply, or any other think-time action, your code may now produce errors. To avoid this, you must use **CommitWork** after you close the Component Interface and before you start any user interactive events.
- If you have two consecutive calls to **CallAppEngine** in a FieldChange event, followed by a think-time event or something that would require a user response (like DoModal), your code may now produce errors. To avoid this, you must issue use **CommitWork** between calls to CallAppEngine, and before you start any user interactive events (like DoModal or a message requiring a user response.)

CHAPTER 2

Understanding PeopleCode and Events

This chapter provides an overview of PeopleCode and events and discusses how to:

- Access PeopleCode in Application Designer.
- Use record field PeopleCode.
- Use component record field PeopleCode.
- Use component record PeopleCode.
- Use component PeopleCode.
- Use page PeopleCode.
- Use menu item PeopleCode.
- Use application message PeopleCode.
- Save and store PeopleCode programs.
- Copy PeopleCode with a parent definition.

Understanding PeopleCode Programs and Events

Every PeopleCode program is associated with some aspect of an Application Designer definition and with an event. Events are predefined points either in the Component Processor flow or in the program flow. As each point is encountered, the event fires on each definition, triggering any PeopleCode program associated with that definition and that event. Each class of definitions in Application Designer can have an *event set*—a group of events appropriate to that definition. A definition can have zero or one PeopleCode programs for each event in its event set.

Application Engine PeopleCode

An Application Engine program can have a PeopleCode program as an action. Though the right-hand drop-down menu on the PeopleCode editor window shows the text "OnExecute," it really isn't an event. Any PeopleCode contained in an Application Engine action is executed only when the action is executed.

Component Interface PeopleCode

A Component Interface can have user-defined methods associated with it. These methods are not part of any

processor flow. They are called as needed by the program executing the Component Interface.

See Also

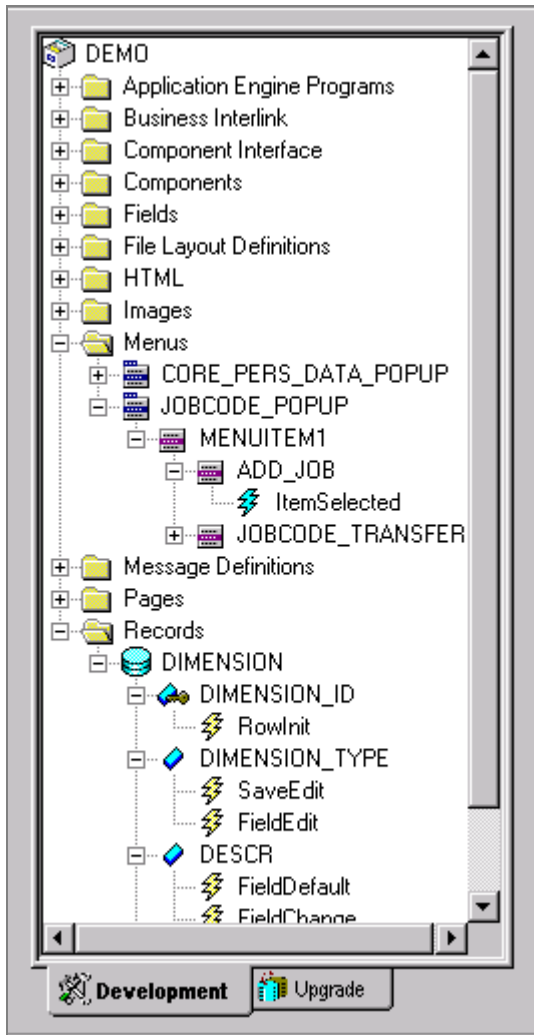
PeopleTools PeopleBook: Peoplesoft Application Engine, "Creating Application Engine Programs," PeopleCode Actions

"Component Interface Classes"

Accessing PeopleCode in Application Designer

You can access PeopleCode associated with the various Application Designer definitions in several ways.

With record fields and pop-up menu items, the Project View displays PeopleCode programs within the project hierarchy using a lightning bolt symbol. The programs are children of the fields and pop-up menu items with which they're associated, and are named according to their associated events, such as ItemSelected, RowInit, or SaveEdit, as shown in the following illustration. Double-click a record field or pop-up menu item program in the Project View to start the PeopleCode Editor and load that program for editing.



PeopleCode programs in the Project View hierarchy

You can associate PeopleCode with many other types of definitions, such as:

- Components
- Pages
- Component Interfaces
- Messages

Such PeopleCode programs do not display in the Project View. Instead, right-click the definition's name and select View PeopleCode from the pop-up menu. You can also access them from their associated definitions, as described for each definition in this chapter.

PeopleCode can also be associated with:

- component records, that is, with specific records included in components.

- component record fields, that is, with specific record fields included in components.

Because component record fields, component records and ActiveX controls do not appear in the Project View, you must access their associated programs through their parent definitions, as described for each definition in this chapter.

See Also

Record Field PeopleCode

Component PeopleCode

Using Record Field PeopleCode

A record is a table-level definition, such as an SQL table, a View, or a Derived/Work Record. Record fields are child definitions of records, and Record Field PeopleCode programs are child definitions of record fields. A record field can have zero or one PeopleCode programs for each event in the record field event set.

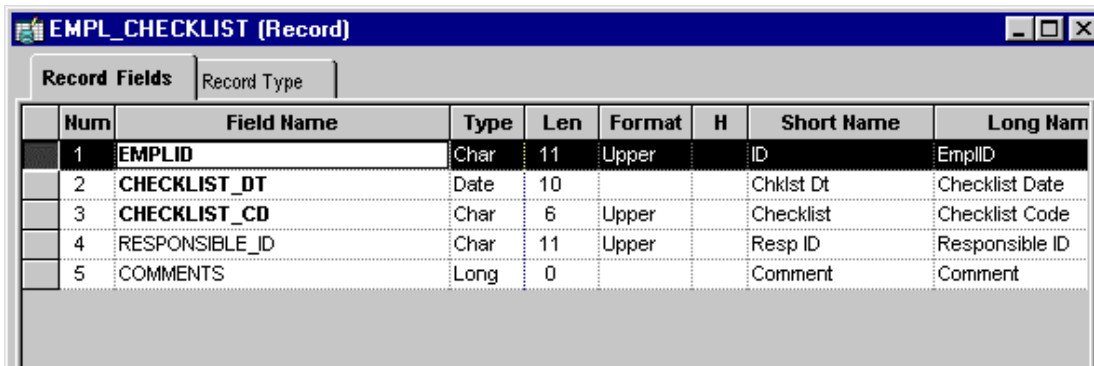
In addition to accessing programs through the Project View, you can also access them from the record definition, and from any page definition that includes that record.

In this section, we discuss how to:

- Access record field PeopleCode from a record definition.
- Access record field PeopleCode from a page definition.
- Distinguish between fields and record fields.
- Associate events with a record field.

Accessing Record Field PeopleCode From a Record Definition

Record definition fields that have PeopleCode associated with them appear bold in all record views.



EMPL_CHECKLIST (Record)							
Record Fields		Record Type					
Num	Field Name	Type	Len	Format	H	Short Name	Long Name
1	EMPLID	Char	11	Upper		ID	EmplID
2	CHECKLIST_DT	Date	10			Chklist Dt	Checklist Date
3	CHECKLIST_CD	Char	6	Upper		Checklist	Checklist Code
4	RESPONSIBLE_ID	Char	11	Upper		Resp ID	Responsible ID
5	COMMENTS	Long	0			Comment	Comment

Record Definition with three fields with PeopleCode

In the previous example, the first three fields (in bold) have PeopleCode associated with them. In addition, if you expand the subrecords in a record definition, any fields in the subrecord that have PeopleCode associated with them display in bold.

To access record field PeopleCode from an open record definition:

1. Click the PeopleCode Display button on the toolbar.

A grid appears with a column for each event in the record field event set. Each cell represents a field-event combination. The column names are abbreviations of the record field event names, for example, **FCh** for the FieldChange event and **RIn** for the RowInit event. A checkmark appears in the appropriate cell for each field/event combination that has an associated PeopleCode program.

2. Access the PeopleCode.

You can access the PeopleCode for a given cell using one of these methods:

- Double-click the cell.
- Right-click the cell, and select View PeopleCode from the pop-up menu.
- Select View, PeopleCode from the main menu.

The PeopleCode Editor appears. If the field/event combination has an associated program, it displays in the editor.

See Also

Record Field Event Set

"Using the PeopleCode Editor"

Accessing Record Field PeopleCode From a Page Definition

You can associate a PeopleCode program with any page control that you can associate with a record field.

To access record field PeopleCode from a page definition, right-click a page control and select View Record PeopleCode from the pop-up menu. The PeopleCode Editor appears, displaying the first event in the event set associated with that control's underlying record field.

Pushbutton controls are a special case. You can associate a PeopleCode program with a pushbutton only if its destination is defined as *PeopleCode Command*. When the end-user clicks a pushbutton defined using this method, the FieldEdit and FieldChange events are triggered, so the PeopleCode must be associated with one of those two events. Typically, you use the FieldChange event.

Page Field Properties dialog box for pushbuttons

To define a command pushbutton:

1. In the page definition, double-click the pushbutton to access its properties.
2. Select *PeopleCode Command* as the pushbutton Destination.
3. Select the record and field with which your pushbutton, and PeopleCode, are associated.

It's best to associate the pushbutton with a Derived/Work record field, which separates its PeopleCode from the PeopleCode associated with any of the page's other underlying record fields. You can then store generic PeopleCode with this field so you can reuse it with pushbuttons on other pages.

4. Click **OK** to return to the page.

Right-click the pushbutton and select View PeopleCode from the pop-up menu to access the PeopleCode Editor.

See Also

"Using the PeopleCode Editor"

Distinguishing Between Fields and Record Fields

It's important to distinguish clearly between *fields* and *record fields*:

- Fields are standalone definitions, on the same level as records, whose attributes are shared across all records that use the field.
- Record fields are owned by the record definitions that include them.

Note. Fields and record fields are also distinct from *component record fields*. These are the record fields used in a component, as they appear in the component's structure view. Within that context, they have their own independent event sets and PeopleCode programs.

Properties of a field, such as data type and size, affect all records that include the field; therefore any change to a field property affects all records that include the field. Properties of a record field, such as PeopleCode programs and key settings, are not shared among records; a change to a record field property affects only the record that owns the record field.

Note. PeopleCode programs are owned by record fields and component record fields, not fields.

See Also

Component Record Field PeopleCode

Record Field Event Set

The following are the events that can be associated with a record field:

- FieldChange Event
- FieldDefault Event
- FieldEdit Event
- FieldFormula Event
- RowInit Event
- RowSelect Event
- RowDelete Event
- PrePopup Event
- SaveEdit Event
- SavePreChange Event

- Workflow Event
- SavePostChange Event
- SearchInit Event
- SearchSave Event

See Also

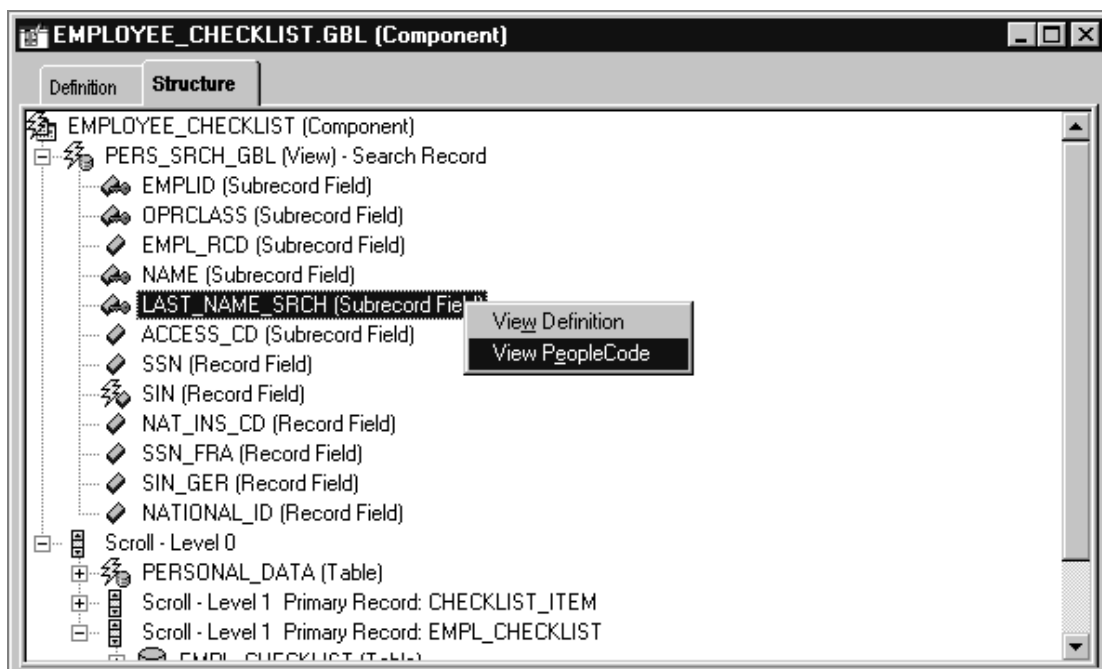
"PeopleCode and the Component Processor"

Using Component Record Field PeopleCode

Component record field PeopleCode is different from *record field* PeopleCode.

Component record field PeopleCode is associated with a record field, but only with respect to a component and one of its events. Use this type of association to tailor your programs to a particular component. This PeopleCode is only accessible through a component's structure display, not from any record definition.

To access PeopleCode associated with a component record field, open the component's structure view, select a field, right-click the field name, and select View PeopleCode from the pop-up menu. A lightning bolt displays next to the field name if PeopleCode is associated with the field *at the component level*. If PeopleCode is associated with the field at the record level, a lightning bolt does not display.



Accessing Component Record Field PeopleCode from the Component Structure

The PeopleCode Editor appears. If that field has associated PeopleCode, the first program in the component record field event set displays in the editor.

Through a component's structure display, you can access the definition of a *record field* included in a page of the component. That record field has its own event set and associated PeopleCode.

See Also

"Using the PeopleCode Editor"

Record Field PeopleCode

Component Record Field Event Set

- FieldChange Event
- FieldDefault Event
- FieldEdit Event
- PrePopup Event

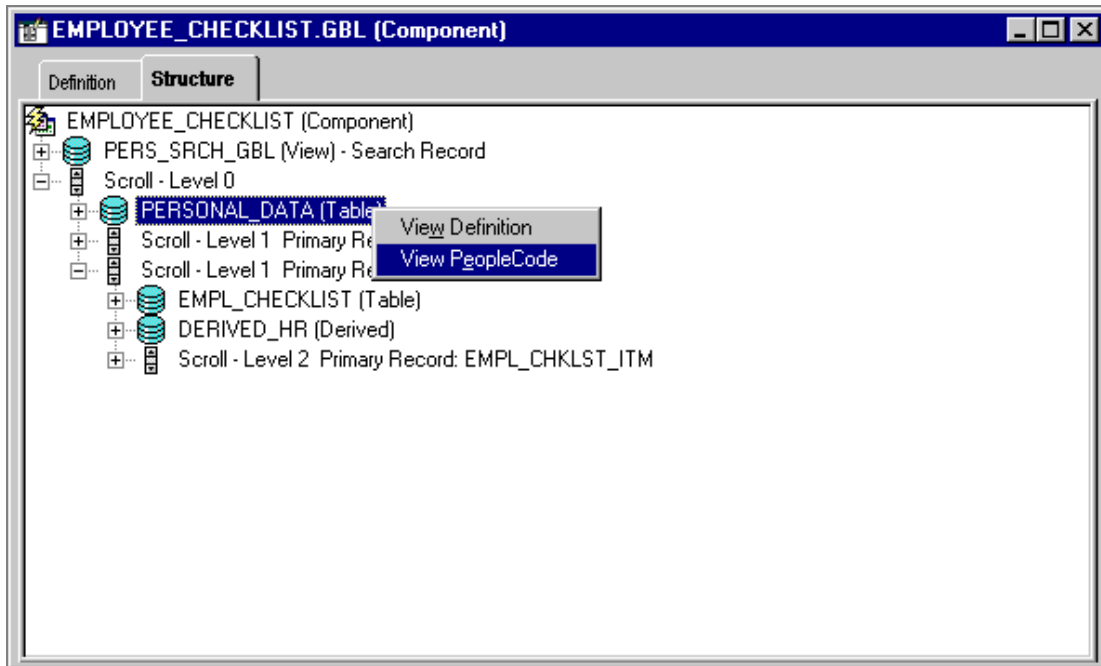
See Also

"PeopleCode and the Component Processor"

Using Component Record PeopleCode

Component record PeopleCode is associated with a record definition, but only with respect to a component and one of its events. Use this type of association to tailor your programs to a particular component. This PeopleCode is directly accessible through a component's structure display, not from the record definition.

To access PeopleCode associated with a component record, open the component's structure view, select a record, right-click the record name, and select View PeopleCode from the pop-up menu.



Accessing Component Record PeopleCode from the Component Structure

The PeopleCode Editor appears. If that record has associated PeopleCode, the first program in the component record event set displays in the editor.

See Also

"Using the PeopleCode Editor"

Component Record Event Set

Search records and non-search records in components have different associated event sets. The following events are associated with component *search* records:

- SearchInit Event
- SearchSave Event

The following events are associated with component *non-search* records:

- RowDelete Event
- RowInit Event

In certain rare circumstances, the Component Processor does not run RowInit PeopleCode for some record fields. The Component Processor runs RowInit PeopleCode when it loads the record from the database. However, in some cases, the record can be initialized entirely from the keys for the component. When this happens, RowInit PeopleCode isn't run.

- RowSelect Event

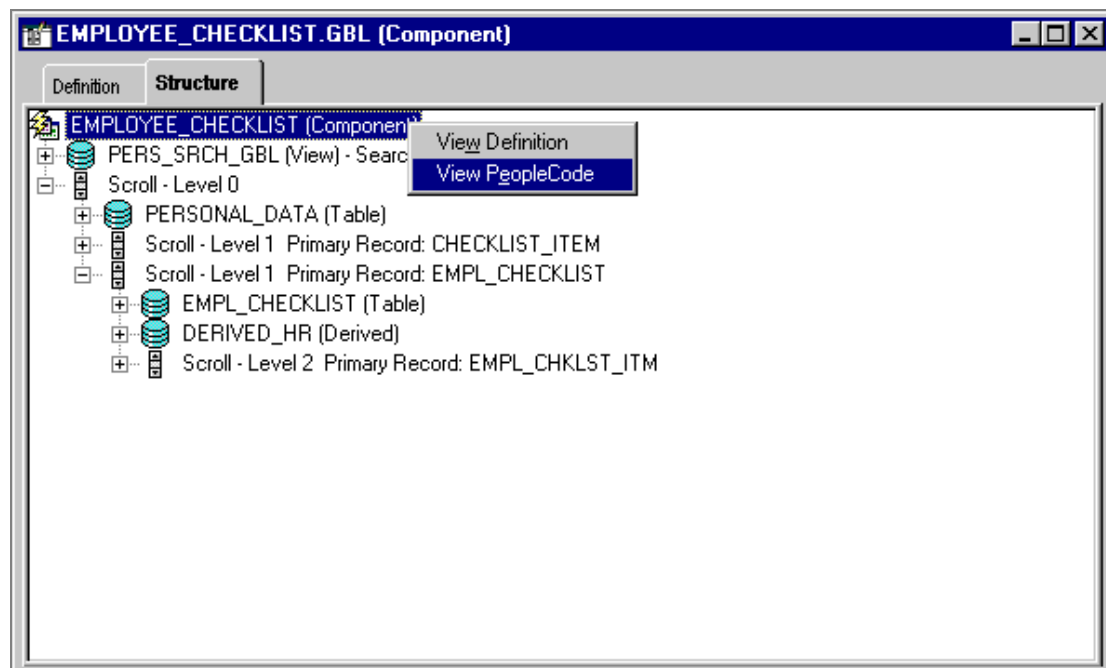
- SaveEdit Event
- SavePostChange Event
- SavePreChange Event

See Also

"PeopleCode and the Component Processor"

Using Component PeopleCode

Component PeopleCode is associated with a component definition and an event. To access PeopleCode associated with a component, open its structure view, select the component name, right-click the name, and select View PeopleCode from the pop-up menu.



Accessing Component PeopleCode from the Component Structure

The PeopleCode Editor appears. If that component has associated PeopleCode, the first program in the component event set displays in the editor.

See Also

"Using the PeopleCode Editor"

Component Event Set

- PostBuild Event
- PreBuild Event
- SavePostChange Event
- SavePreChange Event
- Workflow Event

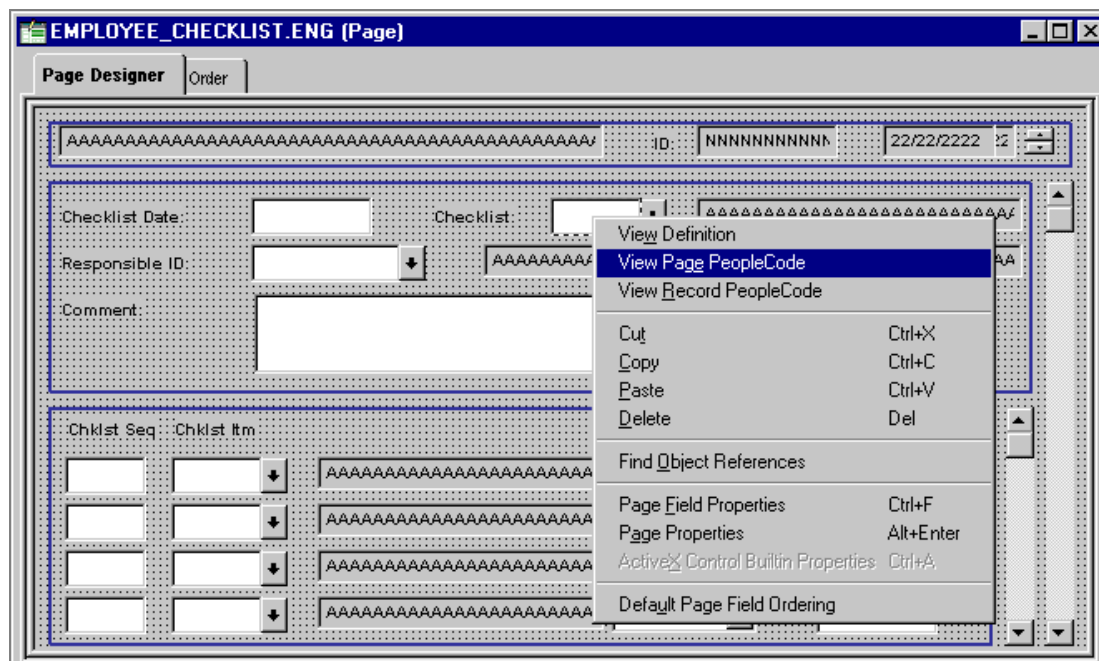
See Also

"PeopleCode and the Component Processor"

Using Page PeopleCode

Page PeopleCode is associated with a page definition. Currently, pages have only the *Activate* event. This event is valid only for pages that are defined as Standard or Secondary. This event is *not* supported for subpages.

To access PeopleCode associated with a page, right-click on any part of the page's definition, except an ActiveX control, and select View Page PeopleCode from the pop-up menu.



Accessing Page PeopleCode from the Page Definition

The PeopleCode Editor appears. If that page has associated PeopleCode, it displays in the editor.

Note. The term "page PeopleCode" refers to PeopleCode programs owned by pages. Do not to confuse page PeopleCode with PeopleCode properties related to the appearance of pages, such as the Visible Page Class property.

See Also

PeopleTools PeopleBook: PeopleCode Reference, "Page Class"

"Using the PeopleCode Editor"

Page Activate Event

The page event set consists of a single event, the *Activate* event, which fires every time the page is activated.

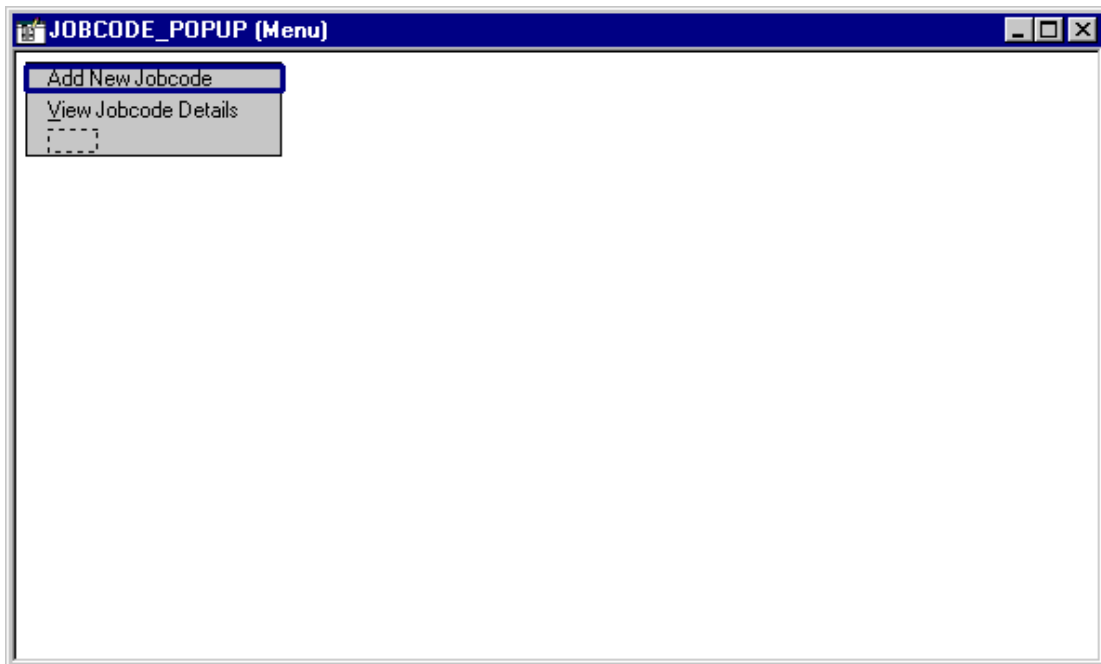
See Also

"PeopleCode and the Component Processor," Activate Event

Using Menu Item PeopleCode

PeopleTools menus are either one of two types, pop-up and standard, both of which are standalone definitions in the project hierarchy. However, you can only associate PeopleCode with menu items in pop-up menus.

Menu item PeopleCode programs are associated with pop-up menu items, which are child definitions of pop-up menus.



Pop-up Menu Definition in Application Designer

To define a PeopleCode pop-up menu item:

1. In the open pop-up menu definition, double-click the menu item to access its properties. If you're creating a new menu item, double-click the empty rectangle at the bottom of the pop-up menu.
2. The Menu Item Properties dialog box appears. If this is a new menu item, enter a name and a label for the item.
3. Select PeopleCode from the Type group.
4. Click **OK** to close the Menu Item Properties dialog box.

To access pop-up menu item PeopleCode:

1. Open the pop-up menu definition.
2. Access the PeopleCode.

Right-click the menu item. If it's defined as a PeopleCode menu item, View PeopleCode is enabled on the pop-up menu.

3. Select View PeopleCode.

The PeopleCode Editor appears with that menu item's associated program, if any, displayed.

Note. The term "menu item PeopleCode" refers to PeopleCode programs owned by menu items. Do not to confuse menu item PeopleCode with PeopleCode functions related to the appearance of menu items, such as CheckMenuItem.

See Also

PeopleTools PeopleBook: PeopleSoft Application Designer, "Creating Menu Definitions"

"Using the PeopleCode Editor"

Menu Item ItemSelected Event

The menu item event set consists of a single event, the ItemSelected Event. This event fires whenever an user selects a menu item from a pop-up menu; therefore this is where you put PeopleCode programs that are executed from menu items.

See Also

"PeopleCode and the Component Processor", ItemSelected Event

Using Message PeopleCode

Message PeopleCode is associated with a message definition.

In this section, we discuss how to:

- Access message PeopleCode.
- Associate events with messages.

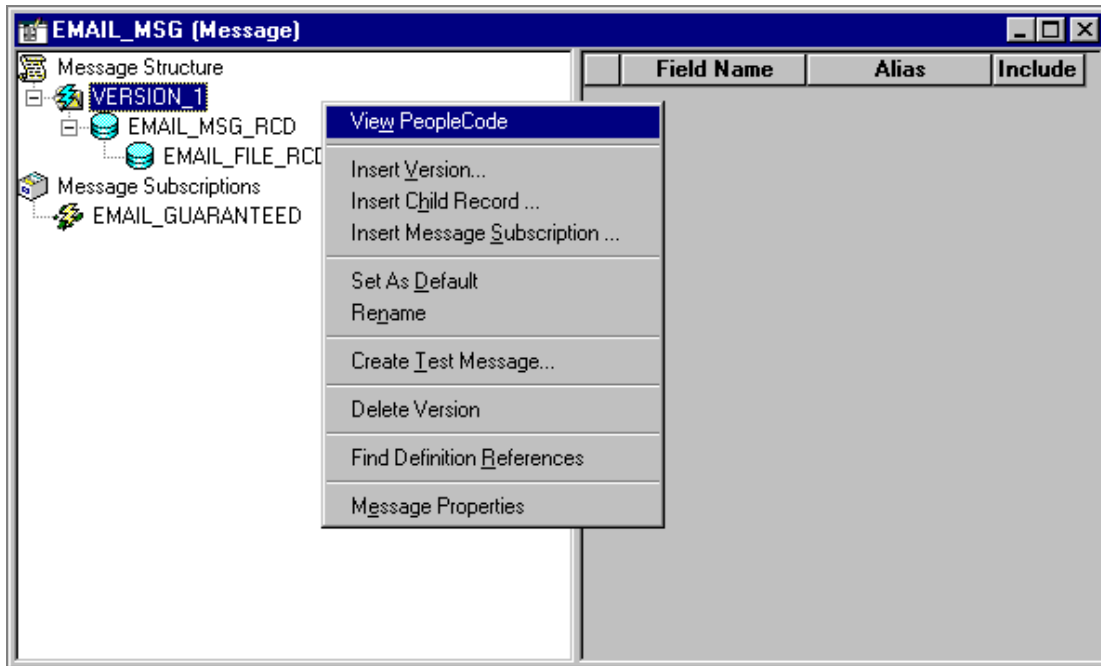
See Also

PeopleTools PeopleBook: PeopleSoft Integration Broker, "Defining Messages"

Accessing Message PeopleCode

In a message definition, you can associate a PeopleCode program with the message itself and with each message subscription included in the definition.

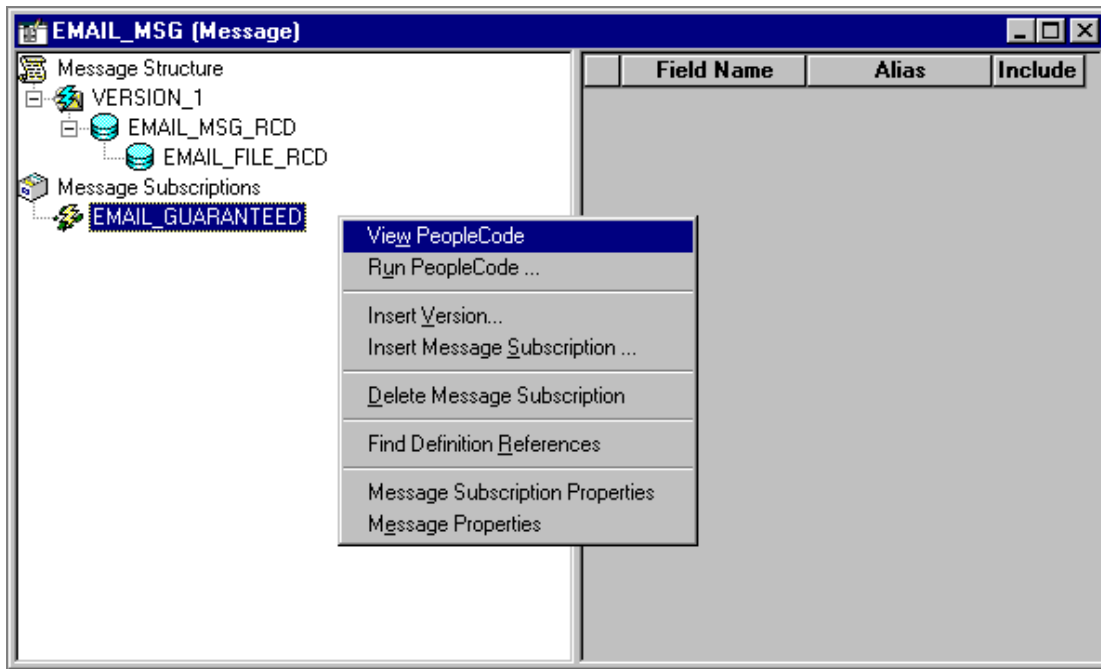
To access PeopleCode associated with a message, open the message definition, right-click anywhere in the Message Structure display, and select View PeopleCode from the pop-up menu.



Accessing Message PeopleCode from the Message Definition

The PeopleCode Editor appears. If that message definition has associated programs, the first one in the application message event set displays in the editor.

Each message subscription entry in a message definition represents a PeopleCode program. To access that program, open the message definition, select the subscription, right-click the subscription name, and select View PeopleCode from the pop-up menu.



Accessing Message Subscription PeopleCode from the Message Definition

The PeopleCode Editor appears, with the message subscription's associated program displayed in the editor.

See Also

"Using the PeopleCode Editor"

Message Event Sets

These events are not considered part of the Component Processor flow, so they're documented separately from the majority of PeopleCode events. The following events are associated with messages:

- OnPublishTransform Event
- OnSubscribeTransform Event
- OnRouteSend Event
- OnRouteReceive Event
- OnRequest Event

See Also

PeopleTools PeopleBook: Peoplesoft Integration Broker, "Defining Messages"

Storing and Saving PeopleCode Programs

When you save an Application Designer definition, PeopleTools saves all PeopleCode programs belonging to that definition that have been added or modified. Component record field PeopleCode and component record PeopleCode belong to the component. If new PeopleCode programs have been added, PeopleTools creates an association between the owning definition and new rows in the PeopleCode table. If a PeopleCode program is deleted, the association between the owning definition and the program is also deleted.

Saving PeopleCode programs together with their owning definitions helps to guarantee the integrity of the application. In the event of system failure, it's unlikely that your application could end up storing PeopleCode programs that are not associated with any definition; or worse, a definition associated with a PeopleCode program that was never saved.

Note. When you save any Application Designer definition, all PeopleCode programs that you've added or changed since the last save are checked, formatted, and saved at the same time. To format and check the syntax of a single PeopleCode program, use the Validate Syntax command instead of the Save command.

Automatic Backup of PeopleCode

A PeopleCode program is automatically saved to a file while you're working on it. This checkpoint occurs at the following times:

- Every 10 keystrokes.
- On a save command, just prior to the save being executed (in case the save doesn't actually execute because the code is invalid).
- When another PeopleCode program is selected to be edited (if you have two PeopleCode editor windows open at the same time, and you move from one to the other).

The file is saved to your temp directory (as specified in your environment), in a file with the following name:

```
PPCMDDYY_HHMMSS.txt
```

where MMDDYY represents the month, date, and year, respectively, of the checkpoint, and HHMMSS represents the hour, minute, and second, respectively.

The top of the checkpoint file contains the following information:

```
[PeopleCode Checkpoint File]
```

```
[RECORD.recordname.FIELD.fieldname.METHOD.eventname]
```

If your PeopleCode program is saved successfully, any checkpoint files associated with that program are automatically deleted.

Copying PeopleCode With a Parent Definition

When you create a copy of an Application Designer definition that contains PeopleCode, Application Designer enables you to choose whether to copy all PeopleCode programs and the definition. Each copy of the definition receives a separate copy of the PeopleCode programs.

To copy a definition with its PeopleCode:
--

1. Open the definition you want to copy.
2. Select File, Save As.

The Save As dialog appears. Type a name for the new definition in the dialog box.

3. Click **OK**, then click **Yes** to copy the PeopleCode.

Click Yes to copy all PeopleCode associated with the definition when asked

Upgrading PeopleCode Programs

You can upgrade PeopleCode programs independently of the definitions with which they're associated.

See Also

The Upgrade Documentation

CHAPTER 3

Using the PeopleCode Editor

This chapter provides an overview of adding PeopleCode to PeopleSoft applications through Application Designer and discusses how to:

- Navigate between PeopleCode programs.
- Use the PeopleCode Editor.
- Generate PeopleCode using drag-and-drop.

Navigating Between PeopleCode Programs

The PeopleCode Editor navigational features have been significantly enhanced, making it a more powerful and convenient tool. After you access a PeopleCode program associated with an Application Designer definition, you can access programs associated with other related definitions, without having to close the Editor window.

In this section, we discuss how to:

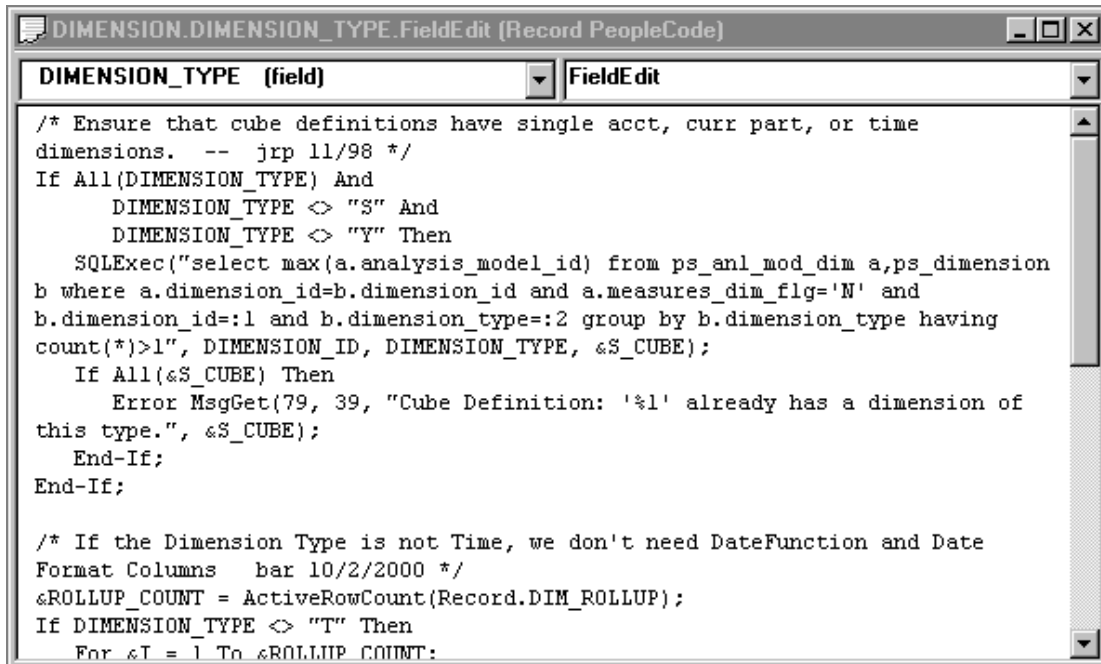
- Understand the PeopleCode Editor window.
- Use the drop-down definition list.
- Use the drop-down event list.

See Also

Understanding PeopleCode and Events

Understanding the PeopleCode Editor Window

Application Designer supplies an independent Editor window for each record, definition, menu, message, and so on, for which you invoke the Editor—these are the *parent definitions*. The Editor window's title bar displays the name and type of the parent *definition*.



PeopleCode Editor window with record field PeopleCode

The Editor window contains the main *edit pane*, the drop-down *definition list* at the upper-left, and the drop-down *event list* at the upper-right. The drop-down lists enable you to navigate directly to the PeopleCode associated with related *child definitions*—for example, fields within a record—and their event sets.

Note. When you make a selection from either drop-down list box, your selected entry has a yellow background, indicating that you must click in the edit pane before you can start typing.

You can open as many Editor windows as you want, and resize them in Application Designer. Each line of code wraps automatically, based on the window's current width. A vertical scroll bar appears if the program has more lines than the Editor can display in the edit pane.

Note. You cannot open two Editor windows for a single parent definition, or for any two of its child definitions.

See Also

Using the Drop-down Definition List

Using the Drop-down Event List

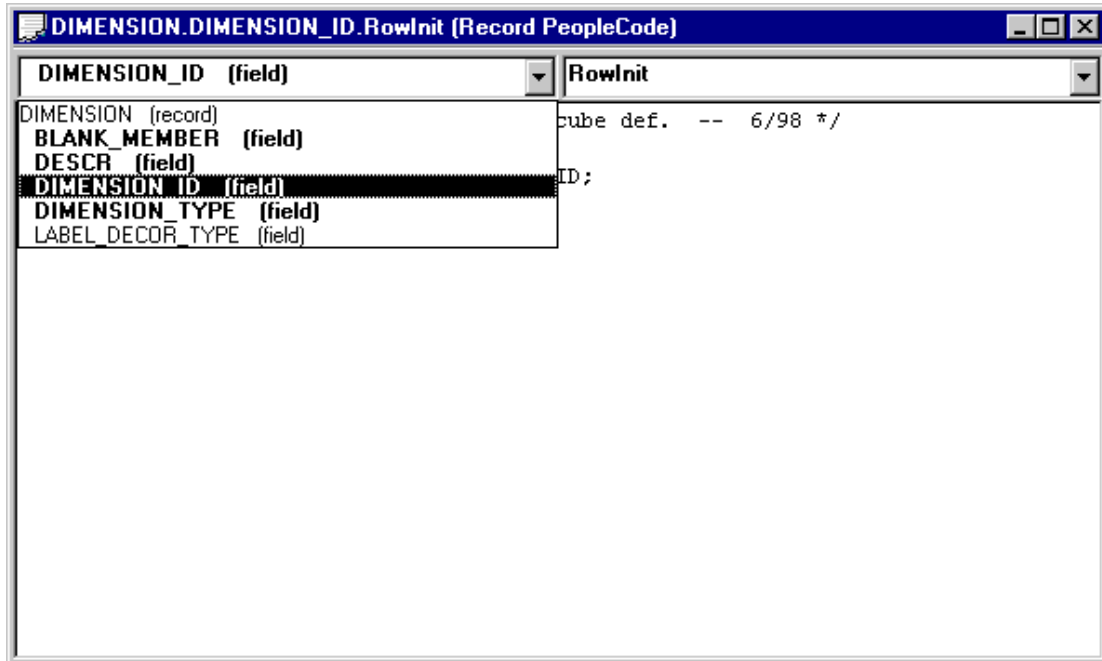
Using the Drop-down Definition List

The PeopleCode Editor's drop-down definition list enables you to navigate between PeopleCode programs that are associated with a given parent definition and its children. The

list displays the complete hierarchy of child definitions to which you can navigate. The structure of the definition list depends on the type of parent definition. Each type of list is illustrated and described in the following sections.

Selecting a Record Field

The record PeopleCode drop-down definition list displays all the record fields included in a record. Each record field displayed in bold has a PeopleCode program associated with at least one of the events in its event set.

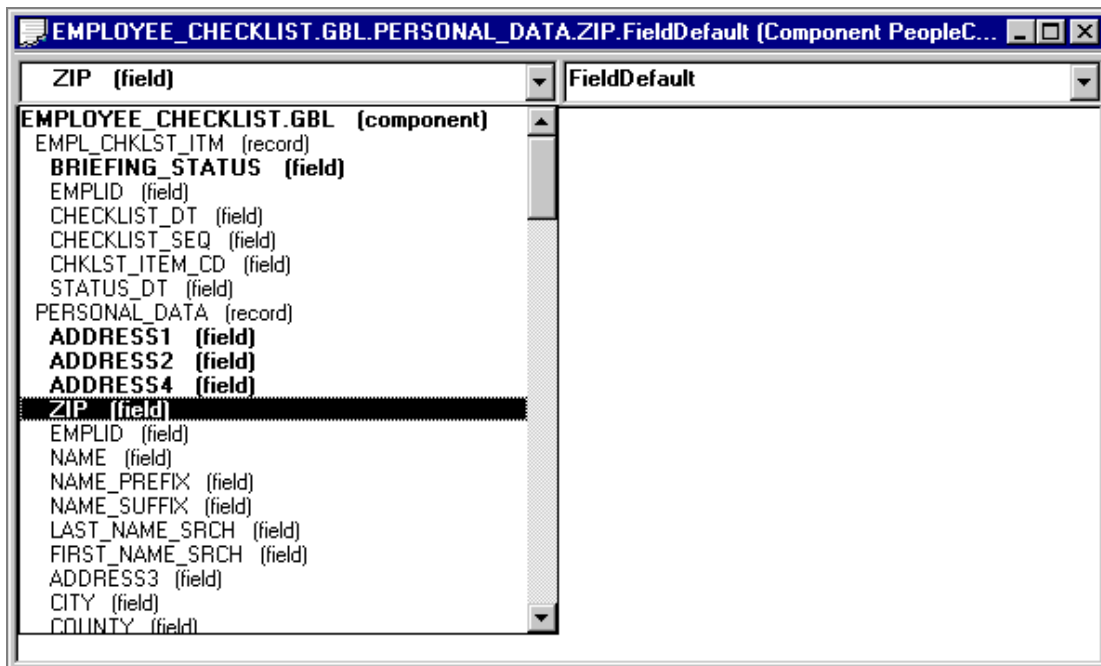


Selecting a record field from the Record Definition List

The record name appears at the top of the list, but you cannot associate PeopleCode with just a record; you can only associate it with a record field. The record name displays as a visual clue to clarify the location of the record fields.

Selecting a Component Definition

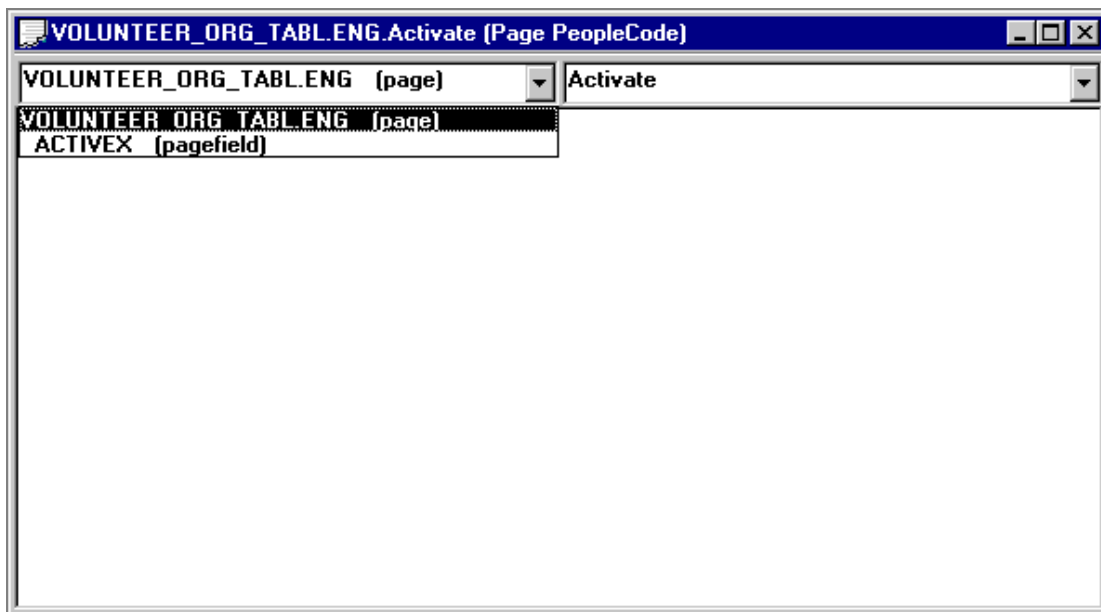
The component PeopleCode drop-down definition list displays all the component record fields and component records included in a component. Each definition displayed in bold has a PeopleCode program associated with at least one of the events in its event set.



Selecting a component record field from the Component Definition List

Selecting a Page

The page PeopleCode drop-down definition list displays the associated page. Each definition displayed in bold has a PeopleCode program associated with at least one of the events in its event set.

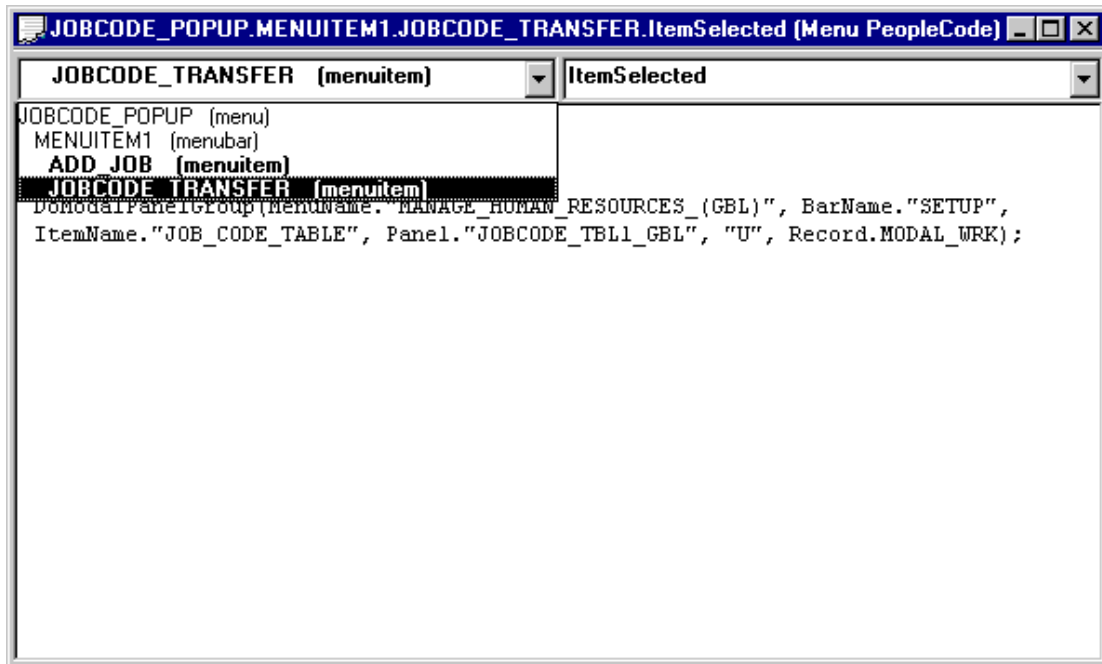


Selecting a page from the Page Definition List

Selecting a Pop-up Menu Item

The pop-up menu PeopleCode drop-down definition list displays the menu, its menu bars, and menu items. Each menu item displayed in bold has an associated PeopleCode program.

Important! You can associate PeopleCode only with menu items in a pop-up menu.

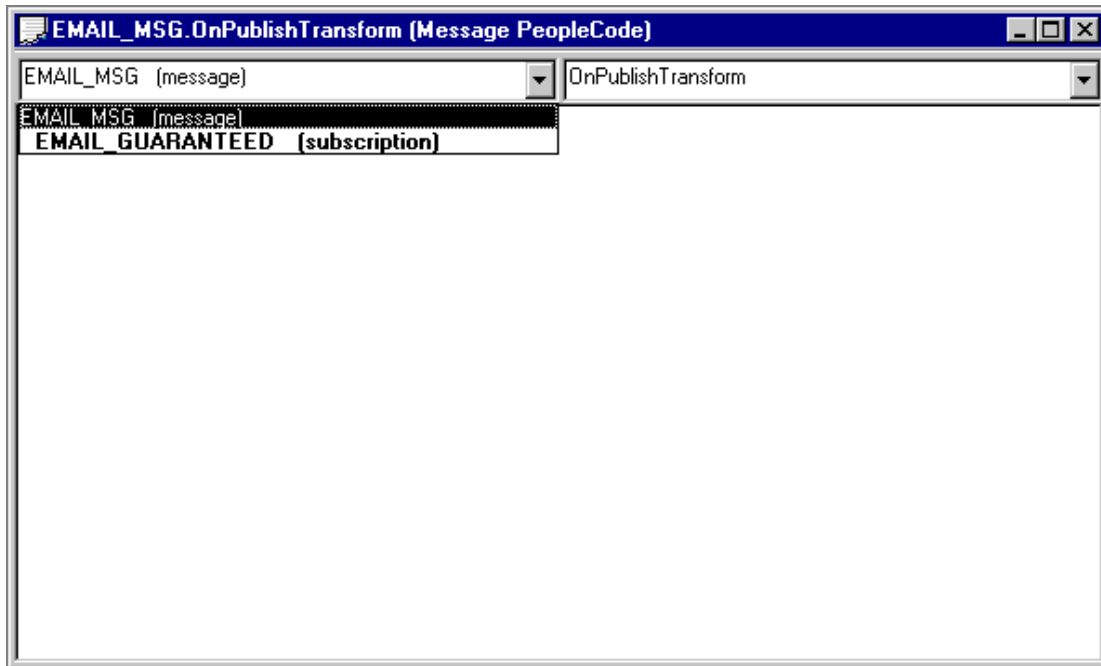


Selecting a PeopleCode menu item from the Menu Definition List

The menu and menu bar names appear on the list, but you cannot associate PeopleCode with just a menu or a menu bar; you can only associate it with a menu item. The menu and menu bar names display as a visual clue to clarify the positions of the PeopleCode menu items in the pop-up menu hierarchy.

Selecting a Message or Message Subscription

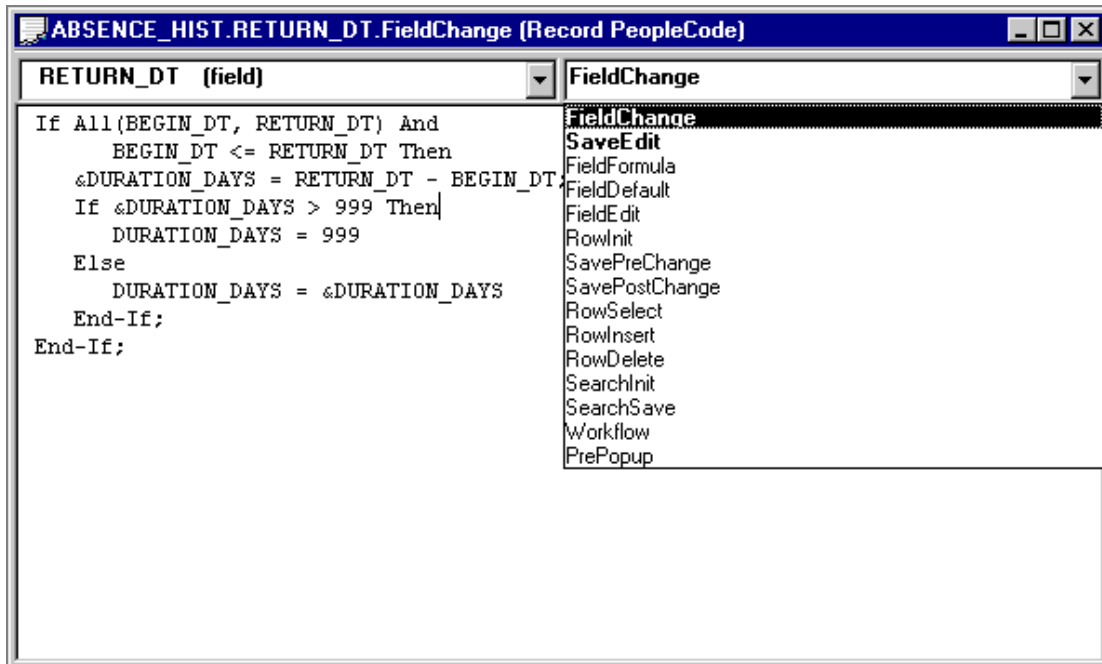
The message PeopleCode drop-down definition list displays a message, and all the message subscriptions included in the message definition. Each definition displayed in bold has a PeopleCode program associated with at least one of the events in its event set. The message subscriptions *are* PeopleCode programs, so they will always be bold.



Selecting a message from the Message Definition List

Using the Drop-down Event List

The PeopleCode Editor's drop-down event list enables you to select an event from the event set of the currently selected definition. Use this event list to navigate between PeopleCode programs that are associated with that definition. For every definition-event combination with associated PeopleCode, the event name is displayed in bold, and it appears at the top of the event list.



Selecting an event from the Record Event Set

See Also

“Understanding PeopleCode and Events“

“PeopleCode and the Component Processor“

Using the PeopleCode Editor

The PeopleCode Editor works much like any other text editor, but has capabilities specifically geared toward the PeopleTools environment.

- Its editing functions are integrated with the menus and toolbar of Application Designer and are also accessible from a pop-up window.
- It checks, formats, and saves all programs associated with Application Designer definitions simultaneously when any definition is saved.
- It includes a Validate Syntax command for checking and formatting a single PeopleCode program without saving.
- It supports standard Windows drag-and-drop editing.
- You can open separate instances of the Editor simultaneously, and you can use a drag-and-drop text operation between programs.
- You can open the definition with which the current set of PeopleCode programs is associated from within the PeopleCode Editor.

- You can open a field, record, page, file layout, message definition, or other definitions from a PeopleCode reference to the field, record, page, file layout, or message, and so on.
- You can access the PeopleCode programs associated with a field, record, page, file layout message definition, or other definitions from a PeopleCode reference to the field, record, page, file layout, or message, and so on.
- You can open a PeopleCode Editor window containing an external function definition from a function declaration or function call.
- You can press F1 with the cursor in a PeopleCode built-in function, method, meta-SQL, and so on, to open the PeopleSoft help for that item.

In this section, we discuss how to:




- Use editing functions.
- Use auto formatting.
- Use drag-and-drop editing.
- Access PeopleCode external functions.
- Access definitions and associated PeopleCode.
- Use context-sensitive help.
- Select a font for the PeopleCode Editor.





See Also

"PeopleCode and Events," How PeopleCode Programs are Stored and Saved

Editing Functions

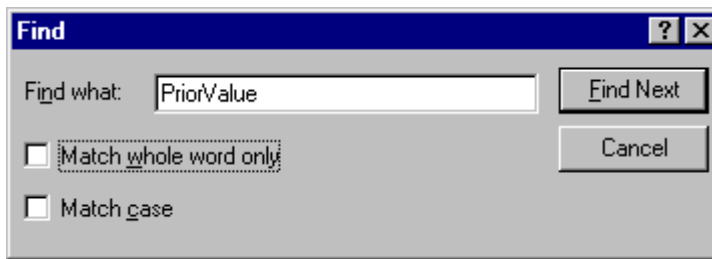
The PeopleCode Editor supports the standard editing functions Save, Cancel, Cut, Copy, Paste, Find, Replace, and Undo; from the PeopleCode Editor pop-up menu. Cut, Copy, and Paste use standard Windows keyboard shortcuts. You can also cut, copy, and paste within the same PeopleCode program or across multiple programs.

Command	Key	Button
Save	CTRL+S	
Cancel	ESC	
Cut	CTRL+X or SHIFT+DEL	
Copy	CTRL+C or CTRL+INS	

Command	Key	Button
Paste	CTRL+V or SHIFT+INS	
Find	CTRL+F	
Replace	CTRL+H	
Undo	CTRL+Z OR ALT+BKSP	
Validate		

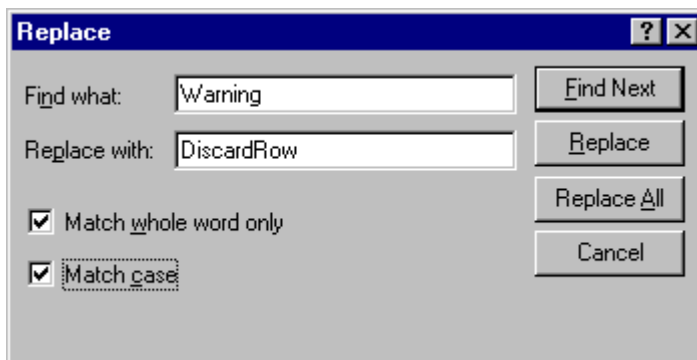
Find and Replace

When you use the Find and Replace functions, any text string that is highlighted appears when either the Find or Replace dialog boxes are called. For example, if you select the function PriorValue, it appears in the Find dialog box when it's called.



Find dialog box


You can move through finding and replacing text strings one string at a time, or click **Replace All** to replace globally. The Undo function is available to undo the last replace or replace all.



Replace dialog box

Validating Syntax

To check the syntax of the current PeopleCode program and format it if it is syntactically correct, do one of the following:

- Click the Validate Syntax button  on the Application Designer toolbar.
- Select Tools, Validate Syntax from the Application Designer menu.
- Right-click in the PeopleCode Editor window, then select Validate Syntax from the pop-up menu.

Validate is a utility with several functions, such as finding undeclared variables, mismatching data types, or invalid methods or properties for a class. You can check either a single component or an entire project.

Any errors or warnings produced by Validate are displayed in the Validate tab at the bottom of the PeopleCode Editor window.

Any variables that you don't declare are automatically declared for you, and a warning message displays in the Validate tab for each undeclared variable. You can right click in the Validate tab and select Clear to delete all the warnings listed there, then use the Validate Syntax utility again to ensure that your code runs without errors or warnings.

Note. This feature is convenient if you have written multiple PeopleCode programs and you want to check the syntax of one without saving. All of the PeopleCode programs associated with an item (record, component, message definition, and so on) are checked prior to saving.

See Also

PeopleTools PeopleBook: Peoplesoft Application Designer, "Using Peoplesoft Application Designer Projects," Validating Projects

"Using the PeopleCode Debugger," Compiling All PeopleCode Programs

Auto Formatting

You do not need to format your PeopleCode statements; you need only to use the correct syntax. In fact, when you save or validate, the system formats the code according to the rules in the PeopleCode tables—no matter how you entered it originally. It automatically converts field names to uppercase and indents statements for you. This makes your PeopleCode look consistent with other programs in the system.

PeopleCode is case-insensitive, except for quoted literals. PeopleCode does not format anything surrounded by quotation marks. String comparisons, however, are case-sensitive. When you compare the contents of a field or a variable to a string literal, make sure the literal is in the correct case.

Drag-and-Drop Editing

In addition to the standard keyboard shortcuts and toolbar buttons for editing, you can copy or move text within a window or between two PeopleCode Editor windows, using the mouse and the CTRL key on your keyboard.

Note. You cannot open two Editor windows for a single parent definition, or for any two of its child definitions.

To move text between instances of the PeopleCode Editor:

1. Select the text you want to move.
2. Click in the selection and drag the mouse to the other PeopleCode Editor window.
3. When the cursor appears at the place where you want to insert the text, release the mouse button.

To copy text between instances of the PeopleCode Editor:

1. Select the text you want to move.
2. Holding down the CTRL key, click in the selection and drag the mouse to the other PeopleCode Editor window.
3. When the cursor appears at the place where you want to insert the text, release the mouse button.

Accessing PeopleCode External Functions

An external PeopleCode function is a function written in PeopleCode (as opposed to a built-in function or external DLL function) and defined in a program outside the one from which it is called. External PeopleCode functions can be defined in any Record PeopleCode program, but typically they are stored in the FieldFormula event in records beginning with FUNCLIB_.

The PeopleCode Editor provides immediate access to external PeopleCode function definitions. Right-click the function name in the program where the function is called, then select View Function *FunctionName* from the pop-up menu. This opens a new PeopleCode Editor window containing the external function definition.

Note. Internet Scripts are contained in records similar to FUNCLIB_ records. However, they're named WEBLIB_XXX.

Accessing Definitions and Associated PeopleCode

You can open a field, record, page, message, and other definitions from the PeopleCode Editor. Or you can open a new PeopleCode Editor window containing the programs associated with a field, record, page, and so on.

To open a definition from the PeopleCode Editor:

1. Right-click a PeopleCode definition reference
2. Select View Definition from the pop-up menu.

For example, you could open definitions by clicking in any of the following references:

```
RECORD.BUS_EXPENSE_PER
BUS_EXPENSE_PER.EXPENSE_PERIOD_DT
PAGE.BUSINESS_EXPENSES
```

If you access a record definition from a record field reference (that is, *recordname.fieldname*) the specified record field is selected when the record definition opens.

To open a new PeopleCode editor window:

1. Right-click a reference to the definition.
2. Select View PeopleCode from the pop-up menu.

For example, you can access record PeopleCode from the following record and record field references:

```
RECORD.BUS_EXPENSE_PER
BUS_EXPENSE_PER.EXPENSE_PERIOD_DT
```

Context-Sensitive Help

The PeopleCode Editor has context-sensitive online reference help for all PeopleCode built-in functions, methods, properties, system variables, and meta-SQL. To access online help, place the cursor in the name of what you want to look up, then press F1. If there is a corresponding entry in the online reference system it will be displayed; otherwise a "No Help Available" error message appears.

If more than one entry is applicable, a pop-up window listing all applicable entries displays. Select the correct entry.

To use the F1 functionality, you must have specified where the documentation exists on the system, on the PeopleTools Options page, in *F1 Help URL*.

The following is an example of *F1 Help URL*:

```
http://Pandora/doc/flsearch.htm?ContextID=%CONTEXT_ID%&LangCD=%LANG_CD%
```

After you set the F1 help location, you must exit all PeopleTools sessions and start again before the F1 functionality is active.

See Also

PeopleTools PeopleBook: Data Management, "PeopleTools Utilities," PeopleTools Options

Selecting a Font for the PeopleCode Editor

The default font for the PeopleCode Editor is 9-point Courier New. You can change the font in the PeopleCode Editor. This may be necessary for some languages that do not display correctly using Courier New.

To change the PeopleCode editor font:
--

1. Select Tools, Options.
2. Select the PeopleCode tab.

Use this dialog to change the font in the PeopleCode editor.

Note. When you select a font for the PeopleCode Editor, the font selection dialog provides choices based on a character set appropriate for your international version of Windows. If you experience trouble embedding foreign characters (such as Thai characters) in PeopleCode, you might need to change the font setting. If you are trying to display Thai characters in Windows 95, you might also need to change your keyboard input settings for the characters to display correctly. You can change your keyboard input settings from the Input Locales tab on the Windows Regional Settings control page, or on the Keyboard control page.

Generating PeopleCode Using Drag-and-Drop

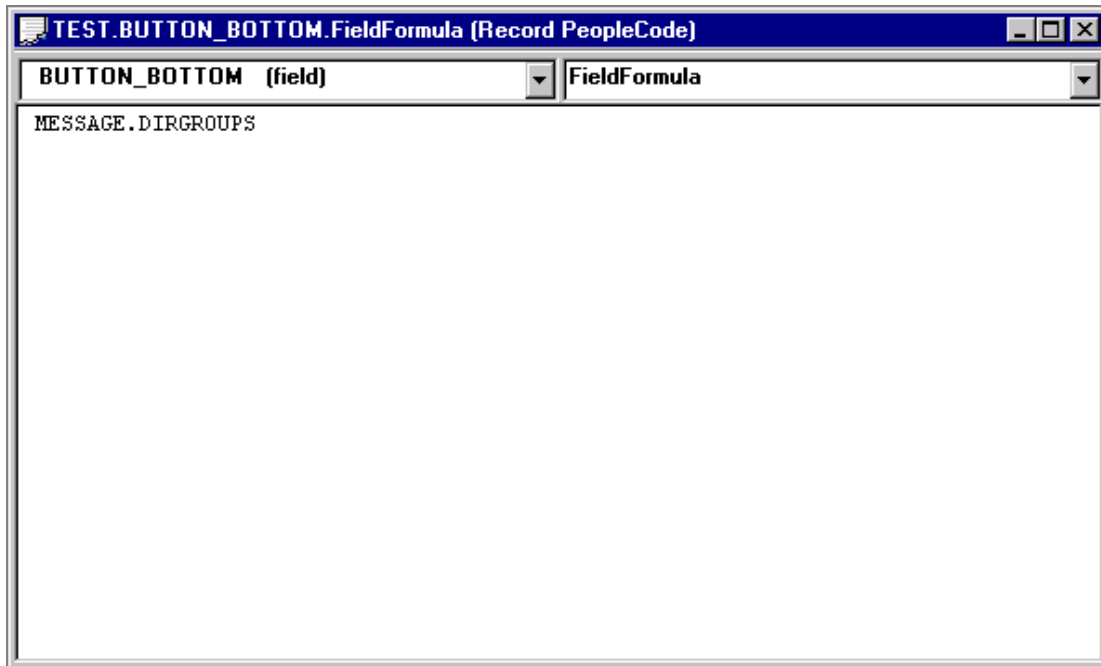
You can generate references to definitions using a drag-and-drop operation. You can also generate PeopleCode templates for accessing Business Interlinks and Component Interfaces.

In this section, we discuss how to:

- Generate definition references.
- Generate PeopleCode for a Business Interlink.
- Generate PeopleCode for a Component Interface.

Generating Definition References

You can drag-and-drop definitions, such as menus, records, record fields, pages, and so on, from a project into an open PeopleCode editor window. Doing this generates a reference to the definition. For example, suppose your project contains a message definition named DIRGROUPS. If you select the DIRGROUPS message definition from the project, drag it into an open PeopleCode window, the following PeopleCode Editor window appears:



Example PeopleCode editor window

Generating PeopleCode for a Business Interlink

After you create a Business Interlink definition, you must use PeopleCode to instantiate an Interlink object and execute the interlink plug-in. This PeopleCode can be long and complex. Rather than write it directly, you can drag and drop the Business Interlink definition from Application Designer's Project View into an open PeopleCode edit pane. Application Designer analyzes the definition and generates initial PeopleCode as a template, which you can modify to suit your purpose.

The following is just a snippet of the code that is generated:

```
/* ==>
    This is a dynamically generated PeopleCode template to be used only as a
    helper to the application developer.
    You need to replace all references to '<*>' OR default values with references
    to PeopleCode variables and/or a Rec.Fields.*/

/* ==> Declare and instantiate: */
Local Interlink &QE_AE_NONSSL__1;
```

```

Local BIDocs &inDoc;
Local BIDocs &outDoc;
Local boolean &RSLT;
Local number &EXECRSLT;
&QE_AE_NONSSL__1 = GetInterlink(INTERLINK.QE_AE_NONSSL_BI);

.

.

.

.

```

See Also

PeopleSoft Business Interlink Application Developer Guide

Generating PeopleCode for a Component Interface

After you create a Component Interface definition, you can use PeopleCode to access it. This PeopleCode can be long and complex. Rather than write it directly, you can drag and drop the Component Interface definition from Application Designer Project View into an open PeopleCode edit pane. Application Designer analyzes the definition and generates initial PeopleCode as a template, which you can modify to meet your requirements.

The following is just a snippet of the code that is generated:

```

/* ==>
This is a dynamically generated PeopleCode template to be used only as a
helper
to the application developer.
You need to replace all references to '<*>' OR default values with references
to PeopleCode variables and/or a Rec.Fields. */

Local ApiObject &oSession;
Local ApiObject &oCurrencyCdCi;
Local ApiObject &oPSMessageCollection;
Local ApiObject &oPSMessage;
Local File &LogFile;
Local number &i;
Local String &strErrMsgSetNum, &strErrMsgNum, &strErrMsgText, &strErrMsgType;

.

.

.

```

You can also access a Component Interface using COM. You can automatically generate a Visual Basic template, a Java template, or a C template, similar to the PeopleCode template, to begin.

To generate a template:

1. Open a Component Interface in Application Designer.
2. Right-click anywhere in the open Component Interface.
3. Select the template type that you want to generate from the pop-up menu.

You must save the Component Interface before generating the template.

When the template is successfully generated, a message displays with the full path and name of the file containing the template.

4. Open the generated file and modify the source code to meet the needs of your application.

The following is the initial code snippet that is generated:

Option Explicit

'==>

'This is a dynamically generated Visual Basic template to be used only as a

' helper

'to the application developer.

'You need to replace all references to '<*>' OR default values with references

' to

'Visual Basic variables.

Dim oSession As PeopleSoft_PeopleSoft.Session

Private Sub ErrorHandler()

 '***** Display PeopleSoft Error Messages *****

 If Not oSession Is Nothing Then

 If oSession.ErrorPending Or oSession.WarningPending Then

 Dim oPSMessageCollection As PSMessageCollection

 Dim oPSMessage As PSMessage

 Set oPSMessageCollection = oSession.PSMessages

 Dim i As Integer

 For i = 1 To oPSMessageCollection.Count

 Set oPSMessage = oPSMessageCollection.Item(i)

 Debug.Print "(" & oPSMessage.MessageNumber & ", " & oPSMessage.MessageSetNumber & ") : " & oPSMessage.Text

 Next i

 '***** Done processing messages in the collection; OK to

delete *****

 oPSMessageCollection.DeleteAll

```

        End If
    End If
End Sub

.

.

.

.

```

See Also

PeopleSoft Component Interfaces

Generating PeopleCode for a File Layout

After you create a File Layout definition, you can use PeopleCode to access it. This PeopleCode can be long and complex. Rather than write it directly, you can drag and drop the File Layout definition from Application Designer Project View into an open PeopleCode edit pane. Application Designer analyzes the definition and generates initial PeopleCode as a template, which you can modify to meet your requirements.

The following is just a snippet of the code that is generated:

```

Function EditRecord(&REC As Record) Returns boolean ;
    Local integer &E;
    REM    &REC.ExecuteEdits(%Edit_Required + %Edit_DateRange + %Edit_YesNo +
    %Edit_TranslateTable + %Edit_PromptTable + %Edit_OneZero);
    &REC.ExecuteEdits(%Edit_Required + %Edit_DateRange + %Edit_YesNo +
    %Edit_OneZero);
    If &REC.IsEditError Then
        For &E = 1 To &REC.FieldCount
            &MYFIELD = &REC.GetField(&E);
            If &MYFIELD.EditError Then
                &MSGNUM = &MYFIELD.MessageNumber;
                &MSGSET = &MYFIELD.MessageSetNumber;
                &LOGFILE.WriteLine("****Record:" | &REC.Name | ", Field:" |
                &MYFIELD.Name );
                &LOGFILE.WriteLine("*****" | MsgGet(&MSGSET, &MSGNUM, ""));
            End-If;
        End-For;
        Return False;
    Else
        Return True;
    End-If;
End-Function;

.

```

-
-
-

CHAPTER 4

Introducing the SQL Editor

This chapter provides an overview of the SQL editor and discusses how to:

- Access the SQL editor.
- Understanding SQL properties.
- Use the SQL editor.

See Also

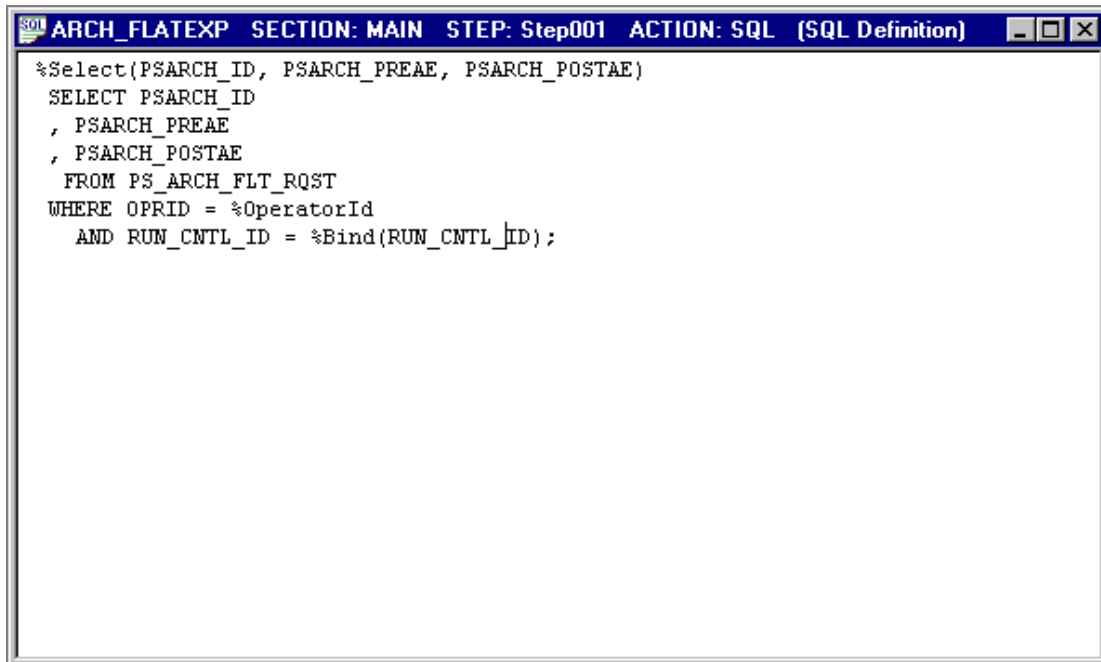
wPeopleTools PeopleBook: Peoplesoft Integration Broker, "Applying Transformation, Translation and Filtering"

Understanding the SQL Editor Window

Use the SQL editor to create SQL for SQL definitions, record views, and Application Engine programs.

The SQL editor and the PeopleCode editor interfaces are similar. You can add, delete, and change text: you can use the find and replace function; and you can validate the SQL. When you save a SQL definition, the code is automatically formatted (indented, and so on), the same as it is for a PeopleCode program.

The Editor window's title bar displays either the name of the SQL definition, or the name of the component that contains the SQL. For example, if the SQL statement is part of an Application Engine program, the names of the program, the section, the step and the action are listed in the title bar.



SQL Editor Window with Application Engine Program SQL

The Editor window consists of the main *edit pane*.

For SQL definitions and SQL used with records, there is also a drop-down *database list* at the upper left. For SQL definitions, you can also use a drop-down *effective date list* at the upper right.

Note. When you make a selection from either drop-down list box, your selected entry has a yellow background, indicating that you must click in the edit pane before you can start typing.

Understanding SQL Properties

Access the definition properties for the SQL definition by either:

- Pressing ALT+Enter
- Selecting File, Definition Properties
- Right-clicking in the definition, and selecting Definition Properties.

Use the General properties to specify a description for the SQL definition as well as additional comments. The description displays in PeopleSoft Application Designer search lists.

Use the Advanced properties to display an effective date with the SQL definition.

Note. The Audit SQL field on the Advanced properties is not used this release.

Accessing the SQL Editor

In this section, we discuss how to use the SQL editor to create and edit SQL for the following components:

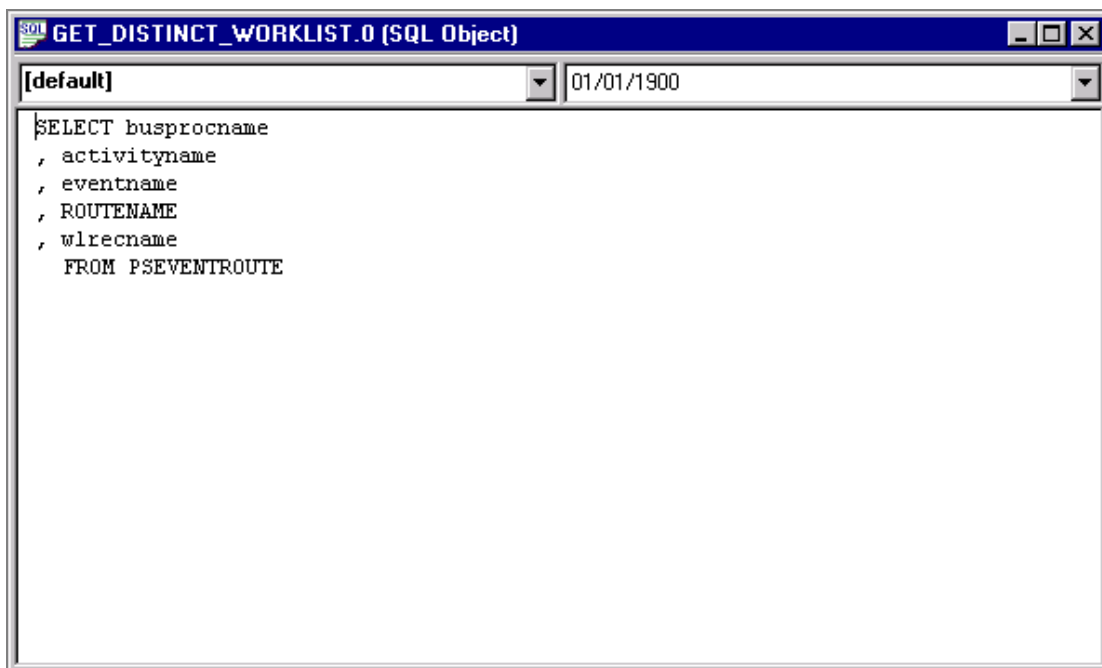
- SQL definitions.
- Dynamic view or SQL view records.
- Application Engine programs.

You access the SQL editor differently for each type of component.

SQL Definitions

A SQL definition contains SQL statements, which can be entire SQL programs or just fragments that you want to reuse. You can access a SQL definition using Application Designer. You can add SQL definitions to a project. In addition, they're upgradable.

You can create, change, or delete SQL definitions using Application Designer. You can also do the same programmatically with the SQL class in PeopleCode.



SQL definition with effective date

To create a SQL definition:

1. From Application Designer, select File, New, SQL.
2. Specify the database type that you want associated with the SQL definition.

You can associate more than one database type with a single SQL definition. In PeopleCode, you can specify the appropriate database type for the program. However, at least one of the SQL statements must be of type Default.

3. (Optional) Specify an effective date.

SQL definitions can be effective-dated. To specify an effective date with your SQL definition, go to the Advanced tab of the SQL definition object property. Access the object properties by doing one of the following:

- Go to File, Object Properties.
- Select the SQL definition, right-click, then select Object Properties.
- Press Alt + Enter.
- Click the Advanced tab, then click Show Effective Date. When you click OK, the SQL definition shows a date in the right-hand drop-down menu.

4. Add the SQL code.

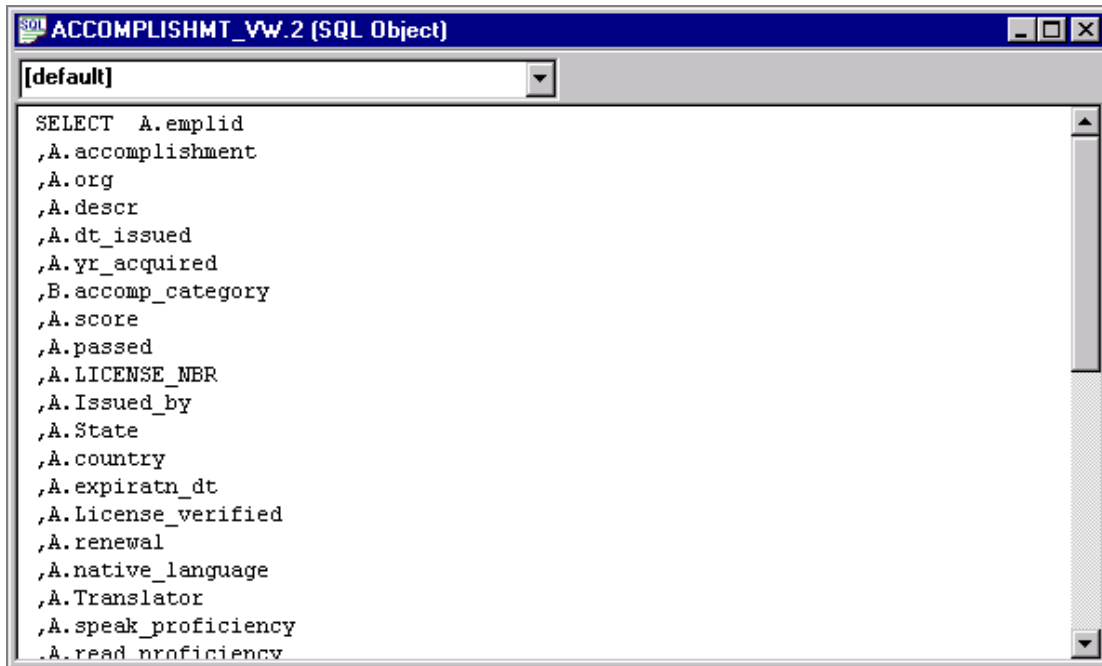
You do not need to format your code. The SQL editor formats it when you save the SQL definition.

See Also

PeopleTools PeopleBook: Peoplesoft Application Designer, "Using Application Designer"
"SQL Class"

Dynamic View or SQL View Records

When you create a SQL View or Dynamic View record definition, you must enter a SQL View Select Statement to indicate what field values you want to join from which tables. You do this in the SQL editor.



SQL Editor for SQL View Record Definition

To access the SQL editor with records:

1. Open or create the dynamic view or SQL view record definition.
2. Click the Record Type tab.
3. Click the Click to Open SQL Editor button.

You can select a database type, but not an effective date, from the SQL editor for dynamic view and SQL view record definitions.

Considerations with View SQL and UNIONS

If you've selected **Platform Compatibility** in Application Designer, you can not use UNIONS in a view. If you try to specify a UNION SELECT, you won't be able to save the SQL definition.

See Also

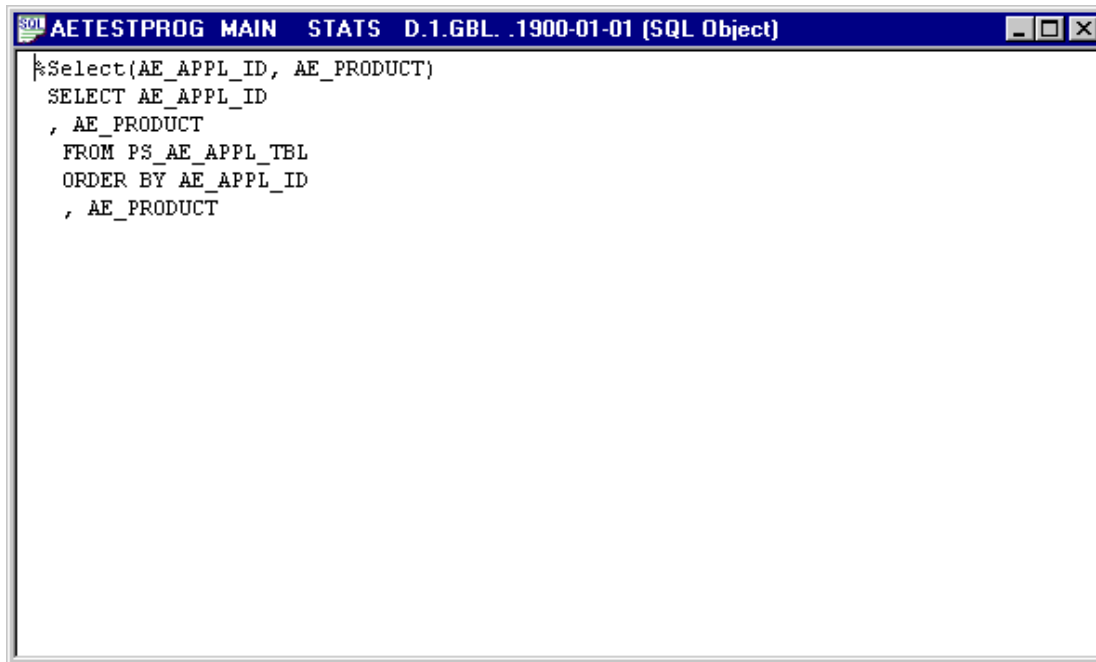
PeopleTools PeopleBook: Data Management, "PeopleTools Utilities," PeopleTools Options

Application Engine Programs

You can access the SQL editor from the following action types:

- Do Select

- Do Until
- Do When
- Do While
- SQL



Application Engine SQL Editor window

To access the SQL editor in an Application Engine program:

1. Open the Application Engine program.
2. Select the action.
3. Either right-click and select View SQL, or select View, SQL.

Select the database type and effective date for this SQL in the section, not in the SQL editor.

See Also

PeopleTools PeopleBook: Peoplesoft Application Engine, "Introducing Application Engine," Using Meta-SQL

Using the SQL Editor

The SQL editor works similar to any other text editor. You can use the same functions with it as with the PeopleCode editor: cut, paste, find, replace, and so on.

When you right-click in an open SQL editor window, the pop-up menu lists all the available functions for the SQL editor:

Format Display	
Validate Syntax	
Resolve Meta SQL	
Delete Statement	F8
Cut	Ctrl+X
Copy	Ctrl+C
Paste	Ctrl+V
Undo	Ctrl+Z
Find...	Ctrl+F
Replace...	Ctrl+H
Object Properties...	

SQL Editor pop-up menu

The following functions are available for the SQL editor, but are *not* available for the PeopleCode editor.

Function	Description
Format Display	You do not need to format your SQL statements; you need to use the correct syntax only. When you save or validate, the system formats the code according to the rules in the PeopleCode tables—no matter how you entered it originally. It automatically converts field names to uppercase and indents statements for you. The SQL then looks consistent with other programs in the system.
Resolve Meta-SQL	If there is meta-SQL in the SQL, select Resolve Meta-SQL to expand the meta-SQL statement in the Output Window, under the Meta-SQL tab.

For example, the following code expands as follows:

```
%Join(COMMON_FIELDS, PSAEAPPLDEFN ABC, PSAESECTDEFN XYZ)
```

<pre>ABC.AE_APPLID = XYZ.AE_APPLID AND ABC.VERSION = XYZ.VERSION AND ABC.LASTUPDOPRID = XYZ.LASTUPDOPRID AND ABC.LASTUPDDTTM = XYZ.LASTUPDDTTM</pre>
--

Meta-SQL expanded in Output Window

Delete Statement	You can delete standalone SQL statements. This menu item is not enabled with SQL statements that have a database type of Default with no effective date, or a database type of Default and an effective date of 01/01/1900.
------------------	---

See Also

"Using the PeopleCode Editor", Editing Functions

CHAPTER 5

Introducing Application Packages

This chapter provides an overview of Application Packages as well as the Application Packages editor and discusses how to:

- Understand application packages
- Use the Application Class Editor
- Edit application package classes

See Also

PeopleCode Reference PeopleBook, "Application Classes"

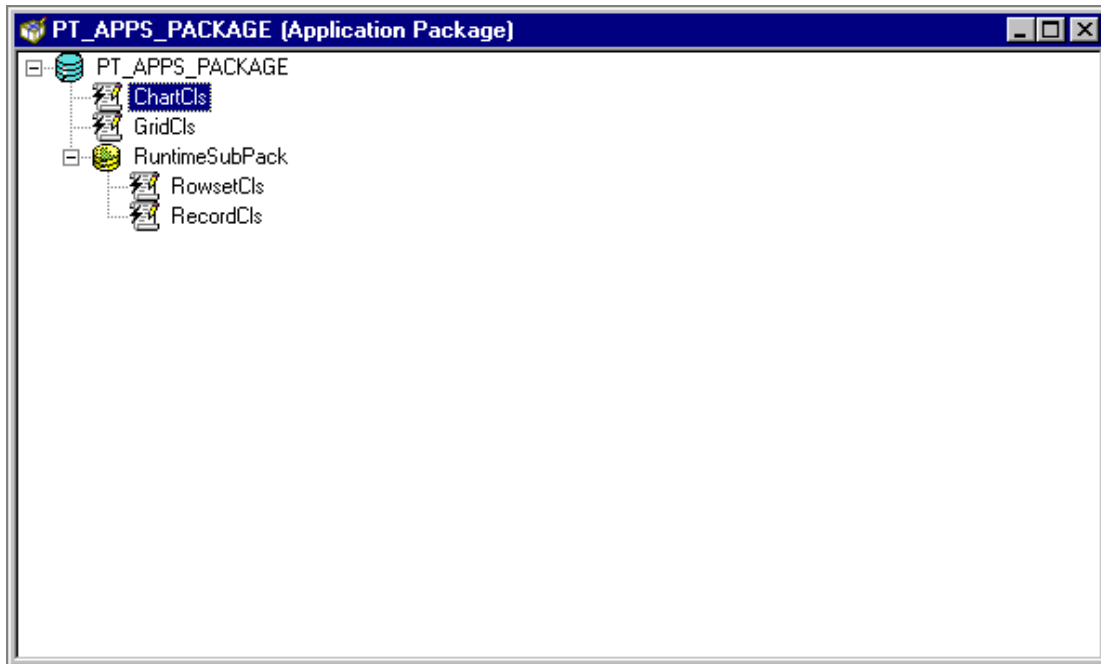
Understanding Application Packages

Use the Application Packages editor to create application packages. A *package* is a container for other packages or for application classes. A *subpackage* is any package other than the primary package—that is, the package defined parent package in the current definition.

The Editor window's title bar displays the name of the Application Package definition. The main window displays the classes and other application packages that make up the Application Package definition.

Application packages can contain classes, and possibly other packages, in a hierarchy. The primary package is represented by a blue database symbol, while subpackages are represented with yellow database symbols. *Classes* are represented by lightening bolts and scripts, and are used to host related application object classes (PeopleCode programs).

In the following example, PT_APPS_PACKAGE is the primary package, and RuntimeSubPack is a subpackage. ChartCls, GridCls are classes in the PT_APPS_PACKAGE Application Package, while RowsetCls and RecordCls are classes in the RuntimeSubPackage.



Application Package main window

You can create a subpackage with the same name as another package or subpackage within the same Application Package definition, as long as the fully qualified name is unique for each subpackage. Each subpackage is differentiated by the full path name of the class, that is, from the package definition name and the subpackage name.

For example, suppose in the application class PT_FRUIT, where PT_FRUIT is the primary class, you had the following structure of subpackages (no classes are listed in this example):

```
PT_FRUIT
  RAW
  RECIPIES
    RAW
    COOKED
  SMOOTHIES
    FROZEN FRUIT
    RAW
```

Given this example, there are three subpackages names RAW. The fully qualified name for each is unique. For example, the first one is qualified by the name of the primary package, that is:

```
PT_FRUIT:RAW
```

The others are also qualified by the subpackages that contain them.

```
PT_FRUIT:RECIPIES:RAW
```

```
PT_FRUIT:SMOOTHIES:RAW
```

Similarly, you cannot create two classes with the same name within a given package or subpackage. You can create classes with the same name within the same Application Package definition, just like subpackages, as long as their fully qualified name is unique. Each class is differentiated by the full path name of the class, that is, from the package definition name and the subpackage name.

Note. You cannot create a structure where there are more than two levels of subpackages defined below the primary package.

To create a new application package:

1. Open PeopleSoft Application Designer.
2. Select File, New, Application Package.

A new Application Package displays.

To insert a new package or class:

1. Open an application package.
2. Select a package.
3. Insert the new item.

Select the package, then select Insert, Package, or Insert, Application Class.

Using the Application Class Editor

When you right-click in an open Application Classes editor window, the pop-up menu lists all the available functions:



Application Class Editor pop-up menu

Cut	Not available for this release.
Copy	Not available for this release.
Paste	Not available for this release.
Delete	Deletes either a class or a package. The PeopleCode text is not actually deleted until you save the Application Package. Therefore, deleted classes PeopleCode can be recovered by reinserting the class node, as long as you haven't used save in the interim.
Insert App Class	Insert an application class. Classes themselves cannot have children (subclasses). Therefore, application classes can be inserted only into an existing package (the primary package or any subpackage).
Insert Package	Insert an application package. Only packages can have children (subpackages). Therefore, you can only insert packages from an existing package or subpackage. You cannot create a structure where there are more than two levels of subpackages defined below the root package.
Rename	Renames either a class or a subpackage. When you save the definition, all PeopleCode programs associated with the renamed class are also updated. To rename the primary package definition, use File, Rename.
View PeopleCode	View the associated PeopleCode. PeopleCode can be only defined for application classes, and is not directly related to package nodes.
Print	Print the application package definition, including all the PeopleCode in the classes.

Considerations Editing Application Classes

You can rename a class in an Application Package definition. When you rename a class, the PeopleCode that calls that class is automatically renamed.

The primary application package is named the same as the Application Package definition. To rename the primary application package, use the File, Rename function. Again, if you rename an Application package, any PeopleCode that uses that name is automatically also renamed.

The Application Package Editor does not support cut and paste operations for classes or subpackages in the tree view between existing packages, either in the same application package definition or in different definitions. Obviously, you can work around this by inserting new subpackage and class nodes where needed and use the clipboard to copy/paste PeopleCode text from class to class.

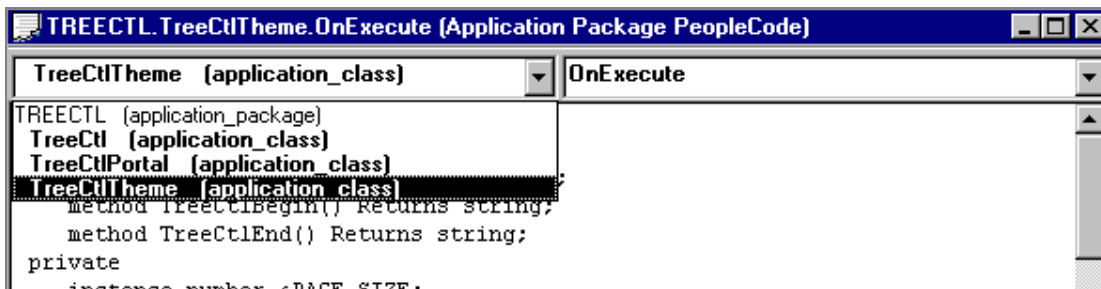
To copy the primary package of your definition, use File, SaveAs.

Editing Application Package Classes

From an Application Package you can access the PeopleCode programs associated with the classes of the package.

The Application Packages editor and the PeopleCode editor interfaces are similar. You can add, delete, and change text; you can use the find and replace function; you can validate the syntax. When you save your application package, the code is automatically formatted (indented and so on), just as it is in the PeopleCode editor.

The editor window contains the main *edit pane*, the drop-down *definition list* at the upper-left, and the drop-down *event list* at the upper-right.



Example Application Packages Editor

Only one event is defined for an application class, OnExecute. This isn't an event in the Component Processor flow. The application class is executed when called.

The drop-down list on the upper-left enables you to navigate directly to the PeopleCode associated with every class in the package, as well as every subpackage and its classes.

To edit an application class:

1. Open the Application Package.
2. Select a class.
3. Either select View, PeopleCode, or right-click and select View PeopleCode from the pop-up menu.

A PeopleCode editor window displays.

See Also

"Using the PeopleCode Editor," Editing Functions

CHAPTER 6

PeopleCode Language

This chapter provides an overview of the syntax and fundamental elements of the PeopleCode language and discusses the following topics:

- Data types.
- Statements.
- Functions.
- Expressions.
- Operators.

It assumes that you have some familiarity with a structured programming language, such as C or Visual Basic.

Understanding the PeopleCode Language

In its fundamentals, PeopleCode syntax resembles that of other structured programming languages. Some aspects of the PeopleCode language, however, are specifically related to the PeopleTools environment. Definition name references, for example, enable you to refer to PeopleTools definitions, such as record definitions or pages, without using hard-coded string literals. Other language features, such as the PeopleCode data types and metastrings, reflect the close interaction of PeopleTools and SQL.

Data Types

The conventional data types available in previous releases are the core of PeopleCode functionality. The object data types are used to instantiate objects from the PeopleTools classes. The appropriate use of each data type is demonstrated where the documentation discusses PeopleCode that uses that data type.

PeopleSoft recommends that you declare your variables before you use them.

In this section, we discuss the following topics:

- Conventional data types.
- Object data types.

See Also

Variables

Conventional Data Types

- ANY

When variables and function return values are declared as ANY, the data type is indeterminate, enabling PeopleTools to determine the appropriate type of value based on context. Undeclared local variables are ANY by default.

- BOOLEAN
- DATE
- DATETIME
- FLOAT
- INTEGER

Note. The FLOAT and INTEGER data types should be used only where a performance analysis indicates that the increased speed is useful and an application analysis indicates that the different representations won't affect the results of the computations.

- NUMBER
- OBJECT
- STRING
- TIME

Object Data Types

For most classes in PeopleTools, you need a corresponding object data type to instantiate objects from that class.

See "Understanding Objects and Classes in PeopleCode."

Data Buffer Access Types

- Field
- Record
- Row

- Rowset

Page Display Types

- Grid
- GridColumn
- Page

Internet Script Types

- Cookie
- Request
- Response

Miscellaneous Object Types

- ASection
- Array
- File
- Interlink
- BIDocs

Note. BIDocs and Interlink objects used in PeopleCode programs run on the application server can only be declared as type Local. You can declare Interlinks as Global only in an Application Engine program.

- JavaObject

Note. JavaObject objects can only be declared as type Local.

- Message
- OptEngine
- ProcessRequest
- SoapDoc
- SQL
- SyncServer

- TransformData

Note. TransformData objects can only be declared as type Local.

- XmlDoc
- XmlNode

Note. XmlNode objects can only be declared as type Local.

API Object Type

- ApiObject

Use this data type for any API object, such as a session object, a tree object, a Component Interface, a PortalRegistry, and so on.

The following ApiObject data type objects can be declared as Global:

- Session
- PSMessages collection
- PSMessages
- All Tree classes (trees, tree structures, nodes, levels, and so on.)
- All Query classes

All other ApiObject data type objects *must* be declared as Local.

Considerations Using FLOAT, INTEGER, and NUMBER

The type INTEGER is a number represented as a 32 bit signed twos complement number, so it has a range of -2,147,483,648 to 2,147,483,647.

The type FLOAT is a number represented using the machine floating binary point (double precision) representation. This floating binary point representation is *not* appropriate for exact calculations involving decimal fractions—in particular calculations involving money. For example, because a tenth (1/10 or .1) cannot be exactly represented in floating binary point, a floating binary point sum of .10 + .10 will not be equal to .20. The default number representation in PeopleCode is a floating decimal point representation, which avoids this difficulty.

Operations (other than division) are done using integer arithmetic if the operands are both integer *and* the destination is integer, even if the variable is declared as NUMBER. The destination is considered to be integer if one of the following is true:

- It's an assignment to an integer variable or parameter.
- It's an array subscript.

- It's the right-hand operand of a comparison and the left-hand operand is an integer.
- It's a when-expression part of an evaluate statement and the expression evaluated at the start of the evaluate statement is integer.
- It's a for-loop initial, limit or step expression and the control variable of the for is integer.

Division (the / operator) is never performed using integer arithmetic. It's always performed using the floating-decimal-point arithmetic, even if the result variable is declared as INTEGER.

Comments

There are three ways to insert comments into your PeopleCode. You can surround comments with /* at the beginning and */ at the end. You can also use a **REM** (remark) statement for commenting. Put a semicolon at the end of a **REM** comment. If you do not, everything up to the end of the next statement will be treated as part of the comment.

If you want to nest comments, that is, enclose one set of comments with a another set, use the symbol <* at the start and *> at the end. You generally use this when you're testing code and want to comment out a section that already contains comments.

```
<* this program is no longer valid
commenting out entire thing REM This is an example of commenting PeopleCode;
/* ----- Logic for Compensation Change ----- */
/* Recalculate compensation change for next row. Next row is based on prior
value of EFFDT. */
calc_next_compchg(&OLDDT, EFFSEQ, 0);
/* Recalculate compensation change for current row and next row. Next row is
based on new value of EFFDT. */
calc_comp_change(EFFDT, EFFSEQ, COMP_FREQUENCY, COMPRATE, CHANGE_AMT,
CHANGE_PCT);
calc_next_compchg(EFFDT, EFFSEQ, 0);
*>
```

Note. All text between the <* and *> comment markers is scanned. If you have mismatched quotation marks, invalid assignments, and so on, you may receive an error when using these comments.

Use comments to explain, preferably in language comprehensible to anyone reading your program, what your code does. Comments also enable you to differentiate between PeopleCode delivered with the product and PeopleCode that you add or change. This differentiation helps in your analysis for debugging and upgrades.

Note. PeopleSoft strongly suggests that you use comments to place a *unique identifier* marking any changes or enhancements that you have made to a PeopleSoft application. This makes it possible for you to search for all the changes you have made. This is particularly helpful when you are upgrading a database.

Statements

A statement can be a declaration, an assignment, a program construct (such as a **Break** statement or a conditional loop), or a subroutine call.

In this section, we discuss the following topics:

- Separators.
- Assignment statements.
- Language constructs.
- Control statements.
- Conditional loops.

Separators

PeopleCode statements are generally terminated with a semicolon. The PeopleCode language accepts semicolons even if they're not required, such as after the last statement executed within an **If** statement. This enables you to consistently add semicolons after each statement.

Extra spaces and line breaks are ignored—and in fact are removed by the PeopleCode Editor when you save the code.

Assignment Statements

The assignment statement is the most basic type of statement in PeopleCode. It consists of an equal sign with a variable name on the left, and an expression on the right:

```
varname = expression;
```

The expression on the right is evaluated, and the result is placed in the variable named on the left. Depending on the data types involved, the assignment is passed either *by value* or *by reference*.

Assignment by Value

In most types of assignment, the result of the right-hand expression is assigned to the variable as a newly created value, in its own allocated memory area. Subsequent changes to the value of that variable have no effect on any other data.

Assignment by Reference

When both sides of an assignment statement are object variables, the result of the assignment is not a copy of the object. You're making a copy of the reference only. The variable names do not refer to separate and distinct objects; they both refer to the same object.

For example, both `&AN` and `&AN2` are arrays of number. Assigning `&AN2` to `&AN` does *not* make `&AN2` a distinct array. They still both point to the same information.

```
Local array of number &AN, &AN2;
Local number &NUM;

&AN = CreateArray(100, 200, 300);
&AN2 = &AN;
&NUM = &AN[1];
```

The assignment does not allocate any memory or copy any part of the original object. It makes `&AN2` refer to the same array to which `&AN` refers. Any changes you make to the value of either variable also affect the other. On the other hand, assigning `&NUM` to the first element in `&AN` (100), is *not* an object assignment. It's an assignment by value.

Note. In PeopleCode the equal sign can function as either an assignment operator or a comparison operator, depending on context.

Language Constructs

PeopleCode language constructs include:

- Branching structures: **If** and **Evaluate**.
- Loops and conditional loops: **For**, **Repeat**, and **While**.
- **Break** and **Exit** statements for escaping loops and terminating program execution.
- The **Return** statement for returning from functions.
- Variable and function declaration statements: **Global**, **Component**, **Local**, and **Declare Function**.
- The **Function** statement for defining functions.
- **Class** definitions statements.

- The **Try**, **Catch** and **Throw** statements for error handling.

Functions as Subroutines

PeopleCode, like C, does not have subroutines as we generally refer to them. PeopleCode subroutines are the subset of PeopleCode functions only that are defined to return no value or to return a value optionally. Calling a subroutine is the same as calling a function with no return value:

```
function_name([param_list]);
```

See Also

Control Statements

Functions

PeopleTools PeopleBook: PeopleCode Reference, "PeopleCode Built-in Functions", Function, Declare Function, CreateException, Try

Control Statements

These sections describe the following control statements, which control the flow of execution in a PeopleCode program.

Branching Statements

Branching statements control the flow of execution based on their evaluation of conditional expressions.

If...Then...Else

The syntax of the **If..Then...Else** statement is:

```
If condition Then
    [statement_list_1]
[Else
    [statement_list_2]]
End-if;
```

It evaluates the Boolean expression *condition*. If *condition* is TRUE, the **If** statement executes the statements in *statement_list_1*. If *condition* is FALSE, then it executes the statements in the **Else** clause; or, if there is no **Else** clause, it does nothing.

Evaluate

Use the **Evaluate** statement in cases where you want to check multiple conditions. Its syntax is:

```
Evaluate left_term
    When [relop_1] right_term_1
```

```

        [statement_list]
    .
    .
    .

    When [relop_n] right_term_n
        [statement_list]
    [When-other
        [statement_list]]
End-evaluate;

```

It takes an expression, *left_term*, and compares it to compatible expressions (*right_term*) using the relational operators (*relop*) in a sequence of **When** clauses. If *relop* is omitted, then = is assumed. If the result of the comparison is TRUE, it executes the statements in the **When** clause, then moves on to evaluate the comparison in the following **When** clause. It executes the statements in all of the **When** clauses for which the comparison evaluates to TRUE. If and only if none of the **When** comparisons evaluates to TRUE, it executes the statement in the **When-other** clause (if one is provided). For example, the following **Evaluate** statement executes only the first **When** clause. &USE_FREQUENCY in this example can only have one of three string values:

```

evaluate &USE_FREQUENCY
when = "never"
    PROD_USE_FREQ = 0;
when = "sometimes"
    PROD_USE_FREQ = 1;
when = "frequently"
    PROD_USE_FREQ = 2;
when-other
    Error "Unexpected value assigned to &USE_FREQUENCY."
end-evaluate;

```

To end the **Evaluate** after the execution of a **When** clause, you can add a **Break** statement at the end of the clause:

```

evaluate &USE_FREQUENCY
when = "never"
    PROD_USE_FREQ = 0;
    Break;
when = "sometimes"
    PROD_USE_FREQ = 1;
    Break;
when = "frequently"
    PROD_USE_FREQ = 2;
    Break;
when-other
    Error "Unexpected value assigned to &USE_FREQUENCY."
end-evaluate;

```

In some rare cases you may want to make it possible for more than one of the **When** clauses to execute:

```

evaluate &PURCHASE_AMT
when >= 100000
    BASE_DISCOUNT = "Y";
when >= 250000
    SPECIAL_SERVICES = "Y";
when >= 1000000

```

```
MUST_GROVEL = "Y";
end-evaluate;
```

For Loops

The **For** statement repeats a sequence of statements a specified number of times. Its syntax is:

```
For count = expression1 to expression2
    [Step i];
    statement_list
End-for;
```

The **For** statement initializes the value of *count* to *expression1*, then increments *count* by *i* each time after it executes the statements in *statement_list*. It continues until *count* is equal to *expression2*. If the **Step** clause is omitted, then *i* equals one. If you want to count backwards from a higher value to a lower value, then use a negative value for *i*. You can exit a **For** loop using a **Break** statement.

The following example demonstrates the **For** statement:

```
&MAX = 10;
for &COUNT = 1 to &MAX;
    WinMessage("Executing statement list, count = " | &COUNT);
end-for;
```

See Also

PeopleTools PeopleBook: PeopleCode Reference, "PeopleCode Built-in Functions," If, Evaluate, For

Conditional Loops

Conditional loops, **Repeat** and **While**, repeat a sequence of statements, evaluating a conditional expression each time through the loop. The loop terminates when the condition evaluates to True. You can exit from a conditional loop using a **Break** statement. If the **Break** statement is in a loop embedded in another loop, the break applies only to the inside loop.

Repeat

The syntax of the **Repeat** statement is:

```
Repeat
    statement_list
Until logical_expression;
```

The **Repeat** statement executes the statements in *statement_list* once, then evaluates *logical_expression*. If *logical_expression* is False the sequence of statements is repeated until *logical_expression* is True.

While

The syntax of the **While** statement is:

```

While logical_expression
    statement_list
End-while;

```

The While statement evaluates `logical_expression` before executing the statements in ***statement_list***. It continues to repeat the sequence of statements until ***logical_expression*** evaluates to False.

See Also

PeopleTools PeopleBook: PeopleCode Reference, "PeopleCode Built-in Functions," Repeat, While

Functions

PeopleCode supports the following types of functions:

Built-in	The standard set of PeopleCode functions described in PeopleCode Built-in Functions. These can be called without being declared.
Internal	Functions that are defined (using the Function statement) within the PeopleCode program in which they are called.
External PeopleCode	PeopleCode functions defined outside the calling program—generally contained in record definitions serving as function libraries.
External non-PeopleCode	Functions stored in external (C-callable) libraries.

In addition, PeopleCode also supports methods. The main difference between a built-in function and a method is:

- A built-in function, in your code, is on a line by itself, and does not (generally) have any dependencies. You do not have to instantiate an object before you can use the function.
- A method can only be executed from an object, using dot notation. You have to instantiate the object first.

In this section, we discuss the following topics:

- Defining functions.
- Declaring functions.
- Calling functions.
- Function return values.
- How to avoid function naming conflicts.

See Also

"Understanding Objects and Classes in PeopleCode"

PeopleTools PeopleBook: PeopleCode Reference, "PeopleCode Built-in Functions," Function, Declare Function

Defining Functions

PeopleCode functions can be defined in any PeopleCode program. Function definitions must be placed at the top of the program, along with any variable and external function declarations.

By convention, PeopleCode programs are stored in records whose names begin in FUNCLIB_, and they are always attached to the FieldFormula event.

See Also

PeopleTools PeopleBook: PeopleCode Reference, "PeopleCode Built-in Functions", Function

Declaring Functions

If you call an external function (that is, a function defined outside the program where it is called) from a PeopleCode program, you must declare the function at the top of the program. The syntax of the function declaration varies, depending on whether the external function is written in PeopleCode or compiled in a dynamic link library.

See Also

PeopleTools PeopleBook: PeopleCode Reference, "PeopleCode Built-in Functions", Declare Function

Calling Functions

Functions are called with this syntax:

```
function_name([param_list])
```

The parameter list (***param_list***), is a list of expressions, separated by commas, that the function expects you to supply. Items in the parameter list can be optional, or required.

You can check the values of parameters that get passed to functions at runtime in the Parameter window of the PeopleCode debugger.

If the return value is required, then the function must be called as an expression, for example:

```
&RESULT = Product (&RAISE_PERCENT, .01, EMPL_SALARY);
```

If the function has an optional return value it can be called as a subroutine. If the function has no return value, it *must* be called as a subroutine:

```
WinMessage(64, "I can't do that, " | &OPER_NICKNAME | ".");
```

Parameters are always passed to internal and external *PeopleCode functions* by reference. If the function is supposed to be effecting a change in the data the caller passes, you must also pass in a variable.

Built-in function parameters can be passed by reference or by value, depending on the function. External C function parameters can be passed by value or by reference, depending on the declaration and type.

Note. A built-in function is a function provided by PeopleCode, such as RevalidatePassword, AddAttachment, and so on. A PeopleCode function is one that you write yourself, or is written outside the PeopleSoft system.

See Also

"Debugging Your Application"

Passing Variables to Functions

Function Return Values

Functions can return values of any supported data type; and some functions do not return any value.

Optional return values occur only in built-in functions—you cannot define a function that optionally returns a value. Optional return values are typical in functions that return a Boolean value indicating whether execution was successful. For example, the following call to **DeleteRow** ignores the Boolean return value and deletes a row:

```
DeleteRow(RECORD.BUS_EXPENSE_PER, &L1_ROW, RECORD.BUS_EXPENSE_DTL, &L2_ROW);
```

The following example checks the return value and returns a message saying whether it succeeded:

```
if DeleteRow(RECORD.BUS_EXPENSE_PER, &L1_ROW, RECORD.BUS_EXPENSE_DTL, &L2_ROW)
then
    WinMessage("Row deleted.");
else
    WinMessage("Sorry -- couldn't delete that row.");
end-if;
```

Avoiding Function Naming Conflicts

If you define a function with the same name as a built-in function, the function that you defined takes precedence over the built-in function. That is, the user-defined function is used in the PeopleCode program, *not* the built-in function.

Anytime you compile PeopleCode in the PeopleCode Editor (that is, whenever you save a PeopleCode program) a warning message displays in the Validate tab, indicating that a user-defined function has the same name as an existing built-in function.

In addition, if you select Compile All PeopleCode, an error message is generated in the log file for every user-defined function that has the same name as a built-in function.

The following is an example error message:

```
User defined function IsNumber is overriding the builtin function of the same
name. (2,98)
```

If you notice that you named a function the same as a built-in function, and that the built-in function does what you're trying to achieve, you should replace your function with a reference to the built-in. Why? The built-in is probably more efficient. In addition, it also reduces the confusion for people who have to maintain your code, because if they miss the warning message in the Validate tab, they might assume the built-in is being called when in fact it is a local function of the same name.

Expressions

Expressions evaluate to values of any of the PeopleCode data types. A simple PeopleCode expression can consist of a constant, a temporary variable, a system variable, a record field reference, or a function call. Simple expressions can be modified by unary operators (such as a negative sign or logical NOT), or combined into compound expressions using binary operators (such as a plus sign or logical AND).

Definition name references evaluate to strings equal to the name of a PeopleTools definition, such as a record, page, or message. They enable you to refer to definitions without using string literals, which are difficult to maintain.

Metastrings (also called meta-SQL) are special expressions used within SQL string literals. At runtime the metastrings expand into the appropriate SQL for the current database platform.

In this section, we discuss the following topics:

- Constants.
- Functions as expressions.
- Variables.
- System variables.
- Metastrings.
- Record field references.
- Definition name references.

Constants

PeopleCode supports numeric, string, and Boolean constants, as well as user-defined constants. It also supports the constant Null, which indicates an object reference that does not refer to a valid object.

Note. You can express DATE, DATETIME, and TIME values by converting from STRING and NUMBER constants using the Date, Date3, DateTime6, DateTimeValue, DateValue, Time3, TimePart, and the TimeValue functions. You can also format a DATETIME value as text using FormatDateTime.

Numeric

Numeric constants can be any decimal number. Some examples are:

```
7
0.8725
-172.0036
```

String Constants

String constants can be delimited by using either single (') or double (") quote marks. If a quote mark occurs as part of a string, the string can be surrounded by the other delimiter type. As an alternative, you can include the delimiter twice. Some examples are:

```
"This is a string constant."
'So is this.'
'She said, "This is a string constant."'
'She said, ""This is a string constant.""'
```

Use the following code if you need to include a literal quotation mark as part of your string:

```
&cDblQuote = '''; /* singlequote doublequote singlequote */
```

The following also produces a string with a single double quotation mark in it:

```
&cDblQuote = """"; /* dquote dquote dquote dquote */
```

You can also directly embed the doubled double quote in the desired strings, such as:

```
&sImage = Char(10) | '<IMG SRC="%IMAGE(' | &pImageName | ')"' ;
```

Boolean Constants

Boolean constants represent a truth value. The two possible values are:

```
True
False
```

Null Constant

Null constants represent an object reference value that does not refer to a valid object. This means that calling a method on the object or trying to get or set a property of it fails. The Null constant is just the keyword:

```
Null
```

User-Defined Constants

You can define constants at the start of a PeopleCode program. Then you can use the declared constant anywhere that the corresponding value would be allowed. Constants can be defined as numbers, strings, or Boolean values.

User-defined constants can only be local, that is, you cannot declare them as Global or Component.

The following is an example of user-defined constant declarations:

```
Constant &Start_New_Instance = True;
Constant &Display_Mode = 0;
Constant &AddMode = "A":
Local Field &Start_Date;
. . .
MyFunction(&Start_New_Instance, &Display_Mode, &Add_Mode);
```

See Also

PeopleTools PeopleBook: PeopleCode Reference, "PeopleCode Built-in Functions", Date, Date3, DateTime6, DateTimeValue, DateValue, Time3, TimePart, TimeValue, FormatDateTime

Functions as Expressions

You can use any function that returns a value as an expression. It can be used on the right side of an assignment statement, passed as a parameter to another function, or combined with other expressions to form a compound expression.

See Also

PeopleTools PeopleBook: PeopleCode Reference, "PeopleCode Built-in Functions," Functions

Variables

There are the following types of variables available in your program:

- System variables

System variables provide access to system information. System variables have a prefix of the '%' character, rather than the '&' character. Use these variables wherever you can use a constant, passing them as parameters to functions or assigning their values to fields or to temporary variables.

- User-defined variables

These variable names are preceded by an "&" character wherever they appear in a program. Variable names can be 1 to 1000 characters, consisting of letters A-Z and a-z, digits 0-9, and characters #, @, \$, and _.

User-Defined Variable Declaration and Scope

The difference between the variable declarations concerns their life spans:

Global	Valid for the entire session.
Component	Valid while any page in the component in which it's defined stays active.
Local	Valid for the life of the PeopleCode program or function in which it's defined.

You can declare variables using the **Global**, **Local**, or **Component** statement, or you can use local variables without declaring them. Here are some examples:

```
Local Number &AGE;
Global String &OPER_NICKNAME;
Component Rowset &MY_ROWSET;
Local Any &SOME_FIELD;
Local ApiObject &MYTREE;
Local Boolean &Compare = True;
```

Variable declarations are usually placed above the main body of a PeopleCode program (along with function declarations and definitions). The exception is the Local declaration, which you can use within a function or the main section of a program. You can declare variables as any of the PeopleCode data types. If a variable is declared as an ANY data type, or if a variable is not declared, PeopleTools uses an appropriate data type based on context.

Note. PeopleSoft recommends that you declare a variable as an explicit data type unless the variable is going to hold a value of an unknown data type.

Global variables remain defined and keep their values throughout a PeopleSoft session and can be accessed from different components and applications, including an Application Engine program. A Global variable must be declared, however, in each PeopleCode program where it's used. PeopleSoft recommends you use Global variables rarely, because they are difficult to maintain.

Global variables are not available to a portal or applications on separate databases. They are available only on applications and Portals in the *same* database.

Component variables provide an intermediate level between the Global and Local scope. They remain defined and keep their values while any page in the component in which they're defined remains active. Similar to a Global variable, a Component variable must be declared in each PeopleCode program where it's used.

Component variables act the same as Global variables when an Application Engine program is called from a page (using **CallAppEngine**.)

Component variables remain defined after a **TransferPage**, **DoModal**, or **DoModalComponent** function. However, variables declared as Component do *not* remain defined after using the **Transfer** function, whether you're transferring within the same component or not.

Local variables declared at the top of a PeopleCode program (or within the main, that is, non-function, part of a program) remain in scope for the life of that PeopleCode program. Local variables declared within a function are valid to the end of the function and not beyond.

You can check the values of your Local, Global, and Component variables at runtime in the different variable windows of the PeopleCode debugger. Local variables declared within a function are displayed in the Function Parameters window.

Considerations Using Local Variables

A local variable declared in the main (that is, non-function) part of a program can be affected by statements anywhere in the program. For example, suppose there are two functions in RECORD_A.FIELD_A.FieldFormula, FUNC_1 and FUNC_2, and both modify a local variable named &TEMP. They could affect each other, as they're both using the same variable name in the same PeopleCode program.

If, however, FUNC_3 is defined in RECORD_B.FIELD_B.FieldFormula and makes reference to &TEMP, it is not the same &TEMP as in RECORD_A.FIELD_A.FieldFormula. This becomes important when FUNC_1 calls FUNC_3. Technically both functions exist at the same time, one inside the other, but &TEMP is a different variable for each of them. However if FUNC_1 calls FUNC_2 then &TEMP is the same variable for both.

Lifetime of Local Variables

A local variable is valid for the life of the PeopleCode program or function in which it's defined. However, what exactly is the life of a PeopleCode program?

A PeopleCode program is defined as what the PeopleCode Editor in Application Designer presents in a single window: a chunk of PeopleCode text associated with a single item (that is, a record field event, a component record event, and so on.)

When the system evaluates a PeopleCode program and calls a function in the same PeopleCode program, a new program evaluation is *not* started.

However, when a function from a *different* PeopleCode program is called (that is, some PeopleCode text associated with a different item), the current PeopleCode program is suspended and it starts evaluating the new program. This means that any local variables in the calling program (called A) are *no longer available*. Those in the called program (called B) are available for use.

Even if the local variables in the A program have the same name as those in the B program, they are different variables and are stored separately.

What happens if the called program (B) in turn calls a function in program A? A new set of program A's variables are allocated and the called function in A uses these new variables. Thus, this second use of program A gets another lifetime, until it returns to program B.

The following is an example of pseudo-code to show how this might work. (This is non-compiled, non-working code. If you wanted to use this example, you'd have to enter one without the external declaration of the function in the other, not yet compiled, one.)

```

Program A (Rec.Field.FieldChange):
  local number &temp;
  declare function B1 PeopleCode Rec.Field FieldFormula;
  /* Uncomment this declaration and comment above to compile this the first
  time.
      function B1
      end-function;
  */

  function A1
  WinMessage("A1: &temp is " | &temp);
  &temp = &temp + 1;
  A2();
  B1();
  A2();
  end-function;

  function A2
  WinMessage("A2: &temp is " | &temp);
  &temp = &temp + 1;
  end-function;

  A1();

Program B (Rec.Field.FieldFormula):
  local number &temp;
  declare function A2 PeopleCode Rec.Field FieldChange;

  function B1
  WinMessage("B1: &temp is " | &temp);
  &temp = &temp + 1;
  A2();
  end-function;

```

When this is compiled and run, it produces the following output:

```
A1: &temp is 0
```

```

A2: &temp is 1

B1: &temp is 0

A2: &temp is 0

A2: &temp is 2

```

Initializing User-Defined Variables in Declarations

If you want to declare and initialize your variables in one step, you can use the following format:

```

Local String &MyString = "New";

Local Date &MyDate = %Date;

```

This is available only for variables with the scope of Local.

Though you can declare more than one variable on a single line, you can only initialize one variable on a line. The following code creates a syntax error when you try to save the program:

```

Local Number &N1, &N2 = 5;

```

You cannot declare a variable, then initialize it in a second declaration statement. The following produces a duplicate declaration error when you try to save the program:

```

Global Number &N1;

...

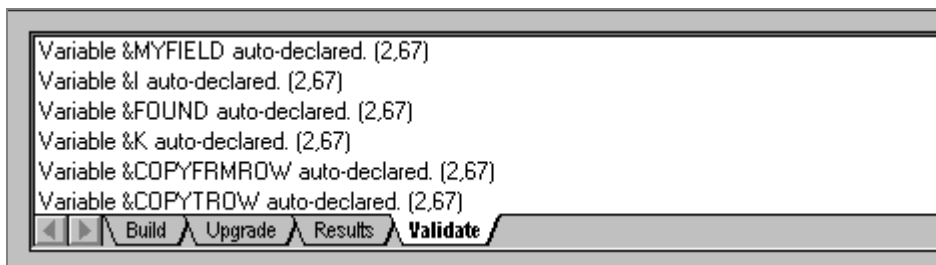
Local String &N1 = "Str"; /* Duplicate definition. */

```

If you do not initialize variables, either when you declare them or before you use them, strings are initialized as null strings, dates and times as nulls, and numbers as zero.

Declaring Variables

PeopleSoft recommends that you declare variables before you use them. If you do not declare a variable, it's automatically declared with the scope Local and the data type Any. You receive a warning message in the Validation tab for every variable that is not declared when you save the PeopleCode program.



Validate tab with auto-declared variables

If you've declared all the variables, you can use these values to ensure you do not have any typos or misspellings. For example, if you declared a variable as `&END_DATE`, then accidentally spell it as `&EDN_DATE`, the "new variable" displays on the Validate tab when you save the program.

Another reason to declare your variables is for the design time checking. If you declare a variable of one data type, then assign to it a value of a different type, the PeopleCode Editor will catch that assignment as a design time error when you try to save the program. With an undeclared variable, the assignment error won't appear until runtime.

The following example produces a design time error when you try to save the program:

```
Local Field &DATE;

&DATE = GetRecord(RECORD.DERIVED_HR);
```

In addition, if you declare variables, Find Object Reference finds imbedded definitions. For example, suppose you wanted to find all occurrences of the field `DEPT_ID`. If you have not declared `&MyRecord` as a record, Find Object References will not find the following reference of the field `DEPT_ID`:

```
&MyRecord.DEPT_ID.Visible = False;
```

Recursive Functions

PeopleCode supports true recursive functions. A function can call itself, and each possibly recursive call of the function has its own independent copy of the parameters and function-local variables.

When writing recursive functions, be careful about passing variables as parameters, because PeopleCode does such calls by reference. This means that if you call a function such as:

```
Function Func(&n as Number)
    &n = 3;
End-Function;

local &x = 5;

Func (&x) ;
```

After the call to `Func(&x)`, `&x` will have the value 3, not 5. If the call was `Func(Value(&x))`, after the call `&x` is still 5.

Restrictions on Variable Use

The following data types can only be declared as Local. They *cannot* be declared as either Global or Component:

- `JavaObject`
- `Interlink`

Note. Interlink objects can be declared as type Global in an Application Engine program.

- TransformData
- XmlNode

The following ApiObject data type objects can be declared as Global:

- Session
- PSMessages collection
- PSMMessage
- All Tree classes (trees, tree structures, nodes, levels, and so on.)
- All Query classes

All other ApiObject data type objects *must* be declared as Local.

Using User-Defined Variables

A user-defined variable can hold the contents of a record field for program code clarity. For example, you may give a variable a more descriptive name than a record field, based on the context of the program. If the record field is from another record, you may assign it to a temporary variable rather than always using the record field reference. This makes it easier to enter the program, and can also make it easier to read.

Also, if you find yourself calling the same function repeatedly to get a value, you may be able to avoid some processing by calling the function once and placing the result in a variable.

Passing Variables to Functions

PeopleCode variables are always passed to functions by reference. This means, among other things, that the function can change the value of a variable passed to it so that the variable will have the new value on return to the calling routine. For example, the **Amortize** built-in function expects you to pass it variables into which it will place the amount of a loan payment applied towards interest (&PYMNT_INTRST), the amount of the payment applied towards principle (&PYMNT_PRIN), and the remaining balance (&BAL). It calculates these values based on information that the calling routine supplies in other parameters:

```
&INTRST_RT=12;
&PRSNT_BAL=100;
&PYMNT_AMNT=50;
&PYMNT_NBR=1;
Amortize(&INTRST_RT, &PRSNT_BAL, &PYMNT_AMNT, &PYMNT_NBR, &PYMNT_INTRST,
&PYMNT_PRIN, &BAL);
&RESULT = "Int=" | String(&PYMNT_INTRST) | " Prin=" | String(&PYMNT_PRIN) | "
          Bal=" | String(&BAL);
```

State of Shared Objects Using PIA

This section uses the following scenario:

- A local and a global variable refer to the same object.
- That object is used in a modal component.
- Instead of completing the modal component, the user clicks the browser's "Back" button.

In general, the global state of the object is restored. If the object hasn't been destroyed from the global state, the global state of the object is used for local references, otherwise, the local state is used for local references.

Here's an example:

```
Global array of number &Global_Array;
Local array of number &Local_Array;

&Global_Array = CreateArray(1, 2, 3);
&Local_Array = &Global_Array
DoModal(Page.PAGENAME, "", -1, -1, 1, Record.SHAREDREC, 1);
/* return to here */
&Local_Array[1] = -1;
&Global_Array[2] = -2;
WinMessage(&Local_Array is " | &Local_Array.Join());
WinMessage(&Global_Array is " | &Global_Array.Join());
```

The following program, program 2, is located on the modal page the user is transferred to:

```
Global array of number &Global_Array;

&Global_Array[3] = -3;
```

The following program, program 3, is also located on the modal page:

```
Global array of number &Global_Array;

&Global_Array = CreateArray(1, 2, -3);
```

If program 2 is run, the output is the following:

&Local_Array is -1, -2, -3

&Global_Array is -1, -2, -3

However, if program 3 is run, thereby destroying the original global state, the output is the following:

&Local_Array is -1, 2, 3

&Global_Array is 1, -2, -3

See Also

PeopleTools PeopleBook: PeopleCode Reference, "System Variables"
 "Debugging Your Application"

System Variables

System variables are preceded by a percent (%) symbol whenever they appear in a program. Use these variables to get the current date and time, information about the user, the current language, the current record, page, or component, and more.

See Also

PeopleTools PeopleBook: PeopleCode Reference, "System Variables"

Metastrings

Metastrings are special SQL expressions. The metastrings, also called meta-SQL, are preceded with a percent (%) symbol, and can be included directly in the string literals. They expand at runtime into an appropriate substring for the current database platform. Meta-SQL are used in functions that pass SQL strings, that is,

- SQLExec
- the scroll buffer functions (ScrollSelect and its relatives)
- in Application Designer to construct dynamic views
- with some rowset object methods (Select, SelectNew, Fill, and so on.)
- with SQL objects
- in Application Engine
- with some record object methods (Insert, Update, and so on.)
- with COBOL

See Also

PeopleTools PeopleBook: PeopleCode Reference, "PeopleCode Built-in Functions",
 SQLExec, ScrollSelect

PeopleTools PeopleBook: PeopleCode Reference, "Meta-SQL"

Record Field References

Use record field references to retrieve the value stored in a record field, or to assign a value to a record field.

Record Field Reference Syntax

References to record fields have the following form:

```
[recordname.]fieldname
```

You must supply the **recordname** only if the record field and your PeopleCode program are in different record definitions.

For example, suppose that in a database for veterinarians you have two records, PET_OWNER and PET. A program in the record definition PET_OWNER must refer to the PET_BREED record field in the PET record definition as:

```
PET.PET_BREED
```

However, a program in the PET record definition can refer to this same record field more directly as :

```
PET_BREED
```

And, if the program is in the PET_BREED record field itself, it can refer to this record field using the caret (^) symbol:

```
^
```

The PeopleCode Editor replaces the caret symbol with the actual record field name.

You can also use object dot notation to refer to record fields, for example:

```
&FIELD = GetRecord(RECORD.PET_OWNER).GetField(FIELD.PET_BREED);
```

See Also

"Referencing Data in the Component Buffer"

Legal Record Field Names

A record field name consists of two parts, the record name and the field name, separated by a period.

The field names used in PeopleCode are consistent with the field names allowed in the field definition. Case is ignored, although the PeopleCode Editor will, for the sake of convention, automatically format field names in uppercase. A field name can be 1 to 18 characters, consisting of alphanumeric characters determined by your current language setting in Windows, and characters #, @, \$, and _.

A record name can be 1 to 15 characters, consisting of alphanumeric letters determined by your current language setting in Windows, and characters #, @, \$, and _.

Definition Name References

Definition name references are special expressions that reference the name of a PeopleTools definition, such as a record, page, component, Business Interlink, and so on. Syntactically, a definition name reference consists of a reserved word indicating the type of definition, followed by a period, then the name of the PeopleTools definition. For example, the following definition name reference:

```
RECORD.BUS_EXPENSE_PER
```

refers to the definition name:

```
BUS_EXPENSE_PER
```

Generally, definition name references are passed as parameters to functions. If you attempt to pass a string literal instead of a definition name reference to such a function, you receive a syntax error.

You also use definition name references outside function parameter lists, for example in comparisons:

```
If (%Page = PAGE.SOMEPAGE) Then
    /* do stuff specific to SOMEPAGE */
End-If;
```

In these cases the definition name reference evaluates to a string literal. Using the definition name reference instead of a string literal enables PeopleTools to maintain the code if the definition name changes.

If you use the definition name reference, and the name of the definition changes, the change automatically "ripples" through the code, so you do not have to change it or maintain it.

In the PeopleCode Editor, if you place your cursor over any definition name reference and right-click, you can select View Definition from the pop-up menu to open the definition.

In addition, for most definitions, if you specify an invalid definition name in the PeopleCode program, that is, a definition that hasn't been created in Application Designer, you receive an error message when you try to save your program.

Legal and Illegal Definition Names

Legal definition names, as far as definition name references are concerned, consist of alphanumeric letters determined by your current language setting in Windows, and the characters #, @, \$, and _.

In some cases, however, the definition supports the use of other characters. You can, for example, have a menu item named "A&M" stored in the menu definition even though "&" is an illegal character in the definition name reference. The illegal character results in an error when you validate the syntax or attempt to save the PeopleCode.

You can avoid this problem in two ways:

- Rename the definition so that it uses only legal characters.

- Enclose the name of the definition in quotes in the reference, for example:

```
ITEMNAME."A&M"
```

The second solution is a commonly used workaround in cases where the definition name contains illegal characters. If you use this notation, the definition name reference is not treated as a string literal: PeopleTools maintains the reference the same way as it does other definition name references.

Note. If your definition name begins with a number, you must enclose the name in quotation marks when you use it in a definition name reference. For example, `CompIntfc."1_DISCIPLIN_ACTN"`.

Reserved Word Summary Table

The following table summarizes the reserved words used in definition name references.

<i>Reserved Word</i>	<i>Common Usage</i>
BARNAME	Used with transfers and modal transfers.
BUSACTIVITY	Used with TriggerBusinessEvent.
BUSEVENT	Used with TriggerBusinessEvent.
BUSPROCESS	Used with TriggerBusinessEvent.
COMPINTFC	Used with Component Interface Classes.
COMPONENT	Used with transfers and modal transfers, as well as for generating URLs.
FIELD	Used with methods and functions to designate a field.
FILELAYOUT	Used with the SetFileLayout File class method.
HTML	Used with the GetHTMLText function.
IMAGE	Used in with functions and methods to designate an image.
INTERLINK	Used with the GetInterlink function.
ITEMNAME	Used with transfers and modal transfers.
MENUNAME	Used with transfers and modal transfers.
MESSAGE	Used with Messaging functions and methods.
NODE	Used with transfers and modal transfers, as well as generating URLs.
PAGE	Used with transfers and modal transfers to pass the page item name (instead of the page name), and with controls and other functions to pass the page name.
PORTAL	Used with transfers and modal transfers, as well as generating URLs.

<i>Reserved Word</i>	<i>Common Usage</i>
RECORD	Used in functions and methods to designate a record.
SCROLL	The name of the scroll area in the page. This name is always equal to the primary record of the scroll.
SQL	Used with SQL definitions.
STYLESHEET	Used with style sheets.

See Also

PeopleTools PeopleBook: PeopleCode Reference, "PeopleCode Built-in Functions", TriggerBusinessEvent, GetHTMLText, GetInterlink

"Component Interface Classes"

"Message Class"

PeopleTools PeopleBook: PeopleCode Reference, "File Classes," SetFileLayout

"SQL Class"

Operators

PeopleCode expressions can be modified and combined using math, string, comparison, and Boolean operators.

In this section, we discuss the following topics:

- Math operators.
- Operations on dates and times.
- String concatenation.
- @ operator.
- Comparison operators.
- Boolean operators.

Math Operators

PeopleCode uses standard mathematical operators:

- + add
- subtract (or unary negative sign)
- * multiply

/ divide

** exponentiation

Exponentiation occurs before multiplication and division; multiplication and division occur before addition and subtraction. Math expressions otherwise are evaluated from left to right. You can use parentheses to force the order of operator precedence.

The minus sign can also, of course, be used as a negation operator, as in the following expressions:

```
-10
- &NUM
- Product(&PERCENT_CUT, .01, SALARY)
```

Operations on Dates and Times

You can add or subtract two date values or two time values, which provides an NUMBER result. In the case of dates, the number represents the difference between the two dates in days. In the case of time, the number represents the difference in seconds. You can also add and subtract numbers to or from a time or date, which results in another date or time. Again, in the case of days the number represents days, and in the case of time the number represents seconds.

The following table summarizes these operations:

<i>Operation</i>	<i>Result Type</i>	<i>Number Represents...</i>
time + number	time	number of seconds
date + number	date	number of days
date - date	number	difference in days
time - time	number	difference in seconds
date + time	datetime	

String Concatenation

The string concatenation operator (||) is used to combine strings. For example, assuming &OPER_NICKNAME is "Dave", the following statement sets &RETORT to "I can't do that, Dave."

```
&RETORT = "I can't do that, " || &OPER_NICKNAME || "."
```

The concatenation operator automatically converts its operands to strings. This makes it easy to write statements that display mixed data types. For example:

```
&DAYS_LEFT = &CHRISTMAS - %Date;
WinMessage("Today is " || %Date || ". Only " || &DAYS_LEFT || " shopping days left!");
```

@ Operator

The @ operator converts a string storing a definition reference into the definition. This is useful, for example, if you want to store definition references in the database as strings and retrieve them for use in PeopleCode; or if you want to obtain a definition reference in the form of a string from the operator using the **Prompt** function.

To take a simple example, if the record field EMPLID is currently equal to "8001", the following expression would evaluate to "8001":

```
@ "EMPLID"
```

The following example uses the @ operator to convert strings storing a record reference and a record field reference:

```
&STR1 = "RECORD.BUS_EXPENSE_PER";
&STR2 = "BUS_EXPENSE_DTL.EMPLID";
&STR3 = FetchValue(@(&STR1), CurrentRowNumber(1), @(&STR2), 1);
WinMessage(&STR3, 64);
```

Note. String literals that reference definitions are not maintained by PeopleTools. Remember that if you store definition references as strings, then convert them with the @ operator in the code, this will create maintenance problems whenever the definition names change.

The following function takes a rowset and a record, passed in from another program, and performs some processing. The **GetRecord** method won't take a variable for the record, however, using the @ symbol you can dereference the record name. Because the record name is never hard-coded as a string, if the record name changes, this code will not have to change.

```
Function Get_My_Row(&PASSED_ROWSET, &PASSED_RECORD)

    For &ROWSET_ROW = 1 To &PASSED_ROWSET.RowCount
        &UNDERLYINGREC = "RECORD." | &PASSED_ROWSET.DBRecordName;
        &ROW_RECORD =
        &PASSED_ROWSET.GetRow(&ROWSET_ROW).GetRecord(@&UNDERLYINGREC);

        /* Do other processing */

    End-For;

End-Function;
```

Comparison Operators

Comparison operators are used to compare two expressions of the same data type. The result of the comparison is a Boolean value. The following table summarizes these operators:

Operator	Meaning
=	equal
!=	not equal
<>	not equal
<	less than
<=	less than or equal to
>	greater than
>=	greater than or equal to

You can precede any of the comparison operators with **Not**, for example:

- Not =
- Not <
- Not >=

Expressions formed with comparison operators form *logical terms* that can be combined using Boolean operators.

String comparisons are case-sensitive. You can use the Upper or Lower built-in functions to do a case-insensitive comparison.

See Also

Upper

Lower

Boolean Operators

The logical operators AND, OR, and NOT are used to combine Boolean expressions. The following table shows the results of combining two Boolean expressions with AND and OR operators:

Exp1	Operator	Exp2	Result
FALSE	AND	FALSE	FALSE
FALSE	AND	TRUE	FALSE
TRUE	AND	TRUE	TRUE
FALSE	OR	FALSE	FALSE
FALSE	OR	TRUE	TRUE
TRUE	OR	TRUE	TRUE

The NOT operator negates Boolean expressions, changing a TRUE value to FALSE and a FALSE value to TRUE.

In complex logical expressions using the operations AND, OR, and NOT, NOT takes the highest precedence, AND is next, and OR is lowest. Use parentheses to override precedence. (It's generally a good idea to use parentheses in logical expressions anyway, because it makes them easier to decipher.) If used on the right side of an assignment statement, Boolean expressions must be enclosed in parentheses.

The following are examples of statements containing Boolean expressions:

```
&FLAG = (Not (&FLAG)); /* toggles a Boolean */

if ((&HAS_FLEAS or &HAS_TICKS) and
    SOAP_QTY <= MIN_SOAP_QTY) then
    SOAP_QTY = SOAP_QTY + OrderFleaSoap(SOAP_ORDER_QTY);
end-if;
```

CHAPTER 7

Understanding Objects and Classes in PeopleCode

This chapter provides an overview of classes and objects and discusses how to:

- Work with objects.
- Assign objects.
- Pass objects.

Understanding Classes and Objects

PeopleSoft delivers classes of objects that you can manipulate with PeopleCode. In addition, you can extend the existing classes, or create your own. The delivered classes may or may not have a GUI equivalent; some are representations of data structures that occur only at runtime. With PeopleCode you can manipulate data in the data buffer easily and consistently. These classes enable you to write code that's more readable, more easily maintained, and more useful.

In this section, we discuss the following topics:

- What is a class?
- What is an object?
- What is instantiating objects?

What is a Class?

A *class* is the formal definition of an object and acts as a template from which an instance of an object is created at runtime. The class defines the properties of the object and the methods used to control the object's behavior.

PeopleSoft delivers predefined classes (such as Array, File, Field, SQL, and so on.) You can create your own classes using the Application Class. You can also extend the functionality of the existing classes using the Application Class.

See Also

PeopleTools PeopleBook: PeopleCode Reference, "Application Classes"

What is an Object?

An *object* represents a unique instance of a data structure defined by the template provided by its class. Each object has its own values for the variables belonging to its class and responds to methods defined by that class. This is the same for classes provided by PeopleSoft and for classes you create yourself.

The class is a template, or blueprint, from which you create an object. After an object has been created (instantiated) from a class, you can change its properties. A *property* is an attribute of an object. Properties define the following:

- Object characteristics, such as name or value.
- The state of an object, such as deleted or changed.

Some properties are read-only and cannot be set, such as Name or Author. Other properties are read-write and can be set, such as Value or Label.

Objects are different from other data structures. They contain code (in the form of *methods*), not just static data. A *method* is a procedure or routine, associated with one or more classes, that acts on an object.

An analogy to illustrate the difference between an object and its class is the difference between a car and the blue Renault Citroen with license plate number TS5800B. A class is a general category, while the object is a very specific instance of that class. Each car comes with standard characteristics, such as four wheels, an engine, or brakes, that define the class and are the template from which the individual car is created. You can change the properties of an individual car by personalizing it with bumper stickers or racing stripes, which is like changing the Name or Visible property of an object. The model and date it's created are similar to read-only properties because you cannot alter them. A tune-up acts on the individual car and changes its behavior, much as a method acts on an object.

What is Instantiating Objects?

A class is the blueprint for something, like a bicycle, a car, or a data structure. An object is the actual thing that's built using that class (or blueprint.) From the blueprint for a bicycle, you can build a specific mountain bike with 23 gears and tight suspension. From the blueprint of a data structure class, you build a specific instance of that class. *Instantiation* is the term for building that copy, or an instance, of a class.

Working With Objects

Generally you instantiate an object (create them from their class) using built-in functions or methods of other objects. Some objects are instantiated from data already existing in the data

buffer. Think about this kind of object instantiation as taking a chunk of data from the buffer, encapsulating it in code (methods and properties), manipulating it, then freeing the references. Some objects can be instantiated from a previously created definition, such as a page or file layout definition.

The following example creates a field object:

```
Local field &MyField
&MyField = GetField();
```

Getxxx built-in functions generally provide access to data that already exists, whether in the data buffers or from an existing definition.

Createxxx functions generally create defined objects that do not yet exist in the data buffer. *Createxxx* functions create only a buffer structure. They do not populate it with data. For example, the following function returns a record object for a record that already exists in the component buffer:

```
&REC = GetRecord();
```

The following example creates a standalone record. However, there is no data in &REC2. The specified record definition must be created previously, but the record doesn't have to exist in the buffer:

```
&REC2 = CreateRecord(EMP_CHKLIST_ITM);
```

The exceptions to the general Get/Create rule are objects that have no built-in functions and can only be instantiated from a session object (such as Tree classes, Component Interfaces, and so on). For *most* of these classes, when you use **Getxxx**, all you get is an *identifier* for the object. To fully instantiate the object, you must use an **Open** method.

In this section, we discuss the following topics:

- Object properties.
- Object methods.

See Also

PeopleTools PeopleBook: PeopleCode Reference, "Session Class"

Object Properties

To set or get characteristics of an object, or to determine the state of an object, you must access its properties through *dot notation* syntax. Follow the reference to the object with a period, followed by the property, and assign it a value. The format is generally as follows:

```
Object.Property = Value
```

The following example hides the field &MYFIELD:

```
&MYFIELD.Visible = False
```

You can return information about an object by returning the value of one of its properties. In the following example, &X is a variable that is assigned the value found in the field &MYFIELD:

```
&X = &MYFIELD.Value
```

In the following example, a property is used as the test for a condition:

```
If &ROWSET.ActiveRowCount <> &I Then
```

Object Methods

You also use dot notation to execute methods. Follow the reference to the object with a period, then by the method name and any parameters the method takes. The format is generally:

```
Object.method();
```

You can string methods and property values together into one statement. The following example strings together the GetField() method with the Name property:

```
If &REC_BASE.GetField(&R).Name = &REC_RELLANG.GetField(&J).Name Then
```

Some methods return a Boolean value: True if the method executes successfully; False if it doesn't. The following method compares all like-named fields of the current record object with the specified record. This method returns as True if all like-named fields have the same value:

```
If &MYRECORD.CompareFields(&OTHERRECORD) Then
```

Other methods return a reference to an object. The **GetCurrEffRow** method returns a row object:

```
&MYROW = &MYROWSET.GetCurrEffRow();
```

Some methods do not return anything. Each method's documentation indicates what it returns.

Many objects have default methods. Instead of entering the name of the method explicitly, you can use that method's parameters. Objects with default methods are *composite objects*, that is, they contain additional objects within them. The default method is generally the method used to get the lower-level object.

A good example of a composite object is a record object. Record definitions are composed of field definitions. The default method for a record object is **GetField**.

The following lines of code are equivalent:

```
&FIELD = &RECORD.GetField(FIELD.EMPLID);
```

```
&FIELD = &RECORD.EMPLID;
```

Note. If the field you're accessing has the same name as a record property (such as, Name) you cannot use the shortcut method for accessing the field. You must use the `GetField` method.

Another example of default methods concerns rowsets and rows. Rowsets are made up of rows, so the default method for a rowset is **GetRow**. The two specified lines of code are equivalent: They both get the fifth row of the rowset:

```
&ROWSET = GetRowSet();

/*the next two lines of code are equivalent */

&ROW = &ROWSET.GetRow(5);
&ROW = &ROWSET(5);
```

The following example illustrates the long way of enabling the NAME field on a second level scroll (the code is executing on the first level scroll):

```
GetRowset(SCROLL.EMPLOYEE_CHECKLIST).GetRow(1).GetRecord(EMPL_CHKLIST_ITM).GetField(FIELD.NAME).Enabled = True;
```

Using default methods enables you to shorten the previous code to the following:

```
GetRowset(SCROLL.EMPLOYEE_CHECKLIST)(1).EMPL_CHKLIST_ITM.NAME.Enabled = True;
```

Expressions of the form *class.name.property* or *class.name.method(..)* are converted to a corresponding object. For example:

```
&temp = RECORD.JOB.IsChanged;
```

is evaluated as if it were

```
&temp = GetRecord(RECORD.JOB).IsChanged;
```

Furthermore,

```
JOB.EMPLID.Visible = False;
```

is evaluated as if it were

```
GetField(JOB.EMPLID).Visible = False;
```

Assigning Objects

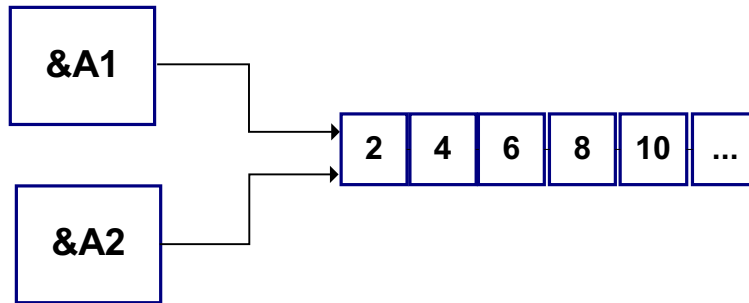
When you assign one object to another, you are not creating a copy of the object, but are making only a copy of the reference.

In the following example, `&A1` and `&A2` do not refer to separate and distinct objects, but refer to the same object. The assignment of `&A1` to `&A2` does not allocate any database memory or copy any part of the original object. It makes `&A2` refer to the same object to which `&A1` refers.

```
Local Array of Number &A1, &A2;
```



```
&A1 = CreateArray(2, 4, 6, 8, 10);
&A2 = &A1;
```

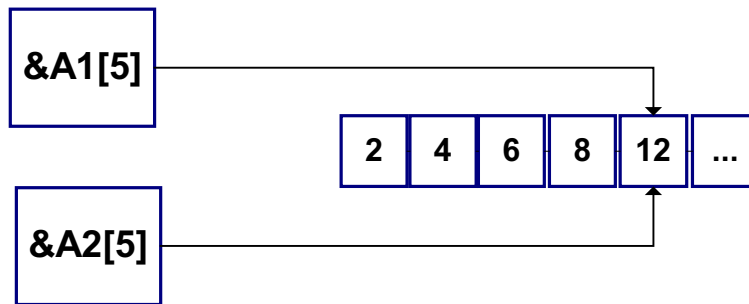


Representation of two arrays

If the next statement is:

```
&A2[5] = 12;
```

then **&A1[5]** also equals 12.



Representation of two arrays with same content

The following example is *not* considered an object assignment:

```
Local number &NUM;
Local Array of Number &A1;

&A1 = CreateArray(2, 4, 6, 8, 10);

&NUM = &A1[3];
```

&NUM is of data type Number, which is not an object type. If you later change the value of **&NUM** in the program, you won't change the element in the array.

Passing Objects

All PeopleCode objects can be passed as function parameters. The application developer can pass complex data structures between PeopleCode functions (as opposed to passing long lists of fields). If a function is passed an object, the function works on the actual object, not on a copy of the object.

In the following simple example, a reference to the Visible property is passed, not the *value* of Visible. This enables the MyPeopleCodeFunction either to get or set the value of the Visible property:

```
MyPeopleCodeFunction(&MyField.Visible);
```

In the following example, the function Process_Rowset loops through every row and record in the rowset it is passed and executes an UPDATE statement on each record in the rowset. This function can be called from any PeopleCode program and can process any rowset that is passed to it.

```
Local Rowset &RS;
Local Record &REC;

Function Process_RowSet(&ROWSET as Rowset);

    For &I = 1 To &ROWSET.RowCount
        For &J = 1 To &ROWSET.Recordcount
            &REC = &ROWSET.GetRow(&I).GetRecord(&J);
            &REC.Update();
        End-For;
    End-For;
End-Function;

&RS = GetLevel0();

Process_RowSet(&RS);
```

The following function takes a rowset and a record passed in from another program. GetRecord won't take a variable for the record; however, using the @ symbol you can de-reference the record name.

```
Function Get_My_Row(&PASSED_ROWSET, &PASSED_RECORD)

    For &ROWSET_ROW = 1 To &PASSED_ROWSET.RowCount
        &UNDERLYINGREC = "RECORD." | &PASSED_ROWSET.DBRecordName;
        &ROW_RECORD =
        &PASSED_ROWSET.GetRow(&ROWSET_ROW).GetRecord(@&UNDERLYINGREC);

        /* Do other processing */

    End-For;

End-Function;
```


CHAPTER 8

Using Methods and Built-In Functions

This chapter provides an overview of using PeopleCode methods and built-in functions and discusses how to:

- Work with restrictions on method and function use.
- Implement modal transfers.
- Implement the multi-row insert feature.
- Use the ImageReference field.
- Use PeopleCode with PeopleSoft Internet Architecture.
- Insert rows using PeopleCode.
- Use the GenerateTree function.
- Understand file attachment architecture.
- Use OLE functions.
- Use the Select and SelectNew methods.
- Use standalone rowsets.
- Use errors and warnings.
- Use RemoteCall.

See Also

PeopleTools PeopleBook: PeopleCode Reference, "PeopleCode Built-in Functions"

Working With Restrictions on Method and Function Use

As a licensee of PeopleTools, you are licensed to use the base portal technology, which is limited to navigation to licensed PeopleSoft applications. If you want to register additional non-PeopleSoft content, customize your homepage, or create any pagelets, you must license PeopleSoft Enterprise Portal.

In this section, we summarize the general restrictions on the use of built-in functions and methods in certain PeopleCode events and discuss the following topics:

- Think-time functions.
- WinMessage and MessageBox.
- Program execution with fields not in the data buffer.
- Errors and warnings.
- DoSave.
- Record object database methods.
- SQL object methods and functions.
- Component interface restricted functions.
- SearchInit PeopleCode function restrictions.
- CallAppEngine.
- ReturnToServer.
- GetPage.
- GetGrid.
- Publish method.

Think-Time Functions

"Think-time" functions suspend processing either until the user has taken some action (such as clicking a button in a message box), or until an external process has run to completion (for example, a remote process).

Think-time functions should be avoided in any of the following PeopleCode events:

- SavePreChange
- Workflow
- RowSelect
- SavePostChange
- Any PeopleCode event that fires as a result of a ScrollSelect, ScrollSelectNew, RowScrollSelect, or RowScrollSelectNew function call
- Any PeopleCode event that fires as a result of a Select or SelectNew rowset method

Violation of this rule can result in application failure.

The following are think-time functions:

- Call to an external DLL
- DoCancel
- DoModal
- DoModalComponent
- Exec (think-time only when synchronous)
- File attach functions
- InsertImage
- Object functions (think-time only when object requires user action) (CreateObject, ObjectDoMethod, ObjectSetProperty, ObjectGetProperty)
- Prompt
- RemoteCall
- RevalidatePassword
- WinExec (think-time only when synchronous)
- WinMessage and MessageBox (depending on *style* parameter)

WinMessage and MessageBox

The **WinMessage** and **MessageBox** functions sometimes behave as think-time functions, depending on the value passed in the function's *style* parameter, which controls, among other things, the number of buttons displayed in the message dialog box.

Note. The *style* parameter is ignored if the message has any severity other than Message.

Here is the syntax of both functions:

```
MessageBox(style, title, message_set, message_num, default_txt [, paramlist])
```

```
WinMessage(message [, style] [, title])
```

Note. The **WinMessage** function is supported for compatibility with previous releases of PeopleTools. Future applications should use MessageBox instead.

If the *style* parameter specifies more than one button, the function behaves as a think-time function and is subject to the same restrictions as other think-time functions (that is, it should never be used from SavePreChange through SavePostChange PeopleCode, or in RowSelect).

If the *style* parameter specifies a single button (that is, the OK button), then the function can be called in any PeopleCode event.

Note. In Windows Client, MessageBox dialogs include an **Explain** button to display more detailed information stored in the Message Catalog. The presence of the **Explain** button has no bearing on whether a message box behaves as a think-time function.

The *style* parameter is optional in **WinMessage**. If *style* is omitted **WinMessage** displays OK and Cancel buttons, which causes the function to behave as a think-time function. To avoid this situation, you should always pass an appropriate value in the **WinMessage** style parameter.

The following table shows the values that can be passed in the *style* parameter. To calculate the value to pass make one selection from each category in the table, then add the selections:

Category	Value	Constant	Meaning
Buttons	0	%MsgStyle_OK	The message box contains one pushbutton: OK.
	1	%MsgStyle_OKCancel	The message box contains two pushbuttons: OK and Cancel.
	2	%MsgStyle_AbortRetryIgnore	The message box contains three pushbuttons: Abort, Retry, and Ignore.
	3	%MsgStyle_YesNoCancel	The message box contains three pushbuttons: Yes, No, and Cancel.
	4	%MsgStyle_YesNo	The message box contains two push buttons: Yes and No.
	5	%MsgStyle_RetryCancel	The message box contains two push buttons: Retry and Cancel.

Note. The following values for *style* can only be used in Windows Client. They have no affect in PeopleSoft Internet Architecture.

Category	Value	Constant	Meaning
Default Button	0	%MsgDefault_First	The first button is the default.
	256	%MsgDefault_Second	The second button is the default.
	512	%MsgDefault_Third	The third button is the default.
Icon	0	%MsgIcon_None	None
	16	%MsgIcon_Error	A stop-sign icon appears in the message box.
	32	%MsgIcon_Query	A question-mark icon appears in the message box.

Category	Value	Constant	Meaning
	48	%MsgIcon_Warning	An exclamation-point icon appears in the message box.
	64	%MsgIcon_Info	An icon consisting of a lowercase letter "I" in a circle appears in the message box.

See Also

PeopleTools PeopleBook: PeopleCode Reference, "PeopleCode Built-in Functions", MessageBox, WinMessage

Program Execution With Fields not in the Data Buffer

Accessing a field that isn't in the data buffer under certain conditions causes a portion of your PeopleCode program to be skipped. The skip occurs only:

- in the Import Manager or if the reference is from the FieldDefault or FieldFormula events, and
- the reference executes on a field that doesn't exist in the data buffer

After the call to the invalid field, execution skips to the next "top-level" statement. Top-level statements are statements that aren't nested inside other statements. The start of a PeopleCode program is a top-level statement. Nesting begins with the first conditional statement (such as While or If) or the first function call.

For example, if your code is executing in a function *and* inside an If...then...end-if statement, and it runs into the skip conditions, the next statement executed is the one *after* the end-if, still inside the function.

Errors and Warnings

Error and **Warning** should not be used in FieldDefault, FieldFormula, RowInit, FieldChange, RowInsert, SavePreChange, WorkFlow, and SavePostChange PeopleCode. An **Error** or **Warning** in these events causes a runtime error that forces cancellation of the component.

See Also

PeopleTools PeopleBook: PeopleCode Reference, "PeopleCode Built-in Functions", Warning, Error

DoSave

Use **DoSave** only in FieldEdit, FieldChange, or MenuItemSelected PeopleCode.

See Also

PeopleTools PeopleBook: PeopleCode Reference, "PeopleCode Built-in Functions", DoSave

Record Object Database Methods

The following record object methods are used to update the database:

- Delete
- Insert
- Update

These methods should only be used in events that allow database updates, that is, in the following events:

- SavePreChange
- WorkFlow
- SavePostChange
- Message Subscription
- FieldChange
- Application Engine PeopleCode action

See Also

PeopleTools PeopleBook: PeopleCode Reference, "Record Class"

SQL Object Methods and Functions

Use the SQL object to update the database. Use these functions and methods only in events that allow database updates, that is, in the following events:

- SavePreChange
- WorkFlow
- SavePostChange
- Message Subscription
- FieldChange
- Application Engine PeopleCode action

Component Interface Restricted Functions

PeopleCode events and functions that relate exclusively to GUI and online processing can't be used by Component Interface. These include:

- Menu PeopleCode and pop-up menus. The **ItemSelected** and **PrePopup** PeopleCode events are not supported. In addition, the **DisableMenuItem**, **EnableMenuItem**, and **HideMenuItem** functions aren't supported.
- Transfers between components, including modal transfers. The **DoModal**, **EndModal**, **IsModal**, **Transfer**, **TransferPage**, **DoModalComponent**, and **IsModalComponent** functions cannot be used.
- Cursor positioning by **SetCursorPos**.
- **SetControlValue** cannot be used.
- **WinMessage** cannot be used.

These functions are ignored when executed using a Component Interface. That is, these functions do nothing and return a default value.

Transfer terminates the current PeopleCode program.

See Also

PeopleTools PeopleBook: Peoplesoft Component Interfaces, "Programming Component Interfaces in PeopleCode"

SearchInit PeopleCode Function Restrictions

You can't use the following functions in SearchInit PeopleCode:

- DoModal
- DoModalComponent
- Transfer
- TransferPage

CallAppEngine

The **CallAppEngine** function should only be used in events that allow database updates because generally, if you're calling PeopleSoft Application Engine, you're intending to perform database updates. This includes the following PeopleCode events:

- SavePreChange (Page)
- SavePostChange (Page)

- Workflow
- Message Subscription
- FieldChange

CallAppEngine cannot be used in an PeopleSoft Application Engine PeopleCode action. If you need to access one Application Engine program from another Application Engine program, use the **CallSection** action.

See Also

PeopleTools PeopleBook: Peoplesoft Application Engine, "Creating Application Engine Programs," Call Section Actions

ReturnToServer

The **ReturnToServer** function returns a value from a PeopleCode application messaging program to the publication or subscription server. You would use this in either your publication or subscription routing code, not in a Component Processor flow event.

See Also

PeopleTools PeopleBook: Peoplesoft Integration Broker, "Defining Messages"

GetPage

The **GetPage** function cannot be used until after the page processor has loaded the page. You shouldn't use this function in an event prior to the PostBuild event.

See Also

"PeopleCode and the Component Processor"

PeopleTools PeopleBook: PeopleCode Reference, "PeopleCode Built-in Functions", GetPage

GetGrid

PeopleSoft builds a grid one row at a time. Because the Grid class applies to a complete grid, you can't use the **GetGrid** function in an event prior to the Activate Event.

See Also

"PeopleCode and the Component Processor", Activate Event

PeopleTools PeopleBook: PeopleCode Reference, "PeopleCode Built-in Functions", GetGrid

Publish Method

If you are using PeopleSoft Messaging, your **Publish** PeopleCode should go in the SavePostChange event, for either the record or the Component.

See Also

PeopleTools PeopleBook: Peoplesoft Integration Broker, "Defining Messages"

SyncRequest

If you are using PeopleSoft Messaging, your **SyncRequest** PeopleCode should go in the SavePostChange event, for either the record or the Component.

See Also

PeopleTools PeopleBook: Peoplesoft Integration Broker, "Defining Messages"

Implementing Modal Transfers

Modal transfers enable you to transfer from one component (the *originating* component) to another component (the *modal* component) modally; that is, requiring the user to press the OK or Cancel buttons on the modal component before returning to the originating component.

Modal transfers provide some control over the order in which the user fills in pages, which is useful where data in the originating component can be derived from data entered by the user into the modal component.

Be careful to not overuse this feature as it forces users to complete interaction with the modal page before returning to the main component.

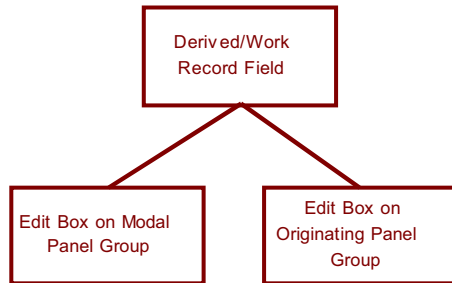
Note. Modal transfers *cannot* be initiated from SearchInit PeopleCode.

A modal component resembles a Windows modal dialog. It displays three buttons: OK, Cancel, and Apply. No toolbars or windows are available while the modal component has the focus.

- The OK button saves changes to the modal component and returns the user to the originating component.
- The Apply button saves changes to the modal component without returning to the originating component.
- The Cancel button returns the user to the originating component without saving changes to the modal component.

Modal components are generally smaller than the page from which they are invoked. Remember that OK and Cancel buttons are added at runtime, thus increasing the size of the pages.

The originating component and the modal component share record fields in a Derived/Work record called a *shared work record*. The Derived/Work fields of this record provide the two components with an area in memory where they can share data. Edit boxes in both components are associated with the same Derived/Work field, so that changes made to this field in the originating component are reflected in the modal component, and vice versa.



Edit boxes on the originating and modal components share the same data

Edit boxes associated with the same Derived/Work fields must be placed at level zero in both the originating component and the modal component.

You can use the shared fields to:

- Pass values assigned to the search keys in the modal component search record. If these fields are missing or invalid, the search dialog box appears, enabling the user to enter search keys.
- Pass other values from the originating component to the modal component.
- Pass values back from the modal component to the originating component.

Considerations Before Implementing a Modal Transfer

Any component accessible through an application menu system can be accessed using a modal transfer. However, to implement a modal transfer, you must make some modifications to pages in both the originating component and the modal component. After these modifications are complete, you can implement the modal transfer using the **DoModalComponent** PeopleCode function from a page in the originating component.

Before beginning this process, you should answer the following questions:

- Should the originating component provide search key values for the modal component? If so, what are the search keys? (Check the modal component's search record.)
- Does the originating component need to pass any data to the modal component? If so, what record fields are needed to store this data?

- Does the modal component need to pass any data back to the originating component? If so, what record fields are needed to store this data?

To implement a modal transfer:

1. Create Derived/Work record fields for sharing data between the originating and modal components.

Create a new Derived/Work record or open an existing Derived/Work record. If suitable record fields exist, you can use them; otherwise create new record fields for any data that needs to be shared between the components. These can be search keys for the modal component, data to pass to the modal component, or data to pass back to the originating component.

2. Add derived work fields to the level-zero area of the originating component.

Add one edit box for each of the Derived/Work fields that you need to share between the originating and modal components to the level-zero area of the page from which the transfer will take place. You will probably want to make the edit boxes invisible.

3. Add these same derived work fields to the level-zero area of the modal component.

Add one edit box for each of the edit boxes that you added in the previous step to the level-zero area of the page to which you are transferring. You will probably want to make the edit boxes invisible.

4. Add PeopleCode to pass values into the Derived/Work fields in the originating component.

If you want to provide search key values or pass data to the modal page, write PeopleCode that assigns appropriate values to the Derived/Work fields before **DoModalComponent** is called.

For example, if the modal component search key is PERSONAL_DATA.EMPLID, you could place the following assignment statement in the Derived/Work field's RowInit event:

```
EMPLID = PERSONAL_DATA.EMPLID
```

You also might assign these values in the same program where **DoModalComponent** is called.

5. Add any necessary PeopleCode to access and change the Derived/Work fields in the modal component.

No PeopleCode is required to pass search key values during the search. However, if other data has been passed to the modal component, you may need PeopleCode to access and use the data. You may also need to assign new values to the shared fields so that they can be used by the originating component.

It is possible that the component was accessed through the menu system and not through a modal transfer. To write PeopleCode that runs only in the component when it is running modally, use the **IsModalComponent** built-in function:

```

If IsModalComponent() Then
    /* PeopleCode for modal execution only. */
End-If

```

6. Add any necessary PeopleCode to access changed Derived/Work fields in the originating component.

If the modal component has altered the data in the shared work fields, you can write PeopleCode to access and use the data after **DoModalComponent** has executed.

See Also

PeopleTools PeopleBook: PeopleCode Reference, "PeopleCode Built-in Functions,"
DoModalComponent, IsModal, EndModalComponent

Implementing the Multi-Row Insert Feature

Enabling the multi-row insert feature in grids or scrolls can reduce response times for transactions that usually require entering many rows of data. Multi-row enables the end-user to specify the number of rows to add to a grid or scroll and is then presented with multiple empty rows for data entry.

To use the multi-row insert feature:

1. Specify deferred Mode Processing.

Multi-row reduces transaction times by eliminating excess server trips. To take full advantage of this feature, the transaction should be set to execute in deferred mode. Deferred mode should be set for the component, all pages in the component, and all fields on those pages.

2. Enable Multi-row.

For each grid or scroll where appropriate, select the Allow Multi-row Insert check box under the Use tab in the grid or scroll property sheet.

3. Add ChangeOnInit PeopleCode.

Setting the ChangeOnInit property for a rowset to FALSE enables PeopleCode to modify data in the rowset during RowInit and RowInsert events without flagging the rows as changed. This ensures that only changes entered by the user will cause the effected row to be saved.

Note. Each rowset that is referenced by a grid or scroll with multi-row enabled should have the ChangeOnInit property for the rowset set to 'FALSE'. This includes child, or lower level rowsets. In addition, this property must be set prior to any RowInsert or RowInit PeopleCode for the affected row.

4. Empty Rows at Save.

After a transaction is saved, any empty rows are discarded before the page is redisplayed to the user. An *empty* row means that the user didn't access the data because PeopleCode or record defaults may have been used to initialize the row for the initial display.

Note. PeopleCode Save processing (that is, SaveEdit and SavePreChange) PeopleCode executes for all rows in the buffer (including the empty ones). Therefore SaveEdit and SavePreChange PeopleCode should be coded so that it is executed only if the field contains data, or if the row properties IsNew and IsChanged are both True. An alternative method is adding PeopleCode in the first save program in the component to explicitly delete any row based on the IsNew and IsChanged properties. If you choose this method, then rows should be deleted from the bottom of the data buffer to the top (last row first).

See Also

"PeopleCode and the Component Processor," Deferred Processing Mode

PeopleTools PeopleBook: PeopleCode Reference, "Rowset Class," ChangeOnInit

Additional Multi-Row Insert Considerations

This feature may not apply for every scroll or grid:

- Multi-row *cannot* be used with effective-dated grid or scrolls.
- Multi-row may not apply if the entire row is populated using PeopleCode, especially if the data is copied from prior rows. In this case, the feature may not apply, or, if it does, the default value of the ChangeOnInit property can be used (the default value is True, which means any PeopleCode updates done in the RowInit or RowInsert events set the IsChanged and IsNew properties to True).

Using the ImageReference Field

After you create an image definition in PeopleSoft Application Designer, if you want to associate it with a field at runtime, the field has to be of type ImageReference. An example of this is referencing a red, yellow, or green light on a page, depending on the context.

To change the image value of an ImageReference field:

1. Create a field of type ImageReference.
2. Create the images you want to use.

These images must be saved in PeopleSoft Application Designer, as Image definitions.

3. Add the field to a record that will be accessed by the page.

4. Add an image control to the page and associate the image control with the ImageReference field.
5. Assign the field value.

Use the keyword **Image** to assign a value to the field. For example:

```
Local Record &MyRec;
Global Number &MyResult;

&MyRec = GetRecord();
If &MyResult Then
    &MyRec.MyImageField.Value = Image.THUMBSUP;
Else
    &MyRec.MyImageField.Value = Image.THUMBSDOWN;
End-If;
```

See Also

PeopleTools PeopleBook: Peoplesoft Application Designer, "Creating Field Definitions"

Using PeopleCode With PeopleSoft Internet Architecture

If you're building pages using View Internet Options, there are a few considerations to take into account when you're creating PeopleCode programs.

As a licensee of PeopleTools, you are licensed to use the base portal technology, which is limited to navigation to licensed PeopleSoft applications. If you want to register additional non-PeopleSoft content, customize your homepage, or create any pagelets, you must license PeopleSoft Enterprise Portal.

In this section, we discuss how to:

- Use internet scripts.
- Use the field object Style property.
- Use the HTML area.
- Use HTML definitions and the GetHTML Text function.

Using Internet Scripts

An Internet Script is a specialized PeopleCode function that generates dynamic web content. Internet Scripts interact with web clients (browsers) using a request-response paradigm based on the behavior of the Hypertext Transfer Protocol.

See Also

PeopleTools PeopleBook: PeopleCode Reference, "Internet Script Classes (iScript)"

Using the Field Object Style Property

On the Use tab of the page properties, you can associate a page with a style sheet component.

The style sheet has several classes of styles defined for it. You can edit each style class to change the font, the color, the background, and so on. Then you can dynamically change the style of a field using the **Style** property on a field object. This does not change the style sheet: only the style class associated with that field.

The following example changes the style class of a field depending on the value entered by the user. This code is in the FieldChange event.

```
Local Field &field;

&field = GetField();

If TESTFIELD1 = 1 Then;
    &field.Style = "PSHYPERLINK";
End-If;

If TESTFIELD1 = 2 Then;
    &field.Style = "PSIMAGE";
End-If;
```

TESTFIELD1:  1

Field with PSHYPERLINK style

TESTFIELD1:  2

Field with PSIMAGE style

See Also

PeopleTools PeopleBook: Peoplesoft Application Designer, "Creating Style Sheet Definitions"

PeopleTools PeopleBook: PeopleCode Reference, "Field Class"

Using the HTML Area

There are two methods to populate an HTML area control. One method is to select **Constant**, and enter your HTML directly into the control.

The other method is to associate the control with a record field, then populate that field with the text you want displayed in the HTML area.

Note. When you associate an HTML area control with a field, make sure the field is long enough to contain the data you want to pass to it. For example, if you associate an HTML area control with a field that is only 10 characters long, only the first 10 characters of your text is displayed.

The following code populates an HTML area with a simple bulleted list. This code is in the RowInit event of the record field associated with the HTML control.

```
Local Field &HTMLField;

&HTMLField = GetField();
&HTMLField.Value = "<ul><li>Item one</li><li>Item two</li></ul>";
```

The following code is in the FieldChange event of a pushbutton. It populates an HTML area (associated with the record field CHART_DATA.HTMLAREA) with a simple list.

```
Local Field &HTMLField;

&HTMLField = GetRecord(Record.CHART_DATA).HTMLAREA;
&HTMLField.Value = "<ul><li>Item one</li><li>Item two</li></ul>";
```

The following code populates an HTML area (associated with the record DERIVED_HTML, the field HTMLAREA) with the output of the GenerateTree function:

```
DERIVED_HTML.HTMLAREA = GenerateTree(&TREECTL);
```

The following tags are unsupported by the HTML area control:

```
<body>
<frame>
<frameset>
<form>
<head>
<html>
<meta>
<title>
```

See Also

Using the GenerateTree Function

PeopleTools PeopleBook: Peoplesoft Application Designer, "Creating HTML Definitions"

Using HTML Definitions and the GetHTMLText Function

If you're using the same HTML text in more than one place, or if it's a large, unwieldy string, you can create an HTML definition in PeopleSoft Application Designer, then use the GetHTMLText function to populate an HTML area control.

The following is the HTML string to create a simple table:

```
<P>
<TABLE>

  <TR bgColor=#008000>
    <TD>
      <P><FONT color=#f5f5dc face="Arial, Helvetica, sans-serif"
        size=2>message 1 </FONT></P></TD></TR>
  <TR bgColor=#0000cd>
    <TD>
      <P><FONT color=#00ffff face="Arial, Helvetica, sans-serif"
        size=2>message 2</FONT></P></TD></TR>
</TABLE></P>
```

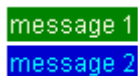
This HTML is saved to an HTML definition called TABLE_HTML:

This code is in the RowInit event of the record field associated with the HTML area control:

```
Local Field &HTMLField;

&HTMLField = GetField();
&string = GetHTMLText (HTML.TABLE_HTML);
&HTMLField.Value = &string;
```

This produces the following:



Example

See Also

PeopleTools PeopleBook: PeopleCode Reference, "PeopleCode Built-in Functions,"
GetHTMLText

Using HTML Definitions and the GetJavaScriptURL Method

HTML definitions can contain JavaScript programs in addition to HTML. If you have an HTML definition that contains a JavaScript, use the GetJavaScriptURL Response method to access (and execute) this JavaScript.

This example assumes the existence in the database of a HTML definition called "HelloWorld_JS", that contains some JavaScript.

```
Function iScript_TestJavaScript ()

    %Response.WriteLine("<script src= " |
    %Response.GetJavaScriptURL(HTML.HelloWorld_JS) | "></script>");

End-Function;
```

See Also

PeopleTools PeopleBook: PeopleCode Reference, "Internet Script Classes (iScript)," GetJavaScriptURL

PeopleTools PeopleBook: Peoplesoft Application Designer, "Creating HTML Definitions"

Inserting Rows Using PeopleCode

When inserting rows using PeopleCode, you can either use the Insert method with a record object, or create a SQL Insert statement using the SQL object.

- If you're doing a single insert, use the Record Insert method.
- If you're in a loop, and therefore calling the insert more than once, use the SQL object.

Why? Because the SQL object uses *dedicated cursors* and if the database you're working with supports it, *bulk insert*.

A *dedicated cursor* means that the SQL gets compiled only once on the database, so PeopleTools looks for the meta-SQL only once. This can increase performance.

For *bulk insert*, inserted rows are buffered and sent to the database server only when the buffer is full or a COMMIT occurs. This reduces the number of roundtrips to the database. Again, this can increase performance.

The following is an example of using the record insert method:

```
&REC = CreateRecord(Record.GREG);
&REC.DESCR.Value = "Y" | &I;
&REC.EMPLID.Value = &I;
&REC.Insert();
```

The following is an example using a SQL object for doing an insert:

```

&SQL = CreateSQL("%INSERT(:1)");
&REC = CreateRecord(Record.GREG);
&SQL.BulkMode = True;
For &I = 1 to 10
    &REC.DESCR.Value = "Y" | &I;
    &REC.EMPLID.Value = &I;
    &SQL.Execute(&REC);
End-For;

```

See Also

PeopleTools PeopleBook: PeopleCode Reference, "Record Class," Insert

PeopleTools PeopleBook: PeopleCode Reference, "SQL Class"

Using the GenerateTree Function

Use the GenerateTree function to display data in a tree format, with nodes and leaves. The result of the GenerateTree function is an HTML string, which can be displayed in an HTML area control. The tree generated by GenerateTree is called an *HTML tree*.

The GenerateTree function displays data from a rowset. You can populate this rowset using existing record data. You can also use the Tree Classes to display data from trees created using PeopleSoft Tree Manager.

To use this function you must set up a page for displaying the data, and populating a standalone rowset with the data to be displayed.



HTML tree example

The positional links at the top of the page (First, Previous, Next, Last, Left, Right) enable the user to navigate around the tree. These links are automatically generated as part of the execution of `GenerateTree`.

The icon next to a node can have a + sign or a - sign in it, depending on whether the node is collapsed or expanded. When a node is collapsed, none of the nodes that report to the collapsed node are displayed, and the icon has a + sign. When a node is expanded, all the nodes that report to it are displayed, and the icon has a - sign. You can collapse and expand a node by clicking in the icon. This enables you to view the tree at different levels of detail. When the icon for a node has no + or - sign in it, it is a terminal node, and cannot be expanded or collapsed.

In this section, we discuss how to:

- Build your HTML tree page.
- Use the HTML tree Rowset records.
- Use tree end-user actions (events).
- Customize PeopleCode for the HTML tree.
- Add mouse-over ability to the HTML tree.
- Add visual selection node indicators.

Building Your HTML Tree Page

The page you use to display the HTML tree must contain the following:

- An HTML area used to display the HTML tree.
- An character field that:
 - has a Page Field name
 - is at least 46 characters long
 - is invisible

Note. The edit box should be invisible, but *not* display-only. If the field is display-only, it can't be written to. Making it invisible makes it not visible to the user, but it still has a buffer that can be written to. **Note.** Page fields that have been specified as the invisible field do *not* need to be marked as Modifiable from HTML unless they are located on a page that is not active when GenerateTree is called. For example, if your application calls GenerateTree from one page then saves the result in a field that's displayed by an HTML area on another page in the component. In this case, the associated event field *must* be marked both Invisible and Modifiable from HTML.

Events are sent to the application from the HTML tree using the invisible field. The events are processed by FieldChange PeopleCode that is attached to the invisible field.

The following is an example page for an HTML tree:

The screenshot shows the 'Page Designer' window with a tab labeled 'Order'. The page layout includes a header section with the following fields:

SetID:	NNNNN	Set Control Value:	NNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
Tree Name:	NNNNNNNNNNNNNNNNNNNNNNNNNNNNNN	Effective Date:	22/22/2222

Below the header is a large, empty rectangular area representing the HTML tree display.

Example PeopleSoft Application Designer HTML tree page

The large area that is selected in the previous illustration is the HTML Area that displays the HTML tree. The HTML Area is attached to the DERIVED_HTML.HTMLAREA field for this example.

The white edit box is the invisible field used to pass events from the HTML tree to the application and is attached to the DERIVED_HTML.TREECTLEVENT field for this example.

The edit box must have a Page Field Name. In this example, the Page Field Name is TREECTLEVENT.

The HTML Tree Rowset Records

The GenerateTree function takes a pre-built and populated rowset as a parameter. This rowset must have a certain structure and contain certain fields. In this example, it's a standalone rowset, that is, the rowset is created using the CreateRowset function. The fields necessary for the rowset are contained in the following record definitions:

- The header record TREECTL_HRD, containing the subrecord TREECTL_HDR_SBR.
- The node record TREECTL_NDE, containing the subrecord TREECTL_NDE_SBR.

The header record is the level 0 record of the HTML tree rowset. It contains the options for the HTML tree, such as the name of the collapsed node image, the height of the images, the number of pixels to indent each node, and so on.

The node record is the level 1 record of the HTML tree rowset. It contains the tree data, and information about the data: is it a dynamic range leaf, what is the level, and so on.

There is a row in the level 1 scroll for each node or leaf in the tree data.

If you want to store additional application data with each node in the tree, you can incorporate the TREECTL_NDE_SBR into a record of your definition and use your record to define the HTML tree rowset.

For example, you might want to store application key values with each node record, so that when a user selects a node you have the data you need to perform the action that you want.

The following are the relevant fields in TREECTL_HDR_SBR:

PAGE_NAME	Name of the page that contains the HTML Area and the invisible field used to process the HTML tree events.
PAGE_FIELD_NAME	Page field name of the invisible field used to process the HTML tree events.
PAGE_SIZE	Number of nodes or leaves to send to the browser at a time. Set to 0 to send all the visible nodes or leaves to the browser. Default value is 0.
DISPLAY_LEVELS	Number of levels to display on the browser at a time. Default value is 8.

COLLAPSED_IMAGE	Collapsed node image name. Default value is PT_TREE_COLLAPSED.
EXPANDED_IMAGE	Expanded node image name. Default value is PT_TREE_EXPANDED.
END_NODE_IMAGE	End node image name. Default value is PT_TREE_END_NODE.
LEAF_IMAGE	Leaf image name. Default value is PT_TREE_LEAF.
IMAGE_WIDTH	Image width in pixels. All four images need to be the same width. Default value is 15 pixels.
IMAGE_HEIGHT	Image height in pixels. All four images need to be the same height. Default value is 12 pixels.
INDENT_PIXELS	Number of pixels to indent each level. Default value is 20 pixels.
TREECTL_VERSION	Version of the HTML tree. Default value is 812. Used with the DESCR_IMAGE field in TREECTL_HDR_SBR.

The following are the relevant fields in TREECTL_NDE_SBR:

LEAF_FLAG	If this is a leaf set to "Y". Default value is "N".
TREE_NODE	Node name.
DESCR	Node description. (optional)
RANGE_FROM	Leaf's range from value.
RANGE_TO	Leaf's range to value.
DYNAMIC_FLAG	If this leaf has a dynamic range, set to "Y". Default value is "N".
ACTIVE_FLAG	Set to "N" for the node or leaf not to be a link. Default value is "Y".
DISPLAY_OPTION	Set to "N" to display the name only. Set to "D" to display the description only. Set to "B" to display both the name and the description. Used for nodes only. Default value is "B".
STYLECLASSNAME	Used to control the style of the link associated with the node or leaf. Default value is PSHYPERLINK.
PARENT_FLAG	If this node is a parent and its direct children will be loaded now, set to "Y". If this node is a parent and its direct children are to be loaded on demand, set to "X". If this node is not a parent, set to "N". Default value is "N".
TREE_LEVEL_NUM	Set to the node's level. Default value is 1.
LEVEL_OFFSET	If a child node is to be displayed more than one level to the right of its parent, specify the number of additional

levels. Default value is 0.

DESCR_IMAGE

Use to display an image after the node or leaf image and before the name or description. There is a space between the two images. The new image isn't scaled. This field takes a string value, the name of an image definition created in PeopleSoft Application Designer.

This field is only recognized if the TREECTL_VERSION field is greater than or equal to 812.

EXPANDED_FLAG

When a node's EXPANDED_FLAG is "Y", the GenerateTree function expects the node's immediate children to be loaded into the &TREECTL Rowset (such as in PostBuild) and GenerateTree generates HTML such that the node is expanded and its immediate children are displayed.

See Also

PeopleTools PeopleBook: PeopleCode Reference, "PeopleCode Built-in Functions," CreateRowset

Using HTML Tree End-User Actions (Events)

The GenerateTree function works with an HTML area control and an invisible field. When an end-user selects a node, expands a node, collapses a node, or uses one of the navigation links, that event (end-user action) is passed to the invisible field, and the invisible field's FieldChange PeopleCode is executed.

The FieldChange PeopleCode Example program checks for expanding (or collapsing) a node, and selecting a node, by checking the first character in the invisible field. The following example checks for whether a node is selected:

```
If Left(TREECTLEVENT, 1) = "S" Then
```

In your application, you can check for the following end-user actions.

Event	Description
Tn	Toggle the node. Expand or collapse, the opposite of the previous state. <i>n</i> is the node's row number in the TREECTL_NODE rowset.
Xn	Expand the node, but first load the children. The children are loaded in PeopleCode, then the event is passed to GenerateTree, so that the HTML can be generated with the node expanded. <i>n</i> is the node's row number in the TREECTL_NODE rowset.
F	Display the first page.
P	Display the previous page.
N	Display the next page.
L	Display the last page.

Event	Description
Q	Scroll the display left one level.
R	Scroll the display right one level.
<i>Sn</i>	Select the node or leaf. <i>n</i> is the node's or leaf's row number in the TREECTL_NODE rowset

Note. Drag-and-drop is *not* supported.

See Also

FieldChange PeopleCode Example

Customizing PeopleCode for the HTML Tree (Part One)

The PeopleCode for initializing the HTML tree for this example was put into the PostBuild event of the component that contained the page with the HTML area used with the HTML tree.

The PostBuild PeopleCode Example program is an example of how to initialize the HTML tree using the Tree classes and load only the root node into the HTML tree rowset.

The first time a user expands a node, the node's direct children are loaded into the HTML tree rowset by the FieldChange PeopleCode Example program, shown in the following section. This chunking functionality enables the HTML tree to support trees of any size with good performance.

You can't just copy either the PostBuild or FieldChange PeopleCode example programs into your application. You must modify them to make them work with your data. Changes that you must make to the PostBuild PeopleCode are as follows.

To modify the PeopleCode for your HTML tree (part one):

1. Set the PAGE_NAME and PAGE_FIELD_NAME fields.

The PAGE_NAME field contains the name of the page that contains the HTML Area and the invisible field that will process the HTML tree events. The PAGE_FIELD_NAME field is the page field name of the invisible field that is used to process the HTML tree events.

Note. This is the Page Field name of the invisible field, *not* the invisible field name.

2. Set the tree specific variables.

The &SET_ID, &USERKEYVALUE, &TREE_NAME, &TREE_DT, and &BRANCH_NAME variables contain the specific information about the tree. Set these values to the tree you want to open. In the example PeopleCode, they are set as follows:

```

&SET_ID = PSTREEDEFN_VW.SETID;
&USERKEYVALUE = "";
&TREE_NAME = PSTREEDEFN_VW.TREE_NAME;
&TREE_DT = PSTREEDEFN_VW.EFFDT;
&BRANCH_NAME = "";

```

3. Set the PAGE_SIZE field.

If you do not want the page to expand vertically to display the tree, set the PAGE_SIZE to a number of rows that will fit inside the HTML Area. If some vertical expansion is okay, but you do not want the page to get too large, set the PAGE_SIZE to whatever value you like. Set the PAGE_SIZE to 0 if you don't care how big the page gets.

4. Set the DISPLAY_LEVELS field to the number of levels that will fit inside the HTML Area.

If this field is set too large, wrapping may occur. Positional links at the top of the HTML area enable the user to navigate as the tree expands.

5. (Optional) Set the DISPLAY_OPTION field.

The default for the DISPLAY_OPTION field is to display both the node name and the description. You can choose to display just the node name or just the description. The values for this field are:

Value	Description
N	Display name only
D	Display description only
B	Display both name and description

6. (Optional) Set the STYLECLASSNAME field for the root node.

The STYLECLASSNAME field controls the style of the link associated with a node or leaf. The default for the STYLECLASSNAME is PSHYPERLINK. If PSHYPERLINK isn't the style you want to use, change this field value to the style you want.

7. Change the last line to assign the output of GenerateTree() to the field attached to the HTML Area that will display the tree.

In this example, the HTML Area control is the DERIVED_HTML.HTMLAREA. You must specify the record and field name associated with the HTML area control on your page.

See Also

PostBuild PeopleCode Example

FieldChange PeopleCode Example

PostBuild PeopleCode Example

```

Component Rowset &TREECTL;

&NODE_ROWSET = CreateRowset (Record.TREECTL_NODE);
&TREECTL = CreateRowset (Record.TREECTL_HDR, &NODE_ROWSET);

&TREECTL.InsertRow(1);
&REC = &TREECTL.GetRow(2).GetRecord(1);

/* Set the HDR options:

1) PAGE_NAME - Name of the page that contains the HTML Area and the invisible
field that will be used to process the HTML tree events.
2) PAGE_FIELD_NAME - Page field name of the invisible field that will be used
to process the HTML tree events.
3) PAGE_SIZE - Number of nodes or leaves to send to the browser at a time.
Set to 0 to send all of the visible nodes or leaves to the browser. Default
value: 0
4) DISPLAY_LEVELS - Number of levels to display on the browser at a time.
Default value: 8
5) COLLAPSED_IMAGE - Collapsed node image name. Default value:
PT_TREE_COLLAPSED
6) EXPANDED_IMAGE - Expanded node image name. Default value: PT_TREE_EXPANDED
7) END_NODE_IMAGE - End node image name. Default value: PT_TREE_END_NODE
8) LEAF_IMAGE - Leaf image name. Default value: PT_TREE_LEAF
9) IMAGE_WIDTH - Image width. All four images need to be the same size.
Default value: 15
10) IMAGE_HEIGHT - Image height. Default value: 12
11) INDENT_PIXELS - Number of pixels to indent each level. Default value: 20
*/
&REC.GetField(Field.PAGE_NAME).Value = "TREECTL_TEST";
&REC.GetField(Field.PAGE_FIELD_NAME).Value = "TREECTLEVENT";
&REC.GetField(Field.PAGE_SIZE).Value = 15;
&REC.GetField(Field.DISPLAY_LEVELS).Value = 8;
&REC.GetField(Field.COLLAPSED_IMAGE).Value = "PT_TREE_COLLAPSED";
&REC.GetField(Field.EXPANDED_IMAGE).Value = "PT_TREE_EXPANDED";
&REC.GetField(Field.END_NODE_IMAGE).Value = "PT_TREE_END_NODE";
&REC.GetField(Field.LEAF_IMAGE).Value = "PT_TREE_LEAF";
&REC.GetField(Field.IMAGE_WIDTH).Value = 15;
&REC.GetField(Field.IMAGE_HEIGHT).Value = 12;
&REC.GetField(Field.INDENT_PIXELS).Value = 20;

&SET_ID = PSTREEDEFN_VW.SETID;
&USERKEYVALUE = "";
&TREE_NAME = PSTREEDEFN_VW.TREE_NAME;
&TREE_DT = PSTREEDEFN_VW.EFFDT;

```

```

&BRANCH_NAME = "";

&MYSESSION = %Session;
&SRC_TREE = &MYSESSION.GetTree();
&RES = &SRC_TREE.OPEN(&SET_ID, &USERKEYVALUE, &TREE_NAME, &TREE_DT,
&BRANCH_NAME, False);

/* Just insert the root node into the &TREECTL Rowset. If the root node has
children, set the &PARENT_FLAG to 'X', so that its children will be loaded on
demand. */

&ROOT_NODE = &SRC_TREE.FindRoot();

If &ROOT_NODE.HasChildren Then
    &PARENT_FLAG = "X";
Else
    &PARENT_FLAG = "N";
End-If;

&NODE_ROWSET = &TREECTL.GetRow(2).GetRowset(1);
&NODE_ROWSET.InsertRow(1);
&REC = &NODE_ROWSET.GetRow(2).GetRecord(1);

/* Set the NODE values:

1) LEAF_FLAG - If this is a leaf set to "Y". Default value: N
2) TREE_NODE - Node name.
3) DESCR - Node description. (optional)
4) RANGE_FROM - Leaf's range from value.
5) RANGE_TO - Leaf's range to value.
6) DYNAMIC_FLAG - If this leaf has a dynamic range, set to "Y". Default
value: N
7) ACTIVE_FLAG - Set to "N" for the node or leaf not to be a link. Default
value: Y
8) DISPLAY_OPTION - Set to "N" to display the name only. Set to "D" to
display the description only. Set to "B" to display both the name and the
description. Only used for nodes. Default value: B
9) STYLECLASSNAME - Used to control the style of the link associated with the
node or leaf. Default value: PSHYPERLINK
10) PARENT_FLAG - If this node is a parent and its direct children will be
loaded now, set to "Y". If this node is a parent and its direct children are
to be loaded on demand, set to "X". Default value: N
11) TREE_LEVEL_NUM - Set to the node's level. Default value: 1
12) LEVEL_OFFSET - If a child node is to be displayed more than one level to
the right of its parent, specify the number of additional levels. Default
value: 0
*/
&REC.GetField(Field.LEAF_FLAG).Value = "N";
&REC.GetField(Field.TREE_NODE).Value = &ROOT_NODE.NAME;

```

```

&REC.GetField(Field.DESCR).Value = &ROOT_NODE.DESCRPTION;
&REC.GetField(Field.RANGE_FROM).Value = "";
&REC.GetField(Field.RANGE_TO).Value = "";
&REC.GetField(Field.DYNAMIC_FLAG).Value = "N";
&REC.GetField(Field.ACTIVE_FLAG).Value = "Y";
&REC.GetField(Field.DISPLAY_OPTION).Value = "B";
&REC.GetField(Field.STYLECLASSNAME).Value = "PSHYPERLINK";
&REC.GetField(Field.PARENT_FLAG).Value = &PARENT_FLAG;
&REC.GetField(Field.TREE_LEVEL_NUM).Value = 1;
&REC.GetField(Field.LEVEL_OFFSET).Value = 0;

&SRC_TREE.Close();
DERIVED_HTML.HTMLAREA = GenerateTree(&TREECTL);

```

Customizing PeopleCode for the HTML Tree (Part Two)

Use the following FieldChange PeopleCode to process the events passed from the HTML tree to the application. The code that processes the load children event loads the direct children of a node the first time the node is expanded by the user. Changes that you must make to the FieldChange PeopleCode are as follows.

To modify the PeopleCode for your HTML tree (part two):

1. Globally change TREECTLEVENT to the name of the invisible field that will be used to process the events.
2. Set the tree specific variables.

The &SET_ID, &USERKEYVALUE, &TREE_NAME, &TREE_DT and &BRANCH_NAME variables contain the specific information about the tree. Set these values to the tree you want to open. In the example PeopleCode, they are set as follows:

```

&SET_ID = PSTREEDEFN_VW.SETID;
&USERKEYVALUE = "";
&TREE_NAME = PSTREEDEFN_VW.TREE_NAME;
&TREE_DT = PSTREEDEFN_VW.EFFDT;
&BRANCH_NAME = "";

```

3. (Optional) Set the DISPLAY_OPTION field.

The default for the DISPLAY_OPTION field is to display both the node name and the description. You can chose to display just the node name or just the description. The values for this field are:

<i>Value</i>	<i>Description</i>
N	Display name only

Value	Description
D	Display description only
B	Display both name and description

4. (Optional) Set the STYLECLASSNAME field for the root node.

The STYLECLASSNAME field controls the style of the link associated with a node or leaf. The default for the STYLECLASSNAME is PSHYPERLINK. If PSHYPERLINK isn't the style you want to use, change this field value to the style you want.

5. Change the assignment of the output of every GenerateTree call to the field attached to the HTML Area that will display the tree.

In this example, the HTML Area control is the DERIVED_HTML.HTMLAREA. You must specify the record and field name associated with the HTML area control on your page.

6. Change the code that processes the select event to perform the action you want when the user selects a node or leaf.

You will see this section marked as *Process Select Event* in the following code sample.

See Also

HTML Tree End-User Actions (Events)

FieldChange PeopleCode Example

```

Component Rowset &TREECTL;

/* process load children event */
If Left(TREECTLEVENT, 1) = "X" Then
    &ROW = Value(Right(TREECTLEVENT, Len(TREECTLEVENT) - 1)) + 1;
    &NODE_ROWSET = &TREECTL.GetRow(2).GetRowset(1);
    &PARENT_REC = &NODE_ROWSET.GetRow(&ROW).GetRecord(1);
    &PARENT_LEVEL = &PARENT_REC.GetField(Field.TREE_LEVEL_NUM).Value;
    &ROW = &ROW + 1;

    &SET_ID = PSTREEDEFN_VW.SETID;
    &USERKEYVALUE = "";
    &TREE_NAME = PSTREEDEFN_VW.TREE_NAME;
    &TREE_DT = PSTREEDEFN_VW.EFFDT;
    &BRANCH_NAME = "";

    &MYSESSION = %Session;
    &SRC_TREE = &MYSESSION.GetTree();

```

```
&RES = &SRC_TREE.OPEN(&SET_ID, &USERKEYVALUE, &TREE_NAME, &TREE_DT,
&BRANCH_NAME, False);
```

```
/* Find the parent node and expand the tree one level below the parent.
Insert just the direct children of the parent node into the &TREECTL Rowset.
If any of the child nodes have children, set their PARENT_FLAG to 'X', so that
their children are loaded on demand. */
```

```
&PARENT_NODE =
&SRC_TREE.FindNode(&PARENT_REC.GetField(Field.TREE_NODE).Value, "");
If &PARENT_NODE.HasChildren Then
    &PARENT_NODE.Expand(2);
```

```
If &PARENT_NODE.HasChildLeaves Then
    /* Load the child leaves into the &TREECTL Rowset. */
    &FIRST = True;
    &CHILD_LEAF = &PARENT_NODE.FirstChildLeaf;
    While &FIRST Or
        &CHILD_LEAF.HasNextSib
        If &FIRST Then
            &FIRST = False;
        Else
            &CHILD_LEAF = &CHILD_LEAF.NextSib;
        End-If;
        If &CHILD_LEAF.Dynamic = True Then
            &RANGE_FROM = "";
            &RANGE_TO = "";
            &DYNAMIC_RANGE = "Y";
        Else
            &RANGE_FROM = &CHILD_LEAF.RangeFrom;
            &RANGE_TO = &CHILD_LEAF.RangeTo;
            &DYNAMIC_RANGE = "N";
        End-If;

        &NODE_ROWSET.InsertRow(&ROW - 1);
        &REC = &NODE_ROWSET.GetRow(&ROW).GetRecord(1);
```

```
/* Set the NODE values:
```

- 1) LEAF_FLAG - If this is a leaf set to "Y". Default value: N
- 2) TREE_NODE - Node name.
- 3) DESCR - Node description. (optional)
- 4) RANGE_FROM - Leaf's range from value.
- 5) RANGE_TO - Leaf's range to value.
- 6) DYNAMIC_FLAG - If this leaf has a dynamic range, set to "Y". Default value: N

- 7) ACTIVE_FLAG - Set to "N" for the node or leaf not to be a link. Default value: Y
- 8) DISPLAY_OPTION - Set to "N" to display the name only. Set to "D" to display the description only. Set to "B" to display both the name and the description. Only used for nodes. Default value: B
- 9) STYLECLASSNAME - Used to control the style of the link associated with the node or leaf. Default value: PSHYPERLINK
- 10) PARENT_FLAG - If this node is a parent and its direct children will be loaded now, set to "Y". If this node is a parent and its direct children are to be loaded on demand, set to "X". Default value: N
- 11) TREE_LEVEL_NUM - Set to the node's level. Default value: 1
- 12) LEVEL_OFFSET - If a child node is to be displayed more than one level to the right of its parent, specify the number of additional levels. Default value: 0

```

*/

    &REC.GetField(Field.LEAF_FLAG).Value = "Y";
    &REC.GetField(Field.TREE_NODE).Value = "";
    &REC.GetField(Field.DESCR).Value = "";
    &REC.GetField(Field.RANGE_FROM).Value = &RANGE_FROM;
    &REC.GetField(Field.RANGE_TO).Value = &RANGE_TO;
    &REC.GetField(Field.DYNAMIC_FLAG).Value = &DYNAMIC_RANGE;
    &REC.GetField(Field.ACTIVE_FLAG).Value = "Y";
    &REC.GetField(Field.DISPLAY_OPTION).Value = "B";
    &REC.GetField(Field.STYLECLASSNAME).Value = "PSHYPERLINK";
    /* Leaves never have children. */
    &REC.GetField(Field.PARENT_FLAG).Value = "N";
    &REC.GetField(Field.TREE_LEVEL_NUM).Value = &PARENT_LEVEL + 1;
    &REC.GetField(Field.LEVEL_OFFSET).Value = 0;

    &ROW = &ROW + 1;
End-While;
End-If;

If &PARENT_NODE.HasChildNodes Then
    /* Load the child nodes into the &TREETL Rowset. */
    &FIRST = True;
    &CHILD_NODE = &PARENT_NODE.FirstChildNode;
    While &FIRST Or
        &CHILD_NODE.HasNextSib
        If &FIRST Then
            &FIRST = False;
        Else
            &CHILD_NODE = &CHILD_NODE.NextSib;
        End-If;
        If &CHILD_NODE.HasChildren Then
            &PARENT_FLAG = "X";
        Else
            &PARENT_FLAG = "N";

```

```

End-If;

/* If the tree uses strict levels, set the &LEVEL_OFFSET to the
number of levels that the child node is to the right of its parent minus 1. */
If &SRC_TREE.LevelUse = "S" Then
    &LEVEL_OFFSET = &CHILD_NODE.LevelNumber -
&PARENT_NODE.LevelNumber - 1;
Else
    &LEVEL_OFFSET = 0;
End-If;

&NODE_ROWSET.InsertRow(&ROW - 1);
&REC = &NODE_ROWSET.GetRow(&ROW).GetRecord(1);
&REC.GetField(Field.LEAF_FLAG).Value = "N";
&REC.GetField(Field.TREE_NODE).Value = &CHILD_NODE.Name;
&REC.GetField(Field.DESCR).Value = &CHILD_NODE.Description;
&REC.GetField(Field.RANGE_FROM).Value = "";
&REC.GetField(Field.RANGE_TO).Value = "";
&REC.GetField(Field.DYNAMIC_FLAG).Value = "N";
&REC.GetField(Field.ACTIVE_FLAG).Value = "Y";
&REC.GetField(Field.DISPLAY_OPTION).Value = "B";
&REC.GetField(Field.STYLECLASSNAME).Value = "PSHYPERLINK";
&REC.GetField(Field.PARENT_FLAG).Value = &PARENT_FLAG;
&REC.GetField(Field.TREE_LEVEL_NUM).Value = &PARENT_LEVEL + 1;
&REC.GetField(Field.LEVEL_OFFSET).Value = &LEVEL_OFFSET;

&ROW = &ROW + 1;
End-While;
End-If;

/* change the parent's PARENT_FLAG from 'X' to 'Y' */
&PARENT_REC.GetField(Field.PARENT_FLAG).Value = "Y";

HTMLAREA = GenerateTree(&TRECTL, TRECTLEVENT);
End-If;

&SRC_TREE.Close();
Else

    /* Process select event. */

    /* As an example, just display the selected node name or leaf range as a
    MessageBox. */

    If Left(TRECTLEVENT, 1) = "S" Then
        &ROW = Value(Right(TRECTLEVENT, Len(TRECTLEVENT) - 1)) + 1;

```

```

&NODE_ROWSET = &TREECTL.GetRow(2).GetRowset(1);
&REC = &NODE_ROWSET.GetRow(&ROW).GetRecord(1);
If &REC.GetField(Field.LEAF_FLAG).Value = "N" Then
    MessageBox(0, "", 0, 0, "The selected node is %1.",
&REC.GetField(Field.TREE_NODE).Value);
Else
    If &REC.GetField(Field.DYNAMIC_FLAG).Value = "N" Then
        If &REC.GetField(Field.RANGE_FROM).Value =
&REC.GetField(Field.RANGE_TO).Value Then
            &TEMP = "[" | &REC.GetField(Field.RANGE_FROM).Value | "]";
        Else
            &TEMP = "[" | &REC.GetField(Field.RANGE_FROM).Value | " - " |
&REC.GetField(Field.RANGE_TO).Value | "]";
        End-If;
    Else
        &TEMP = "[ ]";
    End-If;
    MessageBox(0, "", 0, 0, "The selected leaf is %1.", &TEMP);
End-If;
Else
    /* process all other events */
    HTMLAREA = GenerateTree(&TREECTL, TREECTLEVENT);
End-If;
End-If;

/* done processing the event, so clear it */
TREECTLEVENT = "";

```

Adding Mouse-Over Ability to the HTML Tree

If you want to add mouse-over ability to all the nodes, leaves, and so on to an HTML tree, you must add a few fields to the TREECTL_HDR_SBR, and some additional PeopleCode to the program, both to set the values, and set the images.

To add mouse-over ability to the HTML tree:

1. Add fields to the TREECTL_HDR_SBR record.

Add the following fields to the tree control header record:

- COLLAPSED_MSGNUM
- COLLAPSED_MSGSET
- END_NODE_MSGNUM
- END_NODE_MSGSET

- EXPANDED_MSGNUM
- EXPANDED_MSGSET
- LEAF_NODE_MSGNUM
- LEAF_NODE_MSGSET

2. Add the following PeopleCode to set the desired message set and number for the mouse over text.

```
&REC.GetField(Field.EXPANDED_MSGSET).Value = 2;
&REC.GetField(Field.EXPANDED_MSGNUM).Value = 903;
&REC.GetField(Field.COLLAPSED_MSGSET).Value = 2;
&REC.GetField(Field.COLLAPSED_MSGNUM).Value = 904;
&REC.GetField(Field.END_NODE_MSGSET).Value = 2;
&REC.GetField(Field.END_NODE_MSGNUM).Value = 905;
&REC.GetField(Field.LEAF_MSGSET).Value = 2;
&REC.GetField(Field.LEAF_MSGNUM).Value = 906;
```

3. Add fields to the TREECTL_NDE_SBR.

Add the following fields to the tree control node record:

- DESCR_MSGNUM
- DESCR_MSGSET

4. Add PeopleCode to set the DESCR_MSGNUM and DESCR_MSGSET to the desired values.

These two fields should be set to the correct message number and message set values that contain the text to be used as the mouse-over text.

Adding Visual Selection Node Indicators

Sometimes end-users need a visual indicator to specify which node has been selected, such as a different color or style for the selected node.



Showing selected node style

Use the following procedure to add this support.

To add selected node highlighting:

1. Add the field NODESELECTEDSTYLE to the TREECTL_HDR_SBR record.
2. Add PeopleCode to set the NODESELECTEDSTYLE to provide the highlighting effect.

The NODESELECTEDSTYLE field takes the name of a style class.

The following example uses the PSTREENODESELECTED style.

```
&REC.GetField(Field.NODESELECTEDSTYLE).Value = "PSTREENODESELECTED";
```

The style of the selected node can be set when processing the select event.

Note. You also must reset the style of the previous selected node when processing the select event. To find the previous selected node, you can search the node rowset looking for a node with a STYLECLASSNAME equal to the style you've set for selected nodes. Alternately, you could keep a global variable with the node's index in the rowset. However, if you keep an index variable, you may have to update the index when processing the load children event.

Understanding File Attachment Architecture

File attachments are supported by using PeopleCode built-ins that implement the transfer of a file to or from a browser using the application server. The file is either stored on, or retrieved from, a file server or database tables, referred to as the data storage system.

The general flow is as follows:

```
Browser <---> Web Server <---> Application Server <---> Data Storage System
```

The Browser <--> Web Server transfer is done using a standard HTML form construct. This can be done securely in an encrypted fashion if the web server uses SSL to communicate to the browser. After the file is received at the Web Server, it's stored in a temporary directory for transfer to either the browser (view operation) or the application server (attach operation). After the transfer is complete, the file is deleted.

The Web Server <--> Application Server transfer is done by using a JOLT service, which is securely encrypted. Because this transfer is done using the standard JOLT mechanism, no additional settings to the firewall are required (there's no need to open additional ports.) On the application server, the file is stored in one of the following temporary directories, depending on the operation:

- /tmp/PSFTP/Annnnnnnnnn (for attach operation)
- /tmp/PSFTP/Vnnnnnnnnn (for view operation)

where *nnnnnnnn* is a randomly generated transaction number for either attaching or viewing.

By using separately named subdirectories for each attachment, support for the simultaneously transfer of identically named files is guaranteed, as well as security that attachments won't be mixed up.

The Application Server <--> data storage system transfer is done either using the ftp service or directly to a database table.

- The ftp transfer can be securely encrypted if required. The ftp service is invoked on the application server by a script that is passed to the ftp binary. This script is in /tmp and is called ftp*nnnnnn*.txt and the output of invoking that script is called ftp*nnnnnn*.log (where *nnnnnn* is a random number.) These files are left in /tmp for debugging purposes, with the assumption that access to the application server is restricted and somewhat secure. For your own security, you may want to delete these files on a weekly basis.
- No log file and no script is required for writing to database tables. For debugging purposes, these commands leave traces in the PeopleCode trace files if Internal Tracing (256) is selected.

Note. The Search API does *not* work with database attachments.

Debugging File Attachment Problems

You should first verify that the ftp transfer was successful. To determine this, gain access to the application server and examine the ftp*nnnnnn*.log file and the ftp*nnnnnn*.txt file.

The following is an example of a log file. Notice that it shows that the file transfer was successful.

```
pt-hp01:$ cat ftp1039.txt
open 216.131.219.51
user ANONYMOUS TEST
bin
lcd /
lcd tmp
lcd PSFTP
lcd V13333
cd /
get "IA_FA_Test.TXT" "IA_FA_Test.TXT"
quit
pt-hp01:$
pt-hp01:$ cat ftp1039.log
about to submit ftp -vin /tmp/ftp1039.txt
Connected to 216.131.219.51.
```



```

220 TCARREON052500 Microsoft FTP Service (Version 4.0).
331 Anonymous access allowed, send identity (e-mail name) as password.
230 Anonymous user logged in.
200 Type set to I.
Local directory now /
Local directory now /tmp
Local directory now /tmp/PSFTP
Local directory now /tmp/PSFTP/V13333
250 CWD command successful.
200 PORT command successful.
150 Opening BINARY mode data connection for IA_FA_Test.TXT(72 bytes).
226 Transfer complete.
72 bytes received in 0.06 seconds (1.21 Kbytes/s)
221
File Transfer Successful
pt-hp01:$

```

If the file transfer wasn't successful, error messages in the log can help you determine the problem.

A common reason is that the ftp server was not accessible from the application server. This could be due to a wrong password, wrong account name, or perhaps because the application server was unable to resolve the ftp server hostname. Perhaps the ftp server was down and the application server was unable to route to it. Try pinging the ftp server from the application server system, and then try to ftp to the ftp server from the application server.

If you have an NT file server, it's possible that the hostname for the NT system is not associated with a fixed IP address and is not resolvable using DNS. If you have the application server on a Unix machine (Solaris, HP-UX, AIX), the application server can only resolve the hostname using DNS (or perhaps NIS or /etc/hosts), but *not* using WinBeui or WINS. This means that it won't be able to convert the hostname indicated for the NT file server into an IP address and route to it.

If you suspect this is the problem, try the following:

1. Telnet to the Unix application server.
2. Try to ping the suspect hostname.

For example:

```
ping TCARREON052500
```

If this fails, you must resolve the problem by either specifying the IP address in the ftp URL or putting your hostname into DNS/NIS/hosts so that Unix can resolve it.

Debugging Database Attachments

Turn on tracing for internal PeopleCode command (256) to generate information about each step of the attachment process for database attachments.

Configuring Multiple Application Servers to Support File Attachments

The /tmp/PSFTP directory is created by the first application server process to run on your application server. The assumption is that all such application server processes run with the same UserID. While this may be the case for the majority of people, this is not always the case. Frequently, a single computer runs application servers with many UserIDs.

For proper security, the default /tmp/PSFTP is created with 711 permissions, or "drwx--x--x". This means only the user who created the directory can read and write to it. If you have multiple users that must do this, you must change the permissions.

The best solution is to have all the users in a single group, as specified by /etc/groups, and then have the person who configured the application server telnet to the Unix computer and change the permissions (or create) for /tmp/PSFTP to be "771" or "drwxrwx--x". This permits only the application server processes to write to that directory.

In cases of a test environment where security is not significant, the administrator could change the permissions to 777 or "drwxrwxrwx". This permits all the application server processes to read and write to that directory, but also any user on the system.

Viewing File Attachments

When viewing a file attachment, the browser invokes a viewer based on the Content-Type mime category sent in the response header from the web server.

For example, if the user tried to view an mp3 file, the response header sent to the browser by the web server would be:

```
HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
Date: Mon, 01 Oct 2001 21:25:51 GMT
Content-Type: audio/mpeg
Accept-Ranges: bytes
Last-Modified: Mon, 01 Oct 2001 21:00:26 GMT
ETag: "78e21918bc4ac11:cc8"
Content-Length: 60
```

Notice that the Content-Type is "**audio/mpeg**". The browser uses this mime type to determine that a viewer for "audio/mpeg" is appropriate. If the web server did *not* send this Content-Type header, the browser wouldn't be able to determine the nature of the file being transmitted, and would be unable to invoke the correct application. In such a case, the browser would just try to display the file as "text/plain". This might be the wrong assumption.

The web-server determines the mime type by looking at the extension of the file and mapping it to a mime type. This is done differently depending on the web server and platform.

- For Microsoft IIS, this mapping is done by using the standard Windows extension association mechanism.
- For Apache, this is done by editing a configuration file called "**mime.types**".

- For WebLogic, a file called "**web.xml**" does the association or mapping.

The file "web.xml" is located in WEB-INF subdirectory. This file contains a section that looks like this:

```
</mime-mapping>
<mime-mapping>
  <extension>
    doc
  </extension>
  <mime-type>
    application/msword
  </mime-type>
</mime-mapping>
<mime-mapping>
  <extension>
    xls
  </extension>
  <mime-type>
    application/vnd.ms-excel
  </mime-type>
</mime-mapping>
```

Let's say you want to add an extension that causes ".log" files to be interpreted as plain text files.

RFCs 2045, 2046, 2047, 2048, and 2077 discuss Internet media types and the Internet media type registry. See <<ftp://ftp.iana.org/in-notes/iana/assignments/media-types/>>.

After checking the RFCs, You determine that the correct mime type is "**text/plain**". The following is an example of you would add to the above section of "web.xml":

```
</mime-mapping>
<mime-mapping>
  <extension>
    doc
  </extension>
  <mime-type>
    application/msword
  </mime-type>
</mime-mapping>
<mime-mapping>
  <extension>
    log
  </extension>
  <mime-type>
    text/plain
  </mime-type>
```

```

</mime-mapping>
<mime-mapping>
  <extension>
    xls
  </extension>
  <mime-type>
    application/vnd.ms-excel
  </mime-type>
</mime-mapping>

```

Once you save the file, the log extension is now associated with the content type of "text/plain".

Note. You must restart the WebLogic server before these changes are recognized.

Similar mappings can be achieved on other web servers, such as IBM WebSphere or Netscape Enterprise Server. Consult your web server documentation for details.

Note. When trying to view the objects, the extension must match *exactly* what is setup in the web.xml file. This value is case-sensitive, that is, DOC is not the same as doc. If the object view displays as garbled, chances are that either the extension is not setup in the web.xml file, or there is a case mismatch.

Chunking Considerations

When writing data to database tables, the data is automatically chunked, that is, stored in different rows of the database table. The size of each chunk is determined by the **Max Chunk File Size** field on the PSOPTIONS page. The default value of this field is 28,000 kilobytes.

Because each file is chunked, you can't directly pull data from the table. You should use the provided attachment functions, which automatically put the data back together into one file for you. The chunk size is stored with the file, therefore, if you change the system chunk size you can still retrieve files with different chunk sizes.

DB2 OS390 Considerations

If you store file attachments on DB2 OS390, you're limited to chunk sizes of 3862 for a 4k tablespace, or 32444 for a 32k page tablespace. By default, the only tablespace with 32K is PSIMAGE.

See Also

PeopleTools PeopleBook: Data Management, "PeopleTools Utilities," PeopleTools Options

Using Microsoft Windows NT 4 ftp Server

If you're using ftp, and running Windows NT 4 with the standard Microsoft ftp server, you must take the following into consideration.

Non-Microsoft ftp server programs on Windows NT, the MS Windows 2000 ftp server, as well as all Unix ftp servers do a "chroot" to the home directory of the ftp server automatically.

However, the Windows NT4 ftp server does not follow this behavior. This means that when an ftp client connects to the Microsoft ftp server, they are able to gain access to the entire drive, not just the ftp home directory.

This is not the case with Unix or non-Microsoft ftp server programs: those programs only allow access to the ftp home directory.

For example, suppose you are running the Microsoft Windows NT 4 ftp server and the home directory is as follows:

```
C:\inetpub\ftproot
```

When you log in, your current directory is the following:

```
c:\inetpub\ftproot
```

When you type the following:

```
cd \
```

Your current directory is now c:\.

If you are *not* running the Microsoft Windows NT 4 ftp server, and the home directory is the same:

```
c:\inetpub\ftproot
```

When you type the following:

```
cd \
```

Your current directory is now this:

```
c:\inetpub\ftproot.
```

This means that you *must* know the ftp server associated with the URL.

If you are using a Windows NT 4 ftp server, or another ftp server that shows this behavior, your URLs should have the following format. Note that the home directory of the ftp server must be appended to the address.

```
ftp://ftp:ftp@NT4system/inetpub/ftproot/tmp
```

If you are not using these sorts of servers, your URLs should have the following format:

```
ftp://ftp:ftp@NT4system/tmp
```

Using Non-DNS URLs With Unix

The URL passed to the file attachments generally has the following format:

```
ftp://user:pass@systemname/dir1/dir2
```

This assumes that the domain name *systemname* can be resolved with the Domain Name Server (DNS).

Some Windows systems can resolve *systemname* even when it can't be resolved with the DNS. However, some Unix systems cannot these resolve names.

If you are using a Unix system, and the domain name cannot be resolved with DNS, use the IP address. The following example assumes *systemname* has the IP address of 123.123.123.123.

```
ftp://user:pass@123.123.123.123/dir1/dir2
```

To determine the IP address, you can try to ping the system, that is:

```
ping systemname
```

Peoplesoft recommends using domain names that *can* be resolved. Only use IP addresses when absolutely necessary.

Using the Attachment Functions

PeopleTools supplies a subrecord and a work record you should include in your component when using the file attachment functions. If you want to store attachments in a database, there's an additional subrecord that you must use.

Note. You do not have to include the FILE_ATTACH_SBR subrecord in your component: you could just include the fields. However, PeopleSoft recommends including the subrecord.

The subrecord is called FILE_ATTACH_SBR. It contains the following fields:

ATTACHSYSFILENAME	The system file name, that is, the name of the file.
ATTACHUSERFILE	The user file name, that is, the name of the file that the user selects.

No PeopleCode is associated with this subrecord. Developers should include this subrecord in target records for using attachments.

The work record is called FILE_ATTACH_WRK, and contains the following fields:

ATTACHADD	Contains a PeopleCode program used for adding attachments (the AddAttachment built-in function.)
ATTACHDELETE	Contains a PeopleCode program used for deleting attachments (the DeleteAttachment built-in function.)

ATTACHVIEW Contains a PeopleCode program used for viewing attachments (the ViewAttachment built-in function.)

The PeopleCode program is associated with the FieldChange events for each, and the RowInit event for ATTACHADD. The PeopleCode is actually in the form of functions so this work record acts as a container for the appropriate methods related to attachments. You can choose to include this work record in your component and use component PeopleCode to invoke the functions, or you can use your own work record and just call these functions as needed from within your work record.

For storing attachments in the database, you must create a record that receives the attachments. This record can be called anything, but it must have the FILE_ATTDET_SBR subrecord included in it, and have no additional fields in it. The subrecord FILE_ATTDET_SBR has the following fields:

ATTACHSYSFILENAME	A unique system file name
FILE_SEQ	The file sequence number (used in chunking)
VERSION	Version number
FILE_SIZE	The physical size of the file
LASTUPDDTTM	Last update date and time
LASTUPDOPRID	The EMPL_ID of the last user to update the attachment
FILE_DATA	The data of the file

To use the Attachment functions:

3. Insert the subrecord (FILE_ATTACH_SBR) into a record used with your page.

Typically, this record is inserted as part of the primary database record.

Note. You do not have to include the subrecord in your component: you could just include the fields. However, PeopleSoft recommends including the subrecord.

4. If you want to store the file in the database, insert the subrecord FILE_ATTDET_SBR into a record used with your page.

This record can be called anything, but it must have the FILE_ATTDET_SBR subrecord included in it, and have no additional fields in it. This is the record that you point to with the URL when specifying storing a file in a database table.

5. Add buttons to add, delete, and view attachments as needed.

You can either use the FILE_ATTACH_WRK record for the necessary record fields, or you can use your own work fields.

6. Set up the URL.

If you're using the ftp archive, suppose the system is fileserver.ps.com, the user account is "joe", and the password is "secret". The full URL would be as follows:

```
ftp://joe:secret@fileserver.ps.com/
```

You can either create an entry for this URL in the URL Maintenance page, or specify it in your PeopleCode.

If you specify the URL in your PeopleCode, you don't have to hardcode the user name and password. You can de-reference them using a variable (or record field). This enables you to use encryption with the username and password.

If you're copying to a database table, you can either create an entry for this record (URL) in the URL Maintenance page, or specify it in your PeopleCode, as follows:

```
record://RESUMES_DB
```

7. Modify the PeopleCode to support your application.

If you use the supplied PeopleCode programs, you must specify the URL. You may also want to modify the extension of the expected files (the default is "" which is any file or *.*). You may want to indicate ".doc", ".xl", "*.html", "*.exe", "*.tar"). Read the comments in ATTACHADD.FieldChange program for instructions.

If you choose to use the FILE_ATTACH_WRK for buttons, you may want to use Component PeopleCode for your application specific code. This enables you to keep the code in FILE_ATTACH_WRK generic. If you use your own work fields, you can write your own code and call the functions in FILE_ATTACH_WRK as needed.

See Also

PeopleTools PeopleBook, Data Management, "PeopleTools Utilities," URL Maintenance

Coding Considerations Using the Attachment Functions

All file attachments are done using PeopleCode built-in functions, such as AddAttachment, ViewAttachment, GetAttachment, and so on. These functions move a file into and out of a file storage system, either the database or a file system. The file system movement is done using the ftp protocol.

One of the primary reasons for using these functions is because they abstract the storage of the attachments. This gives customers the choice to use either the database or ftp file systems. Which system to be used is determined by the URL passed as the first parameter in the attachment built-in function. The actual value of the URL is maintained in the URL Maintenance page.

A consequence of this means that the developers does not know whether a customer intends to store attachments in a database or a file system.

The following PeopleCode functions are used with administration work:

- DeleteAttachment deletes unwanted attachments. There is no 'roll-back' ability. For example, suppose a user selects to view an attachment, then cancels before the attachment is displayed. A copy of the attachment may still be in the user's temp directory.
- CopyAttachments moves all of the attachments specified from one file storage system (database or ftp) to another (database or ftp).
- CleanAttachments is an audit-like script that removes orphaned attachments, that is, attachments that exist in the database that aren't referenced by any record field.

If you only have a few files to copy using CopyAttachments, you may want to use the CopyFiles page (part of the PeopleTools Administration pages.) The CleanAttachments function is also available from this page.

If you'd like to schedule a regular job to clean up any orphaned file attachments, you can use the PeopleTools Application Engine program CLEANATT84 program.

It is important to store all references to attachments using the standard ATTACHSYSFILENAME. You should not reuse this field to store incomplete or non-standard versions of the name. For example, you should not store the full URL version of the attachment in this column and then use PeopleCode to parse the URL before invoking attachment commands. While this may work for get or put attachment calls, the Clean Attachments function deletes any file stored in a table that does not have a corresponding reference stored in ATTACHSYSFILENAME.

See Also

PeopleTools PeopleBook: Data Management, "PeopleTools Utilities," Copy File Attachments

Naming Considerations for URLs and File Attachments

If you change a URL on the URL Maintenance Page from ftp to record, you may have to reboot your application server before you can use the URL in your PeopleCode program.

User Filenames for PutAttachment

If a user attaches a file through the browser dialog box generated by AddAttachment, the filename is converted *before* that filename is returned to the AddAttachment call for storage.

For example, "My Resume.doc" is returned through the AddAttachment parameter as "My_Resume.doc", with the space changed to an underscore.

No comparable mapping occurs for user filenames with PutAttachment, which requires the user filename as an input parameter, rather than an output parameter that is returned to the user.

Instead, before storing the ATTACHUSERFILE value, code similar to this should be invoked:

```
&ATTACHUSERFILE = Substitute(&ATTACHUSERFILE, " ", "_");
```

```

&ATTACHUSERFILE = Substitute(&ATTACHUSERFILE, ";", "_");
&ATTACHUSERFILE = Substitute(&ATTACHUSERFILE, "+", "_");
&ATTACHUSERFILE = Substitute(&ATTACHUSERFILE, "%", "_");
&ATTACHUSERFILE = Substitute(&ATTACHUSERFILE, "&", "_");
&ATTACHUSERFILE = Substitute(&ATTACHUSERFILE, "'", "_");
&ATTACHUSERFILE = Substitute(&ATTACHUSERFILE, "!", "_");
&ATTACHUSERFILE = Substitute(&ATTACHUSERFILE, "@", "_");
&ATTACHUSERFILE = Substitute(&ATTACHUSERFILE, "#", "_");
&ATTACHUSERFILE = Substitute(&ATTACHUSERFILE, "$", "_");

```

Using OLE Functions

OLE automation is a Microsoft Windows protocol that enables one application to control another's operation. The applications communicate by means of an *OLE object*. One of the applications (called the *automation server*) makes available an OLE object that the second application (the *client* application) can use to send commands to the server application. The OLE object has *methods* associated with it, each of which corresponds to an action that the server application can perform. The client runs the methods, which cause the server application to perform the specified actions.

PeopleCode includes a set of functions that enable you to get OLE objects from automation servers and to run the objects' methods. In other words, your PeopleCode program can be an OLE client.

PeopleCode includes a set of functions that enable you to control other Microsoft Windows applications through Object Linking and Embedding (OLE). You can connect to any application that's registered as an OLE automation server and invoke its methods.

Note. Differences in Microsoft Windows applications from one release to the next (that is, properties becoming methods or vice versa) can cause problems with ObjectGetProperty, ObjectSetProperty() and ObjectDoMethod().

This section describes the PeopleCode functions you use to communicate with other applications through OLE and discusses:

- Data types.
- How to share a single object instance.
- OLE versus WinExec.

See Also

CreateObject, ObjectDoMethod, ObjectGetProperty, ObjectSetProperty

See the documentation for the OLE-automated application for details about OLE automation and the methods available for a particular OLE automation server.

Data Types

To support OLE, PeopleCode has a special data type—OBJECT—which it uses for OLE objects. The purpose of the OBJECT data type is to hold OLE objects during the course of a session so that you can run its methods. You cannot store OBJECT data for any extended period of time.

Important! OBJECT is a valid data type for variables, but not for record fields. Because OLE objects are by nature temporary, you can't store OBJECT data in a record field, including work record fields.

Some OLE object methods return data to the client. You can use such methods to get data from the automation server, if the method returns the data in a form that PeopleCode can process; that is, the data must be in a PeopleCode-supported data type. If the method returns data in an spreadsheet, for example, you won't be able to accept the data because PeopleCode doesn't support spreadsheets.

Sharing a Single Object Instance

When you need the services of an OLE automation server, you create an instance of its OLE object, using the **CreateObject** function. After you have the object, you can run its methods as often as you like. You do not need to create a new instance of the object each time.

In a typical scenario, you have a PeopleSoft component that needs to access Microsoft Excel, or Word, or some other automation server, perhaps one you have created yourself. Various PeopleCode programs associated with the component must run OLE object methods.

Rather than create a new instance of the OLE object in each PeopleCode program, you should create one instance of the OLE object in a PeopleCode program that runs when the component starts (such as RowInit) and assign it to a global variable. Then, any PeopleCode program can reference the object and invoke its methods.

OLE vs. WinExec

The **WinExec** and **Exec** built-in functions provide another way to start another application from PeopleCode. Unlike the OLE functions, however, **Exec** and **WinExec** don't enable you to control what actions the application takes after you start it. You can start the application—and if you use the synchronous option you can find out when it closes—but you can't affect its course or receive any data in return.

WinExec is appropriate in two situations:

- When you just want to start an application and continue processing.
- When you have a short, unvarying process that you want to run, such as copying a file.

The **Exec** function, unlike **WinExec** and the OLE functions, is not Windows-specific. This means that you can run it on an application server to call an executable on the application server platform, which in PeopleTools release 7 can be either Windows NT or UNIX.

Important! If you use the **WinExec** function with its synchronous option, the PeopleCode program (and the PeopleSoft application) remain paused until the called program is complete. If you start a program that waits for user input, such as Notepad, the application appears to hang until the user closes the called program. The synchronous option also imposes limits on the PeopleCode.

See Also

Exec, WinExec

Using the Select and SelectNew Methods

Select and SelectNew, like the ScrollSelect functions, enable you to control the process of selecting data into a page scroll. You can use these methods only with a rowset. A rowset can be thought of as a page scroll.

A level 0 rowset is a rowset that starts at the level0 of the page, and contains all the data in the component buffers.

A child rowset is a rowset that is contained by an upper level rowset, also called the parent rowset. For example, a level1 rowset could be considered the child rowset of a level0, parent, rowset. Or a level2 rowset could be the child rowset of a level1 rowset.

When a rowset is selected into, any autoselected child rowsets are also read. The child rowsets are read using a where clause that filters the rows according to the where clause used for the parent rowset, using a subselect.

The data contained in a child rowset depends on the row of the parent rowset.

Note. In addition to these methods there is the record object method SelectByKey, which enables you to select into a record object. If you're only interested in selecting a single row of data, you may want to consider this method instead.

In this section, we discuss the following topics:

- What select does.
- Select syntax.

See Also

"Data Buffer Access"

What Select Does

Select selects rows from a table or view and adds the rows to either a rowset or a row. Let's call the record definition of the table or view that it selected from the *select record*. Let's call the primary database record of the top level rowset object executing the method the *default scroll record*.

The select record can be the same as the default scroll record, or it can be a different record definition that has the same key fields as the default scroll record. If you define a select record that differs from the default scroll record, you can restrict the number of fields that are loaded into the buffers on the client work station by including only the fields you actually need.

Select automatically places child rowsets in the rowset object executing the method under the correct parent row. If it cannot match a child rowset to a parent row an error occurs.

Select also accepts an optional SQL string that can contain a WHERE clause restricting the number of rows selected into the scroll area. The SQL string can also contain an ORDER BY clause, enabling you to sort the rows.

Select and SelectNew generate an SQL SELECT statement at runtime, based on the fields in the select record and WHERE clause passed to them in the function call. This gives Select and SelectNew a significant advantage over **SQLExec**: they enable you to change the structure of the select record without affecting the PeopleCode, unless the field affected is referred to in the WHERE clause string. This can make the application easier to maintain.

Also, if you use one of the meta-SQL constructs or shortcuts in the WHERE clause, such as %KeyEqual or %List, even if a field has changed, you won't have to change your code.

Unlike the ScrollSelect functions, neither **Select** or **SelectNew** allow you to operate in *turbo* mode.

Select Syntax

The syntax of **Select** is:

```
Select ([paramlist], RECORD.selrecord [, wherestr, bindvars]);
```

Where *paramlist* is a list of child rowsets, given in the following form:

```
SCROLL.scrollname1 [, SCROLL.scrollname2] . . .
```

The first *scrollname* must be a child rowset of the rowset object executing the method, the second *scrollname* must be a child of the first child, and so on.

This syntax does the following:

- Specifies an optional child rowset into which to read the selected rows.
- Specifies the select record from which to select rows.
- Passes a string containing a SQL WHERE clause to restrict the selection of rows and/or an ORDER BY clause to sort the rows.

Let's examine the different parts of this syntax one at a time.

Specifying Child Rowsets

The first part of the Select syntax specifies a child rowset into which rows are selected. This parameter is optional.

If you don't specify any child rowsets in *paramlist*, **Select** selects from a SQL table or view specified by *selrecord* into the rowset object executing the method. For example, suppose you've instantiated a level 1 rowset &BUS_EXPENSES_PER. The following would select into this rowset:

```
Local Rowset &BUS_EXPENSES_PER;

&BUS_EXPENSES_PER = GetRowset (SCROLL.BUS_EXPSNESE_PER) ;

&BUS_EXPENSES_PER.Select (RECORD.BUS_EXPENSE_VW, "WHERE SETID = :1 and CUST_ID
= :2", SETID, CUST_ID) ;
```

If the rowset executing the method is a level 0 rowset, and you specify **Select** without specifying any child rowsets with *paramlist*, the method reads only a single row, because only one row is allowed at level 0.

Note. For developers familiar with previous releases of PeopleCode: In this situation, the **Select** method is acting like the **RowScrollSelect** function.

If you specify a child rowset in *paramlist*, **Select** selects from a SQL table or view specified by *selrecord* into the child rowset specified in *paramlist*, under the appropriate row of the rowset executing the method.

In the following example, rows are selected into a child rowset BUS_EXPENSE_DTL, matching level-one keys, and with the charge amount equal to or exceeding 200, sorting by that amount:

```
Local Record &REC_EXP;
Local Rowset &BUS_EXPENSE_PER;

&REC_EXP = GetRecord (RECORD.BUSINESS_EXPENSE_PER;

&BUS_EXPENSE_PER = GetRowset (SCROLL.BUS_EXPSNESE_PER) ;

&BUS_EXPENSE_PER.Select (SCROLL.BUS_EXPENSE_DTL, RECORD.BUS_EXPENSE_DTL, "WHERE
%KeyEqual (:1) AND EXPENSE_AMT >= 200 ORDER BY EXPENSE_AMT", &REC_EXP) ;
```

Specifying the Select Record

The record definition of the table or view being selected from is called the *select record*, and identified with RECORD.*selrecord*. The select record can be the same as the primary database record associated with the rowset executing the method, or it can be a different record definition that has compatible fields.

The select record must be defined in PeopleSoft Application Designer and be a built SQL table or view (using **Build, Project**), unless the select record is the same record as the primary database record associated with the rowset.

The select record can contain fewer fields than the primary record associated with the rowset, although it must contain any key fields to maintain dependencies with other records.

If you define a select record that differs from the primary database record for the rowset, you can restrict the number of fields that are loaded into the buffers on the client work station by including only the fields you actually need.

The WHERE Clause

Select accepts a SQL string that can contain a WHERE clause restricting the number of rows selected into the object. The SQL string can also contain an ORDER BY clause to sort the rows.

Select and **SelectNew** generate a SQL SELECT statement at runtime, based on the fields in the select record and the WHERE clause passed to them in the method parameters.

To avoid errors, the WHERE clause should explicitly select matching key fields on parent and child rows. You do this using the %KeyEqual meta-SQL.

Using Select Like RowScrollSelect

If the rowset executing the method is a level 0 rowset, and you specify **Select** without specifying any child rowsets with *paramlist*, the method reads only a single row, because only one row is allowed at level 0.

Note. For developers familiar with previous releases of PeopleCode: In this situation, the **Select** method is acting like the **RowScrollSelect** function.

If you qualify the lower level rowset such that it only returns one row, it will be acting like a RowScrollSelect.

```
&RSLVL1 = GetRowset (SCROLL.PHYSICAL_INV) ;
&RSLVL2 = &RSLVL1 (&PHYSICAL_ROW) .GetRowset (SCROLL.PO_RECEIVED_INV) ;
&REC2 = &RSLVL2.PO_RECEIVED_INV;
If &PO_ROW = 0 Then
    &RSLVL2.Select (PO_RECEIVED_INV, "WHERE %KeyEqual (:1) and qty_available >
0", &REC2);
End-if;
```

See Also

PeopleTools PeopleBook: PeopleCode Reference, "Meta-SQL," %KeyEqual

Using Standalone Rowsets

Standalone Rowsets are like regular Rowsets except they aren't associated with a component or page. Use them when you need to work on data that isn't associated with a component or page buffer. Prior to this release, this was done using work records. You still must build work pages.

Note. Standalone rowsets are *not* connected to the Page Processor, so there are no database updates when they are manipulated. Delete and insert activity on these types of rowsets aren't automatically applied at save time.

As with any PeopleTools object, the scope of standalone rowsets can be Local, Global, or Component. Consider the following code:

```
Local Rowset &MYRS;

&MYRS = CreateRowset (RECORD.SOMEREC) ;
```

This creates a Rowset with SOMEREC as the Level 0 record. The Rowset is unpopulated. Functionally it is the same as an array of records.

To populate the standalone Rowsets you can use:

- The Fill rowset method.
- The CopyTo rowset method.
- The record methods.

This section also discusses how to:

- Use standalone Rowsets to write a file.
- Use standalone Rowsets to read a file.

The Fill Method

The Fill method fills the rowset by reading records from the database, by first flushing out all the contents of the rowset. A where clause must be provided to get all the relevant rows.

```
Local Rowset &MYRS;
Local String &EMPLID;

&MYRS = CreateRowset (RECORD.SOMEREC) ;
&EMPLID = '8001';

&MYRS.Fill("where EMPLID = :1", &EMPLID);
```


The CopyTo Method

The CopyTo method copies like-named fields from a source rowset to a destination rowset. To perform the copy it uses like-named records for matching, unless specified. It works on any rowset except the Application Engine state records.

```
Local Rowset &MYRS1, MYRS2;
Local String &EMPLID;

&MYRS1 = CreateRowset(RECORD.SOMEREC);
&MYRS2 = CreateRowset(RECORD.SOMEREC);

&EMPLID = '8001';

&MYRS1.Fill("where EMPLID = :1", &EMPLID);
&MYRS1.CopyTo(&MYRS2);
```

Now &MYRS2 contains that same data as &MYRS1. In this case both &MYRS1 and &MYRS2 were build using like named records.

If you want to use the CopyTo method where there are no like-named records, you must specify the source and destination records. The following code will copies only like-named fields.

```
Local Rowset &MYRS1, MYRS2;
Local String &EMPLID;

&MYRS1 = CreateRowset(RECORD.SOMEREC1);
&MYRS2 = CreateRowset(RECORD.SOMEREC2);

&EMPLID = '8001';

&MYRS1.Fill("where EMPLID = :1", &EMPLID);
&MYRS1.CopyTo(&MYRS2, RECORD.SOMEREC1, RECORD.SOMEREC2);
```

Adding Child Rowset

The first parameter of the CreateRowset determines the top-level structure. In the previous examples we passed in the name of the record as the first parameter, thus the rowset was based on a record. You can also base the structure on a different rowset. In the following example &MYRS2 inherits the structure of &MYRS1.

```
Local Rowset &MYRS1, MYRS2;

&MYRS1 = CreateRowset(RECORD.SOMEREC1);
&MYRS2 = CreateRowset(&MYRS1);
```

To add a child rowset suppose the following records describe a relationship. The structure is made up of three records.

PERSONAL_DATA

BUS_EXPENSE_PER

BUS_EXPENSE_DTL

Here's how you build rowsets with child rowsets:

```
Local Rowset &rsBusExp, &rsBusExpPer, &rsBusExpDtl;

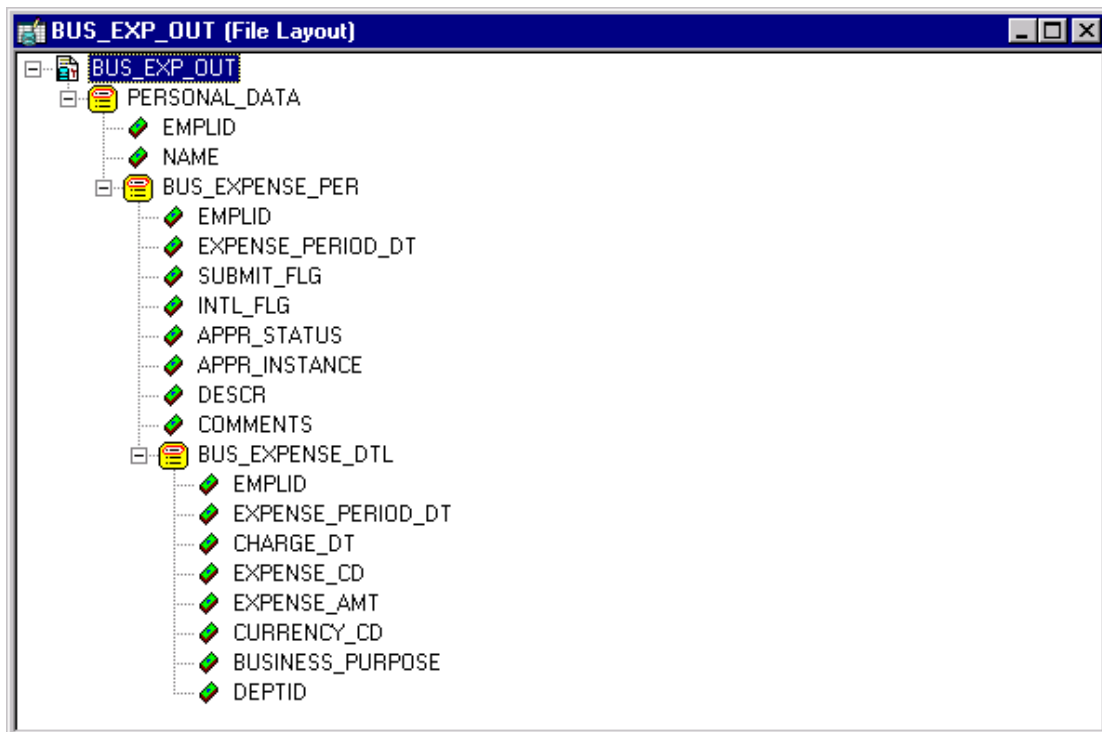
&rsBusExpDtl = CreateRowset(Record.BUS_EXPENSE_DTL);
&rsBusExpPer = CreateRowset(Record.BUS_EXPENSE_PER, &rsBusExpDtl);
&rsBusExp = CreateRowset(Record.PERSONAL_DATA, &rsBusExpPer);
```

Another variation is

```
&rsBusExp = CreateRowset(Record.PERSONAL_DATA,
CreateRowset(Record.BUS_EXPENSE_PER, CreateRowset(Record.BUS_EXPENSE_DTL)));
```

Using Standalone Rowsets to Write a File

The following example writes a file using a file layout that contains parent-child records.



File layout example

```

Local File &MYFILE;
Local Rowset &rsBusExp, &rsBusExpPer, &rsBusExpDtl;
Local Record &rBusExp, &rBusExpPer, &rBusExpDtl;
Local SQL &SQL1, &SQL2, &SQL3;

&rBusExp = CreateRecord(Record.PERSONAL_DATA);
&rBusExpPer = CreateRecord(Record.BUS_EXPENSE_PER);
&rBusExpDtl = CreateRecord(Record.BUS_EXPENSE_DTL);

&rsBusExp = CreateRowset(Record.PERSONAL_DATA,
CreateRowset(Record.BUS_EXPENSE_PER, CreateRowset(Record.BUS_EXPENSE_DTL)));
&rsBusExpPer = &rsBusExp.GetRow(1).GetRowset(1);

&MYFILE = GetFile("c:\temp\BUS_EXP.out", "W", %FilePath_Absolute);
&MYFILE.SetFileLayout(FileLayout.BUS_EXP_OUT);

&EMPLID = "8001";

&SQL1 = CreateSQL("%selectall(:1) where EMPLID = :2", &rBusExp, &EMPLID);
&SQL2 = CreateSQL("%selectall(:1) where EMPLID = :2", &rBusExpPer, &EMPLID);

While &SQL1.Fetch(&rBusExp)
    &rBusExp.CopyFieldsTo(&rsBusExp.GetRow(1).PERSONAL_DATA);
    &I = 1;
    While &SQL2.Fetch(&rBusExpPer)
        &rBusExpPer.CopyFieldsTo(&rsBusExpPer(&I).BUS_EXPENSE_PER);
        &J = 1;
        &SQL3 = CreateSQL("%selectall(:1) where EMPLID = :2 and
EXPENSE_PERIOD_DT = :3", &rBusExpDtl, &EMPLID,
&rsBusExpPer(&I).BUS_EXPENSE_PER.EXPENSE_PERIOD_DT.Value);
        &rsBusExpDtl = &rsBusExpPer.GetRow(&I).GetRowset(1);
        While &SQL3.Fetch(&rBusExpDtl)
            &rBusExpDtl.CopyFieldsTo(&rsBusExpDtl(&J).BUS_EXPENSE_DTL);
            &rsBusExpDtl.InsertRow(&J);
            &J = &J + 1;
        End-While;

        &rsBusExpPer.InsertRow(&I);
        &I = &I + 1;
    End-While;
    &MYFILE.WriteRowset(&rsBusExp);
End-While;
&MYFILE.Close();

```

The previous code generates the following output file.

```
AA8001      Schumacher,Simon
```

BB8001	06/11/1989YNA0	Customer Go-Live Celebration
CC8001 10100	06/11/198906/01/198908226.83	USDEntertain Clients
BB8001	08/31/1989YNA0	Customer Focus Group Meeting
CC8001 10100	08/31/198908/11/1989012401.58	USDCustomer Visit
CC8001 10100	08/31/198908/12/198904250.48	USDCustomer Visit
CC8001 10100	08/31/198908/12/198902498.34	USDCustomer Visit
BB8001	03/01/1998YYP0	Attend Asia/Pacific Conference
CC8001 00001	03/01/199802/15/1998011200	USDConference
CC8001 00001	03/01/199802/16/19980220000	JPYConference
BB8001	05/29/1998NNP0	Annual Subscription
CC8001 10100	05/29/199805/29/199814125.93	USDSoftware, Inc.
BB8001	08/22/1998NNP0	Regional Users Group Meeting
CC8001 10100	08/22/199808/22/19981045.69	USDDrive to Meeting
CC8001 10100	08/22/199808/22/19980912.44	USDCity Parking
BB8001	12/12/1998NNP0	Customer Visit: Nevco
CC8001 00001	12/12/199812/02/199801945.67	USDCustomer Feedback
CC8001 00001	12/12/199812/02/19981010.54	USDTo Airport
CC8001 00001	12/12/199812/03/19980610	USDAirport Tax
CC8001 00001	12/12/199812/03/199804149.58	USDCustomer Feedback
CC8001 00001	12/12/199812/04/1998055.65	USDCheck Voicemail
CC8001 00001	12/12/199812/04/19980988	USDAirport Parking
CC8001 00001	12/12/199812/04/199802246.95	USDCustomer Feedback
CC8001 00001	12/12/199812/04/199803135.69	USDCustomer Feedback

Using Standalone Rowsets to Read a File

Use the following code to read in the file created previously and insert the rows into the database.

```

Local File &MYFILE;
Local Rowset &rsBusExp, &rsBusExpPer, &rsBusExpDtl;
Local Record &rBusExp, &rBusExpPer, &rBusExpDtl;
Local SQL &SQL1;
```

```

&rBusExp = CreateRecord(Record.PERSONAL_DATA);
&rBusExpPer = CreateRecord(Record.BUS_EXPENSE_PER);
&rBusExpDtl = CreateRecord(Record.BUS_EXPENSE_DTL);

&rsBusExp = CreateRowset(Record.PERSONAL_DATA,
CreateRowset(Record.BUS_EXPENSE_PER, CreateRowset(Record.BUS_EXPENSE_DTL)));

&MYFILE = GetFile("c:\temp\BUS_EXP.out", "R", %FilePath_Absolute);
&MYFILE.SetFileLayout(FileLayout.BUS_EXP_OUT);

&SQL1 = CreatesQL("%Insert(:1)");

&rsBusExp = &MYFILE.ReadRowset();
While &rsBusExp <> Null;
    &rsBusExp.GetRow(1).PERSONAL_DATA.CopyFieldsTo(&rBusExp);
    &rsBusExpPer = &rsBusExp.GetRow(1).GetRowset(1);
    For &I = 1 To &rsBusExpPer.ActiveRowCount
        &rsBusExpPer(&I).BUS_EXPENSE_PER.CopyFieldsTo(&rBusExpPer);
        &rBusExpPer.ExecuteEdits(%Edit_Required);
        If &rBusExpPer.IsEditError Then
            For &K = 1 To &rBusExpPer.FieldCount
                &MYFIELD = &rBusExpPer.GetField(&K);
                If &MYFIELD.EditError Then
                    &MSGNUM = &MYFIELD.MessageNumber;
                    &MSGSET = &MYFIELD.MessageSetNumber;
                End-If;
            End-For;
        Else
            &SQL1.Execute(&rBusExpPer);
            &rsBusExpDtl = &rsBusExpPer.GetRow(&I).GetRowset(1);
            For &J = 1 To &rsBusExpDtl.ActiveRowCount
                &rsBusExpDtl(&J).BUS_EXPENSE_DTL.CopyFieldsTo(&rBusExpDtl);
                &rBusExpDtl.ExecuteEdits(%Edit_Required);
                If &rBusExpDtl.IsEditError Then
                    For &K = 1 To &rBusExpDtl.FieldCount
                        &MYFIELD = &rBusExpDtl.GetField(&K);
                        If &MYFIELD.EditError Then
                            &MSGNUM = &MYFIELD.MessageNumber;
                            &MSGSET = &MYFIELD.MessageSetNumber;
                        End-If;
                    End-For;
                Else
                    &SQL1.Execute(&rBusExpDtl);
                End-If;
            End-For;
        End-For;
    End-For;

```

```

        End-If;
    End-For;
    &rsBusExp = &MYFILE.ReadRowset();
End-While;
&MYFILE.Close();

```

Using Errors and Warnings

For the most part, errors and warnings display messages to users informing them about invalid data. For this reason, they are almost always placed in FieldEdit or SaveEdit PeopleCode, or in SearchSave PeopleCode for validation during search processing. In conjunction with edits, errors stop processing, while warnings allow processing to continue. When errors and warnings appear in places other than FieldEdit or SaveEdit, their effects vary.

In this section, we discuss the following topics:

- Syntax of errors and warnings.
- Errors, warnings, and edits.
- Errors and warnings in FieldEdit.
- Errors and warnings in SaveEdit.
- Errors and warnings in RowSelect.
- Errors and warnings in RowDelete.
- Errors and warnings in other events.

Syntax of Errors and Warnings

Errors and warnings require only a message that the Component Processor displays to users. You can code the message into the error or warning statement, or you can use Message Catalog. Peoplesoft recommends using Message Catalog with the MsgGet, MsgGetExplainText, and so on.

Error and warning use the same syntax. For example:

```

Error MsgGet(11100, 180, "Message not found.");
Warning MsgGet(11100, 180, "Message not found.");

```

Errors, Warnings, and Edits

You can use the following PeopleCode events for validation edits: FieldEdit and SaveEdit. The Component Processor applies FieldEdit when the user changes a field, and SaveEdit when the user saves the component. Errors and warnings in these events display a message to the

user. Most errors and warnings appear in these event types, although you can use them elsewhere.

For `FieldEdit`, you can use either the record field or component record field event. Remember that the record field event for each record fires before the component record field event for that record. For `SaveEdit`, you can use the record field or the component record event.

Remember that all record field events for a record fire before the component record events.

Errors and Warnings in `FieldEdit`

An error in `FieldEdit` prevents the system from accepting the new value of a field. The Component Processor highlights the offending field. The user must either change the field back to its original value or to something else which does not trigger the error. A warning enables the Component Processor to accept the new data. The Component Processor does not highlight any field that has warnings.

Errors and Warnings in `SaveEdit`

An error in `SaveEdit` prevents the system from saving any row of data. The Component Processor does not update the database for any field if one field has an error. Although the Component Processor displays an error message, it does not turn any field red. Unlike `FieldEdit` errors, `SaveEdit` errors can happen anywhere on the page or component, for any row of data. The data causing the error may appear on a different page within the same group, or a row of data not currently displayed. If this is the case, the field in error is brought into view by the system.

A warning in `SaveEdit` also is applied to all data in the page or component, but the Component Processor will accept the data, if told to by the user. In a `FieldEdit` warning, the Component Processor displays a message box with the text and two pushbuttons—OK and the standard Explain (the Explain pushbutton returns an explanation for the last message retrieved with the `MsgGet` function). In a `SaveEdit` warning, though, the message box contains an additional pushbutton, Cancel. OK accepts the data, overriding the warning and continuing the save process. Cancel ends the save process.

Because errors and warnings apply to all rows of data and all pages in a group, you must provide the user explicit information about what caused the error. Typically, you use the message catalog function to store messages and substitute variables into them. However, you can also facilitate this by concatenating in a field value. For example, if you have a stack of historical data on the page, you could use the following error statement:

```
Error ("The value exceeds the maximum on "|effdt|".");
```

Errors and Warnings in RowSelect

Note. Errors and warnings should no longer be used in RowSelect processing: instead, use **DiscardRow** and **StopFetching**. The behavior of **Error** and **Warning** in RowSelect PeopleCode, as described here, is retained for compatibility with previous releases of PeopleTools.

RowSelect PeopleCode filters out rows of data after the system applies search record criteria. It also can stop the Component Processor from reading additional rows of data.

A warning causes the Component Processor to reject the current row, but the Component Processor does continue reading more data. An error prevents any more data coming into the page or component. The Component Processor accepts the row that causes the error, but doesn't read any more data. If you want to reject the current row and stop loading additional rows, you must issue a warning and an error.

You must specify text for an error or warning, but the Component Processor does not display messages from RowSelect. You can still use the message text as a way of documenting the program.

See Also

"Understanding PeopleCode and Events"

PeopleTools PeopleBook: PeopleCode Reference, "PeopleCode Built-in Functions,"
DiscardRow, StopFetching

Errors and Warnings in RowDelete

When you delete a row of data (F8), the system prompts you to confirm. If you do confirm, RowDelete PeopleCode takes place, that is, any record field RowDelete PeopleCode fires, and any component record RowDelete PeopleCode. Errors and warnings in RowDelete display a message box.

A warning from RowDelete presents two choices—accept the RowDelete (the OK pushbutton), or cancel the RowDelete (the Cancel pushbutton). Perhaps after they read the warning message, the user will appreciate the RowDelete. An error from RowDelete PeopleCode prevents the Component Processor from removing that row of data from the page.

Errors and Warnings in Other Events

You should not put errors or warning in PeopleCode attached to any of the remaining events (FieldDefault, FieldFormula, RowInit, FieldChange, RowInsert, SavePreChange, WorkFlow, and SavePostChange). However, the Component Processor may issue its own errors and warnings when performing PeopleCode. When this happens, it means the Component Processor tried to do something and failed, and it cannot recover from the error.

These Event types activate processing that an user has no direct control over. So, if the Component Processor comes across an error condition, the user cannot fix it. The Component Processor cancels the transaction to avoid unpredictable results.

See Also

PeopleTools PeopleBook: PeopleCode Reference, "PeopleCode Built-in Functions," Warning, Error

Using RemoteCall

RemoteCall is a PeopleTools feature that enables executing a COBOL program remotely from within a PeopleSoft application. Remote calls are made using the **RemoteCall** PeopleCode function.

Because all PeopleCode runs on the application server, the **RemoteCall** PeopleCode function has more limited utility. However, RemoteCall can still be useful, because it enables you to take advantage of existing COBOL processes.

The **RemoteCall** function:

- Is a synchronous call. The PeopleSoft system passes parameters to the remote program, and then waits while the program runs. When the remote program is done, it returns any results or status information to the client, which then resumes execution. This means that **RemoteCall** is a "think-time" function and subject to certain restrictions.
- Is designed for fast response time, and has an API that provides programs with the response time needed for transaction processing.
- Can be executed with a command pushbutton or from a pop-up menu.
- **RemoteCall** has no scheduling or multi-step job capabilities. Each execution of RemoteCall is independent of others.
- Enables you to reuse existing COBOL code.
- For PeopleTools 8, you can no longer use **RemoteCall** to execute an Application Engine program. Use the **CallAppEngine** function instead.

In this section, we discuss:

- RemoteCall components.
- RemoteCall and PeopleSoft Process Scheduler.
- RemoteCall modifications for PeopleSoft Process Scheduler..
- RemoteCall programming guidelines.

See Also

"Using Methods and Built-in Functions," Think-Time Functions

PeopleTools PeopleBook: PeopleCode Reference, "PeopleCode Built-in Functions," CallAppEngine

RemoteCall Components

The RemoteCall PeopleTools feature consists of the following components:

- PeopleCode API.

This interface consists of the **RemoteCall** PeopleCode function. It is used from PeopleCode to start a remote program and process any results. The PeopleCode program doesn't include any special code to specify where the remote program is executed. TUXEDO can be configured to locally execute the program for testing purposes.

- Remote Program API.

This is used by the remote COBOL program to receive or pass parameters and return status information.

- PeopleSoft RemoteCall Service.

The PeopleSoft application server, PSAPPSRV, advertises the RemoteCall service. The service receives requests from clients and starts the requested program. When the program is completed, it passes the parameters and status code back to the client.

- TUXEDO.

TUXEDO is a message-based transaction monitor for distributed applications. No direct TUXEDO calls need to be implemented in PeopleCode or remote programs.

PeopleCode API

You can execute the **RemoteCall** function from PeopleCode associated with any Component Processor event except SavePostChange, SavePreChange, Workflow, RowSelect, or in any PeopleCode event resulting from a **ScrollSelect** or related function call. However, remote programs that change data should not be run as part of a SaveEdit process, because the remote program may complete successfully even though an error occurs later in the save process.

The preferred method of calling a remote program that changes data is in FieldChange PeopleCode in a record field associated with a command pushbutton, or from a pop-up menu item.

You should not use **RemoteCall** if you expect the remote program to return a large amount of data to the client, because data is passed back only through the parameters of the PeopleCode API.

Authorization

Authorization to run a remote program is like authorization to run a PeopleCode program. Because a remote program is started from PeopleCode, the user has authorization in User Security to use the page that executes the PeopleCode.

Unit of Work

The remote program runs in a different unit of work from the page. A commit is issued by PeopleTools if needed on the client before **RemoteCall** is called. This means that, by default, the remote program does not know about any database changes unless the page is saved before the program is called. After the remote program starts, it runs to completion and commits or ends before returning to the page. In this way, the remote program and the page do not have locking contention. To ensure that the save has actually been done, use the **DoSaveNow** built-in function.

Error Processing

When using RemoteCall to execute a COBOL program, two types of errors can occur:

- PeopleTools errors.

An error could occur in PeopleTools or TUXEDO, or the service might not be found. These are treated as hard errors by PeopleCode. An error message box displays, and that piece of PeopleCode is terminated. So, in the case of a PeopleTools error, the remote program always either returns a Return code of zero, or terminates with a message due to a system error.

- Application-specific errors.

Any error information specific to the remote application must be passed back in regular data variables, and the application can process these in an application-specific way. If you have a status code on which the application depends, you should initialize it to an invalid value to be sure the COBOL program does return the status code.

Client Execution Environment

Because the remote program is executed synchronously, clients receive an hourglass icon and cannot do anything in the current window until the remote application completes. They could move to another window and do processing there; or they could start another PeopleSoft window. They cannot cancel the remote program after it starts. If the program does not terminate in a timely fashion (as determined by the RemoteCall timeout set with PeopleSoft Configuration Manager), RemoteCall attempts to terminate the process and returns an error indicating that the program was terminated.

Remote Program API

The Remote Program API provides the functions to get and put data between the network and the COBOL program. These functions are implemented in C, but are callable from COBOL through the PTPNETRT program. For an example, see the PTPNTEST.CBL program.

Note. If these APIs are called when the program is not running as a remote program, ACTION-GET and ACTION-PUT return an error. All other actions return without doing anything.

Error Processing

If an unexpected error is found, call PTPNETRT with ACTION-RESET, then with ACTION-PUT to send back any error status variables, then with ACTION-DONE to send the buffer.

PeopleSoft RemoteCall Service

The RemoteCall Service serves as a bridge between the PeopleCode API and the remote COBOL programs. In PeopleTools 6, RemoteCall was the only PeopleSoft TUXEDO service. In PeopleTools 7, RemoteCall is one of many services advertised from the PSAPPSRV TUXEDO server, and can be configured as part of the standard domain setup and administration.

The client sends the RemoteCall service request, consisting of the connect information and the program name, as well as any other parameters for the program, to the application server. The RemoteCall service then executes the program and passes it the connect string.

RemoteCall and PeopleSoft Process Scheduler

COBOL application programs initiated by the RemoteCall service use the same COBOL application architecture used by PeopleSoft Process Scheduler. After being initiated by the Dispatcher, COBOL application programs call the COBOL SQL API program, PTPSQLRT, to connect to the RDBMS to compile and execute SQL statements. You can design and implement COBOL programs to be understood by both PeopleSoft Process Scheduler and RemoteCall.

The following guidelines help you to select the optimal method for running a particular COBOL program:

- Use PeopleSoft Process Scheduler for asynchronous processes, or processes that can be scheduled, are multi-step, or that require printed output.
- Use RemoteCall for synchronous processes that are quick ("transaction processing" types of processes).

RemoteCall Modifications for PeopleSoft Process Scheduler

To enable an existing program that runs under PeopleSoft Process Scheduler to run under RemoteCall as well, make the following changes:

- Include the PTCNETRT copy member.
- Include the PTCNCHEK member before the connection call to PTPSQLRT.
- Add the call to PTPNETRT ACTION-DONE just before the program terminates (after the call to disconnect from the database). This should be conditional on whether you are RUNNING-REMOTE-CALL.

- If you are running as a RemoteCall, ensure that PROCESS-INSTANCE OF PRUNSTATUS is not set. Otherwise your calls to PTCPSTAT will try to update the PSPRCSRQST table. This will not cause an error, but it is unnecessary processing.

This program can now run from PeopleSoft Process Scheduler or from RemoteCall. Note that if a program has to pass parameters, it must have RemoteCall-specific ACTION-GET and ACTION-PUT calls.

RemoteCall Programming Guidelines

Keep the following points in mind when using RemoteCall:

- Do not use RemoteCall for long-running batch jobs. As a general rule, if you think execution will take more than 15 seconds, you should not be using RemoteCall, but should instead use PeopleSoft Process Scheduler.
- RemoteCall is meant for running jobs on the server. It should not be used to invoke client-only programs. Support for local calling with RemoteCall is provided solely as a debugging and development aid. For client-only programs, use **Declare Function**, then call the external function from a library.
- If you do not want to modify an existing program, then pass only the program name and run control, and do not return any parameters. This way, the program requires few changes to run as a remote function.

CHAPTER 9

Referencing Data in the Component Buffer

This chapter provides an overview of the Component Buffer and discusses how to:

- Specify data with contextual references.
- Specify data with references using scroll path syntax.

Understanding Component Buffer Structure and Contents

PeopleCode frequently must refer to data in the Component Buffer—the area in memory that stores data for the currently active component.

There are two methods of specifying a piece of data in the Component Buffer from within PeopleCode:

- contextual references, which refer to data relative to the location of the currently executing PeopleCode program.
- references using scroll path syntax, which provide a complete—or absolute—path through the Component Buffer to the referenced component.

In addition to the built-in functions used to access the Component Buffer, PeopleCode provides enhanced access to structured data buffers using the new object syntax. Use the object-oriented PeopleCode to resolve contextual ambiguities when you reference a nonprimary record field that appears on more than one scroll level in a component. As with built-in functions, object syntax provides for both relative and absolute references to Component Buffer data.

See *Understanding Objects and Classes in PeopleCode* for information about object-oriented programming. The Component Buffer consists of rows of buffer fields that hold data for the various records associated with page controls, including primary scroll records, related display records, derived/work records, and translate table records. PeopleCode can reference buffer fields associated with page controls and other buffer fields from the primary scroll record and related display records.

See *What Record Fields Are in the Component Buffer?* for details.

Primary scroll records are the principal SQL table or view associated with a page scroll level. A primary scroll record uniquely identifies a scroll level in the context of its page: each scroll level can have only one primary scroll record; and the same primary scroll record cannot occur

on more than one scroll at the same level of the page. Parent-child relations among primary scroll records determine the dependency structure of the scrolls on the page. That is, the primary record on a level 1 scroll must be a child of the primary record on level 0; the primary record on a level 2 scroll must be a child of the primary record on its enclosing level 1 scroll; and the primary record on a level 3 scroll must be a child of the primary record on its enclosing level 2 scroll.

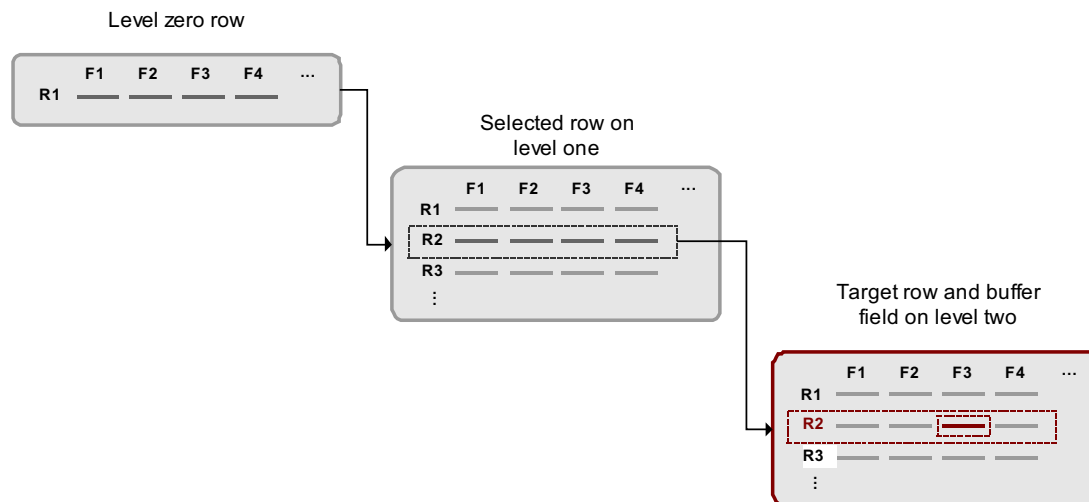
Note. Level 0 may have multiple records.

The hierarchical relations among scrolls, controlled by hierarchical relations among primary scroll records, enable the end-user and PeopleCode to drill down through the scroll hierarchy to access any specific buffer field, including related display, derived/work, and translate table buffer fields, which occupy space on the same rows as the primary scroll record buffer fields with which they are associated.

For example, to access a page field on level 2 of a page, the end-user must:

- Select a field on level 1 of the page.
- Scroll to and select the desired field on level 2 of the page.

The following diagram illustrates this *scroll path* taken by the user:



Scroll path to a buffer field

To access the same field in the Component Buffer, PeopleCode must:

1. Specify a scroll and row on scroll level 1: this selects a subset of dependent rows on level 2.
2. Specify a scroll and row on scroll level 2.
3. Specify the specific recordname.fieldname on the level 2 row.

PeopleCode Component Buffer functions use a common *scroll path syntax* for locating scrolls, rows, and fields in multiple-scroll pages.

In this section, we discuss how to:

- Compare rowsets to scrolls.
- Understand what record fields are in the Component Buffer.

See Also

References Using Scroll Path Syntax and Dot Notation for details about PeopleCode scroll path syntax.

Comparing Rowsets to Scrolls

The rowset is a new programming construct in PeopleTools 8 which enables more consistent, more convenient, and less ambiguous manipulation of buffer data than existing built-in functions can achieve. It's a hierarchical data object that can represent an entire scroll and all of its subordinate scrolls.

A rowset can contain the entire contents of a Component Buffer, or the contents of any lower level scroll plus all of its subordinate buffer data. The hierarchical structure of component levels—scroll, row, record, field—is provided by the new object data types, *Rowset*, *Row*, *Record* and *Field* respectively.

You can access any specific rowset, row, record, or field within the buffer using the dot notation inherent in PeopleTools 8 object-oriented programming. This enables you to reference fields within a record object, records within a row object, and rows within a rowset object as properties of the parent objects.

See Also

"Data Buffer Access"

"Rowset Class"

Understanding Objects and Classes in PeopleCode

What Record Fields Are in the Component Buffer?

The record fields in the Component Buffer are a superset of those accessible to the end user through page controls. In most cases PeopleCode can reference any record field in a scroll's primary scroll record or in a related display record—not just those fields that are associated with page controls. The following table provides specific information for various record types and locations:

Type and Location of Record	Presence in Component Buffer
Primary record on scroll levels greater than zero	On scroll levels greater than zero all record fields from the primary scroll record are in the Component Buffer. This means that on levels greater than zero PeopleCode can refer to any record field on the primary scroll record, even if it is not associated with a page control.
Primary record on scroll level 0	If scroll level 0 of a page contains only controls associated with primary scroll record fields that are search keys or alternate search keys, then only the search key and alternate search key fields will be in the Component Buffer: <i>not</i> the entire record. The values for the fields come from the keylist, and the record won't run RowInit PeopleCode. If level 0 contains at least one record field from the primary scroll record that is <i>not</i> a search key or alternate search key, then <i>all</i> the record fields from the primary scroll record will be available in the buffer. (For this reason you may sometimes need to add one such record field at level 0 of the page to make sure that all the record fields of the level-zero primary record can be referenced from PeopleCode.)
Related display record fields	The buffer contains the related display record field, plus any record fields from the related display record that are referenced by PeopleCode programs. This means that you can reference any record field in a related display record.
Derived/work record fields	Only derived/work record fields associated with page controls are in the Component Buffer. Other record fields from the derived/work record cannot be referenced from PeopleCode.
Translate Table record fields	Only Translate Table fields associated with page controls are available in the Component Buffer. Other fields from the Translate Table cannot be referenced from PeopleCode.

Note. There is a special limitation on RowSelect PeopleCode. In RowSelect PeopleCode, you can refer only to record fields on the record that is currently being processed.

Specifying Data With Contextual References

In a *contextual reference* PeopleCode refers to a row or buffer field determined by the current context; that is, the context in which the PeopleCode program is currently executing.

In this section, we provide an overview of the current context and discuss the following topics:

- Contextual row references.
- Contextual buffer field references.

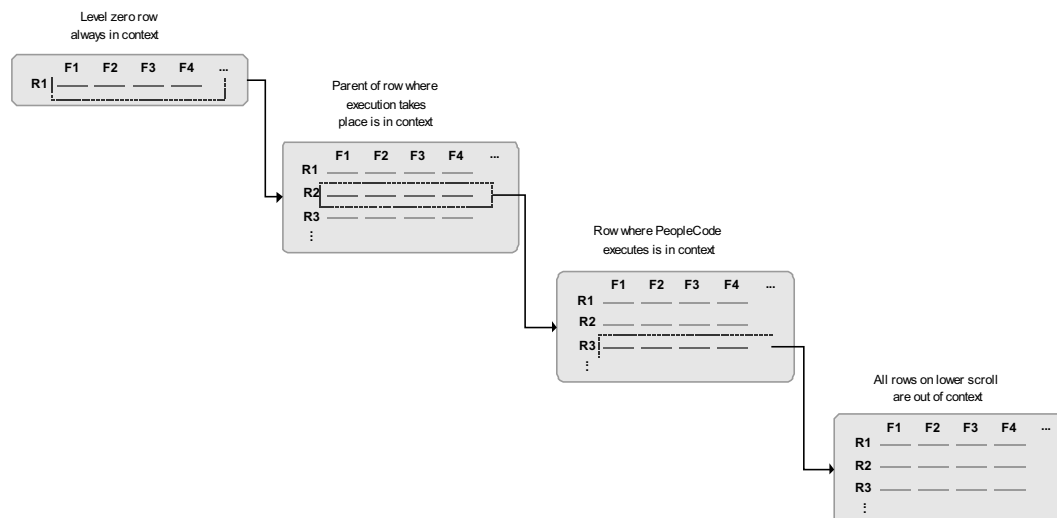
Understanding Current Context

All PeopleCode programs, with the exception of programs associated with standard menu items, execute in a *current context*. The current context determines which buffer fields can be contextually referenced from PeopleCode, and which row of data is the "current row" on each scroll level at the time a PeopleCode program is executing.

The current context is comprised of a subset of the buffer fields in the Component Buffer, determined by the row of data where a PeopleCode program is executing. The current context includes:

- All buffer fields in the row of data where the PeopleCode program is executing.
- All buffer fields in rows that are hierarchically superior to the row where the PeopleCode program is executing.

In the following diagram all rows enclosed in dotted rectangles are part of the current context:



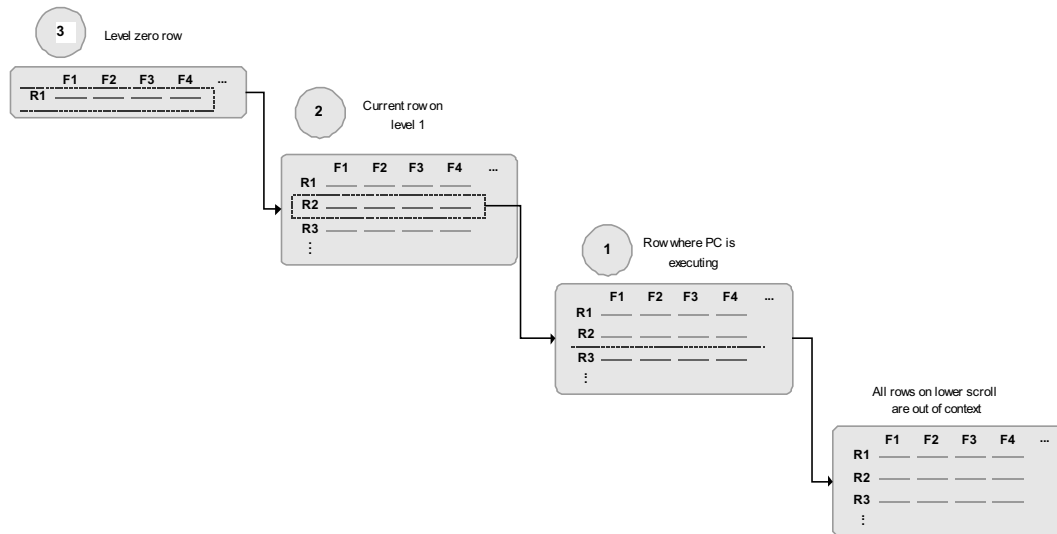
Context of PeopleCode executing on a level 2 scroll

In this example a PeopleCode program is executing in a buffer field on row R3 on scroll level 2. The rows in scroll level 2 are dependent on row R2 on scroll level 1. The rows in scroll level 1 are dependent on the single row at scroll level 0. The current context consists of all the buffer fields at level 2 row R3, level 1 row R2, and level 0 row R1. The rows in the current context on levels one and two are the *current rows* on their respective scrolls. The single row on level 0 is always current, and is included in any current context. All rows other than the current rows and the level 0 row are outside the current context. Note that no current row can be determined on scrolls below the one where the PeopleCode is executing.

With PeopleTools 8, contextual references work within the structure of a rowset object, and can include references to all the field objects, record objects, row objects, and rowset objects in the current context.

Contextual Reference Processing Order

PeopleCode resolves contextual references at runtime by first checking the row where the PeopleCode program is executing; if it doesn't find an appropriate buffer field, it looks in progressively higher rows in the current context. The following diagram indicates this processing order:



Processing order of a contextual reference

In typical pages this processing order is not significant; however, there are some cases where the same record occurs on more than one level of a page. In these cases, it's important to understand exactly how the direct reference is resolved.

See Also

Contextual Buffer Field Reference Ambiguity

Contextual Row References

A contextual row reference refers to a row in the current context on level 1 or lower in the page. Because each scroll uses a unique primary record, the name of that record uniquely identifies whichever row is in the current context for that scroll level. A contextual row reference uses a **RECORD.recordname** component name reference to specify the scroll level of the intended row, resulting in a reference to the current row at the specified scroll level.

For example, you can use contextual row references with the **RecordDeleted**, **RecordNew**, and **RecordChanged** functions:

```
If RecordDeleted(RECORD.SOME_REC) Then...
```

With PeopleTools 8 object-oriented programming, the desired row can be referenced by specifying parent rows or rowsets of the current rowset:

```
If GetRowSet().ParentRowset.ParentRow.IsDeleted Then...
```

In early versions of PeopleTools you could make contextual row references using a *recordname.fieldname* expression:

```
HideRow(SOME_REC.ANY_FIELD)

If RecordDeleted(SOME_REC.ANY_FIELD) Then...
```

This syntax is still supported.

See Also

Understanding Current Context

Contextual Buffer Field References

A *contextual buffer field reference* is a type of PeopleCode expression that refers to a buffer field by specifying a record field. The row of the buffer field is determined by the current context of the PeopleCode program where the reference is made. You can use a contextual buffer field reference to retrieve or update the value in the buffer field, to pass the buffer field value to a function, or to reference an instance of a page control associated with the buffer field. The following statements use contextual buffer field references:

```
SOME_RECORD.SOME_FIELD = &VAL; /* Assigns value of variable to buffer field */
&VAL = SOME_RECORD.SOME_FIELD; /* Assigns value of buffer field to variable */
Hide(SOME_RECORD.SOME_FIELD); /* Hides instance of control associated with
buffer field */
```

With PeopleTools 8 object-oriented programming, a field object incorporates information about both the record field on which the buffer field is based, and the page control with which the buffer field is associated. By referring to the field object, you either make a contextual buffer field reference or you change an interface attribute of the associated page control, depending on the object property you use. The following example has the same effect as a contextual buffer field reference:

```
&MYFIELD.Value = &SOMEVAL; /* Assigns value of a variable to a buffer field
*/
```

Contextual Buffer Field Reference Ambiguity

Nonprimary record fields may appear on more than one scroll level in a page. For example, a page may use a derived/work field DERIVED_JS.CALC_1 as a work field on level 1 and level 2 of the same page. This creates distinct DERIVED_JS.CALC_1 buffer fields for rows on both levels. Because of the order in which PeopleCode resolves contextual buffer field references, if the following contextual reference executes in a PeopleCode program on a level-two row:

```
&VAL = DERIVED_JS.CALC_1;
```

it will always retrieve the buffer field value on the current row on level 2. PeopleCode on level 2 would be unable to retrieve the value of the DERIVED_JS.CALC_1 on level 1 using a contextual reference.

See Contextual Reference Processing Order for details.

To explicitly reference the DERIVED_JS.CALC_1 buffer field on level 1, you must use a Component Buffer function with a scroll path:

```
&VAL = FetchValue(SCROLL.level1_scrollname, CurrentRowNumber(1),
    DERIVED_JS.CALC_1);
```

The **CurrentRowNumber** function returns the current row on level 1: that is, the parent row of the level 2 row where the PeopleCode program is executing.

Ambiguous Contextual References to Buffer Fields on Level 0

Level 0 of a page contains only a single row of data, and the buffer fields in this row are always in the current context. For this reason you can *almost* always refer to a level-zero buffer field using a contextual reference. However, there are unusual cases when referential ambiguity makes it impossible to reference a buffer field on level 0 contextually. For example, a page may use a derived/work field DERIVED_JS.CALC_1 as a work field on level 0 and level 1 of the same page. This creates distinct DERIVED_JS.CALC_1 buffer fields for rows on both levels. Because of the order in which PeopleCode resolves contextual field references, if the following contextual reference executes in a PeopleCode program on a level-one row:

```
&VAL = DERIVED_JS.CALC_1;
```

it will always retrieve the buffer field value on the current row on level 1.

To explicitly reference the DERIVED_JS.CALC_1 buffer field on level 0, you must use a Component Buffer function with this syntax:

```
Function([recordname.]fieldname, rownum)
```

Here **rownum**, because it is on level 0, is always equal to 1. In the example you use this statement:

```
&VAL = FetchValue(DERIVED_JS.CALC_1, 1);
```

Resolving Ambiguous References With Objects

With PeopleTools 8 object-oriented programming, even if two field objects which are in different rowsets contain buffer data that's based on the same underlying record field, references to those objects are inherently unique, because each is instantiated with respect to a specific point in the hierarchy of the buffer. Any manipulation of a field object's interface properties affects only the page control with which it's associated.

The following example instantiates a field object using the "long" form, to emphasize the hierarchical form of the data. It then hides the field's associated page control. Because this is a unique instance of the field, based on its hierarchy, hiding this field does not affect the visibility of any other page control associated with the same record field:

```

&MYFIELD =
GetRowset (SCROLL.EMPL_CHECKLIST) .GetRow (&I) .GetRecord (RECORD.EMPL_CHECKLIST) .G
etField (EMPL_CHECKLIST.EMPLID);
&MYFIELD.Visible = False;

/* the same code, using the "short" form */

&MYFIELD = GetRowset (SCROLL.EMPL_CHECKLIST) .GetRow (&I) .EMPL_CHECKLIST.EMPLID;

```

Note. Any change in a field object's value affects both the underlying record field and the value of any other field object-oriented on the same record field. This behavior is the same as the behavior of contextual buffer field references that alter the field value.

See Also

References Using Scroll Path Syntax and Dot Notation

Specifying Data With References Using Scroll Path Syntax and Dot Notation

A *scroll path* is a construction found in the parameter lists of many Component Buffer functions, which specifies a scroll level in the currently active page. Additional parameters are required to locate a row or a buffer field at the specified scroll level.

The scroll path construction was enhanced in PeopleTools 5 and again in PeopleTools 7.5. PeopleTools 7.5 and 8 support all previous versions of the scroll path syntax, and there is no requirement to update the syntax in existing applications.

See Scroll Path Syntax prior to PeopleTools 7.5 for details about the older variants of this construction.

PeopleTools 7.5 scroll path syntax enables you to eliminate ambiguous references, which, although rare, do sometimes occur in complex components.

See Contextual Buffer Field References for details about the problem of scroll path ambiguity.

PeopleTools 8 adds the convenience of object-oriented dot notation and default methods, which produce inherently nonambiguous references, to PeopleCode programs. There are examples of dot notation in this section and examples of the scroll path syntax available in PeopleTools 7.5, which is still valid in PeopleTools 8.

In this section, we discuss the following topics:

- Scroll path syntax in PeopleTools 7.5.
- Scroll level, row, and buffer field references.
- Scroll path syntax before PeopleTools 7.5.

See Also

Data Buffer Access

Scroll Path Syntax in PeopleTools 7.5

PeopleTools 7.5 offers two constructions for scroll paths: a standard scroll path syntax, which in all cases except data buffer references is identical to the syntax in PeopleTools 6 and 7; and an alternative syntax using a **SCROLL.scrollname** expression. The latter is more powerful in that it can process some rare cases where a **RECORD.recordname** expression results in an ambiguous reference.

Scroll Path Syntax With RECORD.recordname

Here is the standard scroll path syntax:

```
[RECORD.level1_recname, level1_row, [RECORD.level2_recname, level2_row, ]]  
RECORD.target_recname
```

This scroll path syntax is the same as in PeopleTools 6 and 7, except when the referenced component is a buffer field.

See Scroll Path Syntax prior to PeopleTools 7.5 for details about earlier versions of the scroll path syntax.

If the target level (the level you want to reference) is one, you must supply only the **RECORD.target_recname** parameter. If the target scroll level is greater than one, you must provide scroll name and row level parameters for all hierarchically superior scroll levels, beginning at level 1. The following table indicates the scroll path syntax for the three possible target scroll levels:

Target Level	Scroll Path Syntax
1	RECORD.target_recname
2	RECORD.level1_recname, level1_row, RECORD.target_recname
3	RECORD.level1_recname, level1_row, RECORD.level2_recname, level2_row, RECORD.target_recname

If you are referring to a row or a buffer field, additional parameters are required after the scroll path.

The following are the *standard* scroll path syntax parameters:

- RECORD.level1_recname** Specifies the name of a record associated with scroll level 1, normally the primary scroll record. This parameter is required if the target scroll level is two or three.
- level1_row** An integer that selects a row on scroll level 1. This parameter is required if the target scroll level is two or three.

RECORD.level2_recname	Specifies the name of a record associated with scroll level 2, normally the primary scroll record. This parameter is required if the target row is on scroll level 3.
level2_row	An integer that selects a row on scroll level 2. This parameter is required if the target row is on scroll level 3.
RECORD.target_recname	Specifies a record associated with the target scroll level, generally the primary scroll record. The scroll can be on level 1, two, or three of the active page.

Scroll Path Syntax With **SCROLL.scrollname**

As an alternative to **RECORD.recordname** expressions in scroll path constructions, PeopleTools 7.5 permits use of a new **SCROLL.scrollname** expression. Scroll paths using **SCROLL.scrollname** are functionally identical to those using **RECORD.recordname**, except that **SCROLL.scrollname** expressions are more strict: they can refer only to a scroll level's primary record; whereas **RECORD.recordname** expressions can refer to any record in the scroll level, which in some rare cases can result in ambiguous references. (This can occur, for example, if the **RECORD.recordname** expression inadvertently references a related display record in another page in the component.) Use of **RECORD.recordname** is still permitted, and there is no requirement to use the **SCROLL.scrollname** alternative unless it is needed to avoid an ambiguous reference.

The *scrollname* is the same as the scroll level's primary record name.

The scroll name *cannot* be viewed or changed through the Application Designer interface. Here is the complete scroll path syntax using **SCROLL.scrollname** expressions:

```
[SCROLL.level1_scrollname, level1_row, [SCROLL.level2_scrollname, level2_row,
]], SCROLL.target_scrollname
```

The target scroll level in this construction is the scroll level that you want to specify. If the target level is one, you need to supply only the **SCROLL.target_scrollname** parameter. If the target scroll level is greater than one, you need to provide scroll name and row level parameters for hierarchically superior scroll levels, beginning at level 1. The following table indicates the scroll path syntax for the three possible target scroll levels:

Target Level	Scroll Path Syntax
1	SCROLL.target_scrollname
2	SCROLL.level1_scrollname , level1_row, SCROLL.target_scrollname
3	SCROLL.level1_scrollname , level1_row, SCROLL.level2_scrollname , level2_row, SCROLL.target_scrollname

If the component you are referring to is a row or a buffer field, additional parameters are required after the scroll path.

The following are the *alternative* scroll path syntax parameters:

SCROLL.level1_scrollname	Specifies the name of the page's level-one scroll. This is always the same as the name of the scroll level's primary
---------------------------------	--

scroll record. This parameter is required if the target scroll level is two or three.

level1_row

An integer that selects a row on scroll level 1. This parameter is required if the target scroll level is two or three.

SCROLL.level2_scrollname

Specifies the name of the page's level 2 scroll. This is always the same as the name of the scroll level's primary scroll record. This parameter is required if the target row is on scroll level 3.

level2_row

An integer that selects a row on scroll level 2. This parameter is required if the target row is on scroll level 3.

SCROLL.target_scrollname

The scroll name of the target scroll level, which can be level 1, two, or three of the active page.

See Also

Scroll Level, Row, and Buffer Field References

Scroll Level, Row, and Buffer Field References

You can reference a scroll level using the ***scrollpath*** construct only. Functions that reference rows for buffer fields require additional parameters. The following table summarizes the three types of Component Buffer reference:

<i>Target Component</i>	<i>Reference Syntax</i>	<i>Example Function</i>
Scroll level	scrollpath	HideScroll(scrollpath);
Row	scrollpath, row_number	HideRow(scrollpath, row_number);
Field	scrollpath, row_number, [recordname.]fieldname	Fetchvalue(scrollpath, row_number, fieldname);

PeopleTools 8 provides an alternative to the scroll level, row, and field components in the form of the data buffer classes ***Rowset***, ***Row***, ***Record*** and ***Field***, which you reference using dot notation with object methods and properties. The following table demonstrates the syntax for instantiating and manipulating objects in the current context from these classes:

<i>Target Object</i>	<i>Example Instantiation</i>	<i>Example Operation</i>
Rowset	&MYROWSET = GetRowset();	&MYROWSET.Refresh();
Row	&MYROW = GetRow();	&MYROW.CopyTo(&SOMEROW);
Record	&MYRECORD = GetRecord();	&MYRECORD.CompareFields(&SOMERECORD);

Target Object	Example Instantiation	Example Operation
Field	<code>&MYFIELD = GetRecord().fieldname;</code>	<code>&MYFIELD.Label = "Last Name";</code>

The following sections provide examples of functions using scroll path syntax, which refer to an example page from a fictitious veterinary clinic database. The page has three scroll levels, shown in the following table:

Level	Scroll Name (= Primary Scroll Record Name)
0	VET
1	OWNER
2	PET
3	VISIT

The examples given for PeopleTools 8 object-oriented syntax assumes that the following initializing code has been executed:

```
Local Rowset &VET_SCROLL, &OWNER_SCROLL, &PET_SCROLL, &VISIT_SCROLL;

&VET_SCROLL = GetLevel0();
&OWNER_SCROLL = &VET_SCROLL.GetRow(1).GetRowSet(SCROLL.OWNER);
&PET_SCROLL = &OWNER_SCROLL.GetRow(2).GetRowSet(SCROLL.PET);
&VISIT_SCROLL = &PET_SCROLL.GetRow(2).GetRowSet(SCROLL.VISIT);
```

Referring to Scroll Levels

The **HideScroll** function provides an example of a reference to a scroll level. The syntax of the function is:

```
HideScroll(scrollpath)
```

where *scrollpath* is

```
[RECORD.level1_recname, level1_row, [RECORD.level2_recname, level2_row,]]
RECORD.target_recname
```

To reference the level 1 scroll in the example, use this syntax:

```
HideScroll(RECORD.OWNER);
```

This hides the OWNER, PET, and VISIT scrolls on the example page.

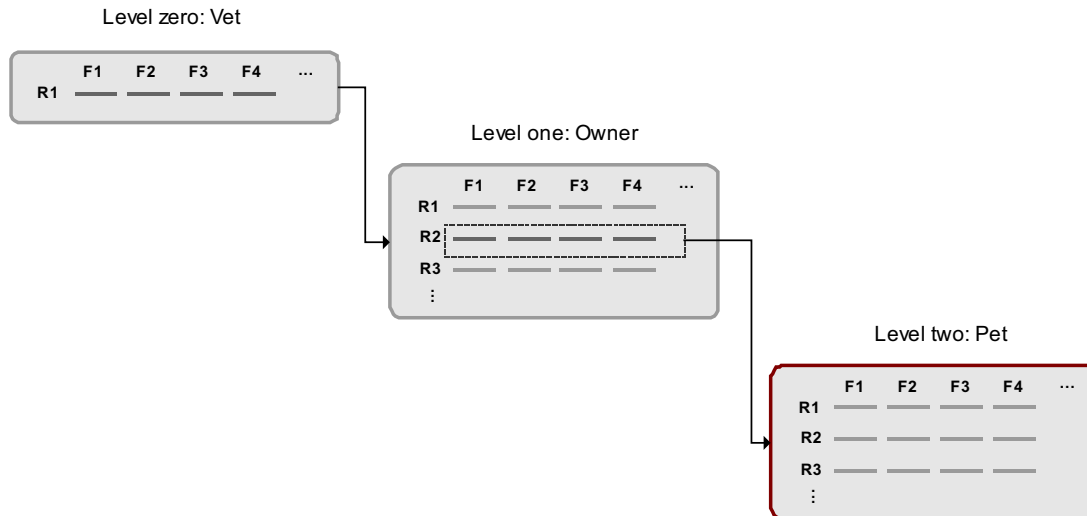
In PeopleTools 8, the object-oriented version of this is:

```
&OWNER_SCROLL.HideAllRows();
```

To hide scroll levels two and below, supply the primary record and row in scroll level 1, then the record identifying the target scroll:

```
HideScroll(RECORD.OWNER, &L1ROW, RECORD.PET);
```

The following diagram indicates the scroll path of this statement, assuming that the value of &L1ROW is 2:



Sample scroll path

Similarly, to hide the VISIT scroll on level 3, you specify rows on scroll levels one and two.

```
HideScroll(RECORD.OWNER, &L1ROW, RECORD.PET, &L2ROW, RECORD.VISIT);
```

If you wanted to use the new **SCROLL.scrollname** syntax, the previous example could be written as this:

```
HideScroll(SCROLL.OWNER, &L1ROW, SCROLL.PET, &L2ROW, SCROLL.VISIT);
```

In PeopleTools 8, the object-oriented version of this is:

```
&VISIT_SCROLL.HideAllRows();
```

Referring to Rows

Referring to rows is precisely the same as referring to scroll areas, except that you need to specify the row you want to select on the target scroll. As an example, let's examine the **HideRow** function, which hides a specific row in the level 3 scroll of the page:

```
HideRow(scrollpath, target_row)
```

To hide row number `&ROW_NUM` on level 1:

```
HideRow(RECORD.OWNER, &ROW_NUM);
```

To do the same using the **SCROLL.scrollname** syntax:

```
HideRow(SCROLL.OWNER, &ROW_NUM);
```

In PeopleTools 8, the object-oriented version of this for the OWNER rowset is:

```
&OWNER_SCROLL(&ROW_NUM).Visible = False;
```

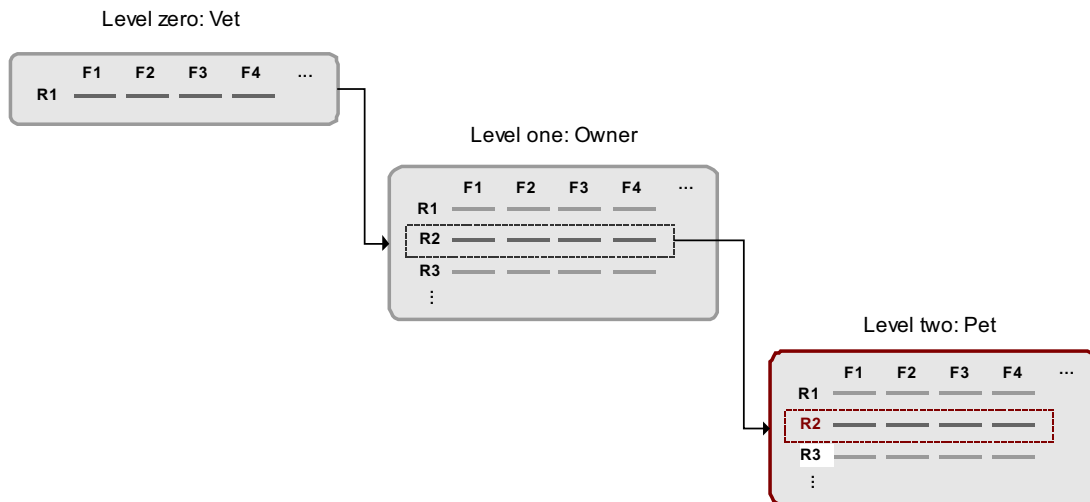
On level 2:

```
HideRow(RECORD.OWNER, &L1_ROW), RECORD.PET, &ROW_NUM);
```

In PeopleTools 8, the object-oriented version of this for the PET rowset is:

```
&PET_SCROLL(&ROW_NUM).Visible = False;
```

The following diagram indicates the scroll path of this statement, assuming that the value of &L1_ROW is 2 and that &ROW_NUM is equal to 2:



Scroll path statement

On level 3:

```
HideRow(RECORD.OWNER, CurrentRowNumber(1), RECORD.PET, CurrentRowNumber(2),  
RECORD.VISIT, &ROW_NUM);
```

In PeopleTools 8, the object-oriented version of this for the VISIT rowset is:

```
&VISIT_SCROLL(&ROW_NUM).Visible = False;
```

Referring to Buffer Fields

Buffer field references require a *[recordname.]fieldname* parameter to specify a record field: the combination of scroll level, row number, and record field name uniquely identifies the buffer field:

```
FetchValue(scrollpath, target_row, [recordname.]fieldname)
```

Assume, for example, that record definitions in the veterinary database have the following fields that you want to reference:

Record	Sample Field
OWNER	OWNER_NAME
PET	PET_BREED
VISIT	VISIT_REASON

You could use the following examples to retrieve values on levels 1, 2, or 3 from a PeopleCode program executing on level 0.

To fetch a value of the OWNER_NAME field on the current row of scroll area one:

```
&SOMENAME = FetchValue(RECORD.OWNER, &L1_ROW, OWNER.OWNER_NAME);
```

In PeopleTools 8, the object-oriented version of this for the OWNER rowset is:

```
&SOMENAME = &OWNER_SCROLL(&L1_ROW).OWNER.OWNER_NAME;
```

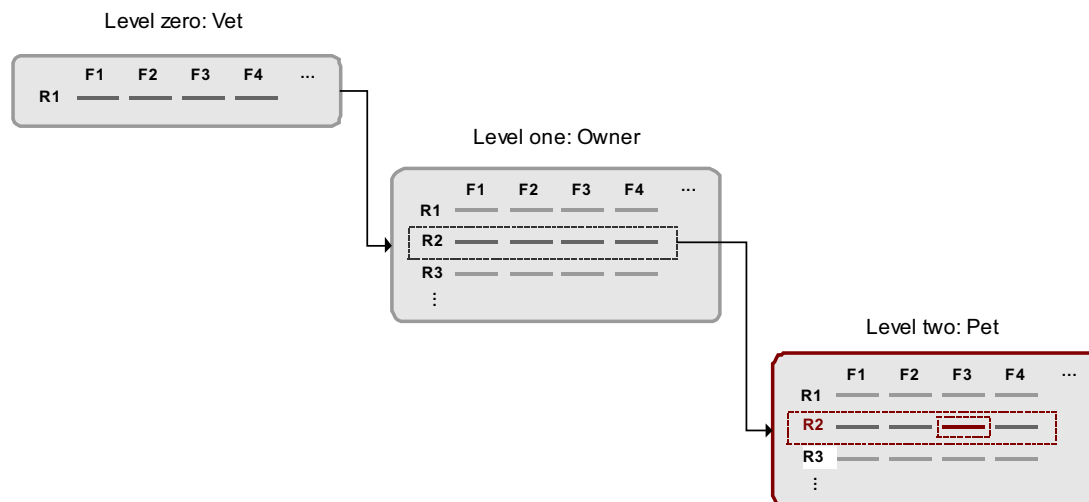
To fetch PET_BREED on level 2:

```
&SOMEBREED = FetchValue(RECORD.OWNER, &L1_ROW, RECORD.PET, &L2_ROW,
PET.PET_BREED);
```

In PeopleTools 8, the object-oriented version of this for the PET rowset is:

```
&SOMEBREED = &PET_SCROLL(&L2_ROW).PET.PET_BREED;
```

The following diagram indicates the scroll path to the target field, assuming that &L1_ROW equals 2, &L2_ROW equals 2, and field F3 is PET.PET_BREED:



Scroll path to target field

To fetch VISIT_REASON on level 3:

```
&SOMEREASON = FetchValue(RECORD.OWNER, &L1_ROW, RECORD.PET, &L2_ROW,
RECORD.VISIT, &L3_ROW, VISIT.VISIT_REASON);
```

To do the same using the *Scroll.scrollname* syntax:

```
&SOMEREASON = FetchValue(SCROLL.OWNER, &L1_ROW, SCROLL.PET, &L2_ROW,
    SCROLL.VISIT, &L3_ROW, SCROLL.VISIT_REASON);
```

In PeopleTools 8, the object-oriented version of this is:

```
&SOMEREASON = &VISIT_SCROLL(&L3_ROW).VISIT.VISIT_REASON;
```

Using CurrentRowNumber

The **CurrentRowNumber** function returns the current row, as determined by the current context, for a specific scroll level in the active page. **CurrentRowNumber** is often used to determine a value for the *level1_row* and *level2_row* parameters in scroll path constructions. Because current row numbers are determined by the current context, **CurrentRowNumber** cannot determine a current row on a scroll level outside the current context (that is, a scroll level below the level where the PeopleCode program is currently executing).

For example, you could use a statement like this to retrieve the value of a buffer field on level 3 of the PET_VISITS page, in a PeopleCode program executing on level 2:

```
&VAL = FetchValue(RECORD.OWNER, CurrentRowNumber(1), RECORD.PET,
    CurrentRowNumber(2), RECORD.VISIT, &TARGETROW, VISIT_REASON);
```

Because the PeopleCode program is executing on level 2, **CurrentRowNumber** can return values for levels one and two, but not three, because level 3 is outside of the current context and has no current row number.

Looping Through Scroll Levels

Component Buffer functions are often used in **For** loops to loop through the rows on scroll levels below the level where the PeopleCode program is executing. The following loop, for example could be used in PeopleCode executing on a level 2 record field to loop through rows of data on level 3:

```
For &I = 1 To ActiveRowCount(RECORD.OWNER, CurrentRowNumber(1), RECORD.PET,
    CurrentRowNumber(2), RECORD.VISIT)
    &VAL = FetchValue(RECORD.OWNER, CurrentRowNumber(1), RECORD.PET,
        CurrentRowNumber(2), RECORD.VISIT, &I, VISIT_REASON);
    If &VAL = "Fleas" Then
        /* do something about fleas */
    End-If;
End-For;
```

A similar construct may be used in accessing other level 2 or level 1 scrolls, such as work scrolls.

In these constructions the **ActiveRowCount** function is often used to determine the upper bounds of the loop. When **ActiveRowCount** is used for this purpose, the loop goes through all of the active rows in the scroll (that is, rows that have not been specified as deleted). If you use **TotalRowCount** to determine the upper bounds of the loop, the loop goes through all of the rows in the scroll: first those that have not been specified as deleted, then those that have been specified as deleted.

See Also

Scroll Path Syntax in PeopleTools 7.5

Scroll Path Syntax prior to PeopleTools 7.5

Understanding Current Context

PeopleTools PeopleBook: PeopleCode Reference, "PeopleCode Built-in Functions,"
CurrentRowNumber

Scroll Path Syntax Before PeopleTools 7.5

In scroll path field references in PeopleTools 5 through 7, a *recordname.fieldname* expression performs two functions, identifying both the target scroll and the target field. For example, in this statement:

```
Hide (RECORD.VET, CurrentRowNumber (1), OWNER.OWNER_NAME, &ROWNUM);
```

the expression OWNER.OWNER_NAME serves both to identify the scroll level and to specify a record field. Compare the PeopleTools 7.5 syntax, where a **RECORD.recordname** expression identifies the scroll level by its primary record:

```
Hide (RECORD.VET, CurrentRowNumber (1), RECORD.OWNER, &ROWNUM,  
OWNER.OWNER_NAME);
```

The pre-7.5 syntax is supported in PeopleTools 7.5, and there is no need to update programs to the new syntax if they are already working correctly. However, the old syntax can sometimes result in ambiguous references if the target record is a Derived/Work or related display field that occurs on more than one level on the page. You can avoid this problem by updating to the new syntax. The best method to avoid ambiguity is to use the alternative **SCROLL.scrollname** construction.

See Also

Scroll Path Syntax with SCROLL.scrollname

CHAPTER 10

Data Buffer Access

This chapter provides an overview of data buffer classes and discusses how to:

- Understand data buffer classes examples.
- Understand current context.
- Access secondary component buffer data.
- Instantiate rowsets using non-component buffer data.

Understanding Data Buffer Access

In addition to the built-in functions you use to access the Component Buffer, there are classes of objects that provide access to structured data buffers using the new PeopleCode object syntax.

The data buffers accessed by these classes are typically the Component Buffers which are loaded when you open a component. However, these classes may also be used to access data from general data buffers, loaded by an Application Message, an Application Engine program, a Business Component, and so on.

The methods and properties of these classes provide functionality that is similar to what has been available previously using the built-in functions. However, they also provide improved consistency, flexibility, and new functionality.

Access Classes

There are four new data buffer classes: Rowset, Row, Record, and Field. These four classes are the foundation for accessing Component Buffer data through the new object syntax.

- A *field* object, which is instantiated from the Field class, is a single instance of data within a record and is based on a field definition.
- A *record* object, which is instantiated from the Record class, is a single instance of a data within a row and is based on a record definition. A record object consists of one to *n* fields.

- A *row* object, which is instantiated from the Row class, is a single row of data that consists of one to n records of data. A single row in a component scroll is a row. A row may have one to n child rowsets. For example, a row in a level 2 scroll may have n level 3 child rowsets.
- A *rowset* object is a data structure used to describe hierarchical data. It is made up of a collection of rows. A component scroll is a rowset. You can also have a level 0 rowset.

Data Buffer Model and Data Access Objects

The data model assumed by these classes is that of a PeopleTools component, where scrollbars or grids are used to describe a hierarchical, multiple-occurrence data structure. These four classes are built on the data model of a PeopleTools component, in which scrollbars or grids are used to describe a hierarchical, multiple-occurrence data structure. You can access these classes using dot notation.

The four data buffer classes relate to each other in a hierarchical manner. The main points to understand these relationships are:

- A *record* contains one or more *fields*.
 - Records contain the fields that make up that record.
- A *row* contains one or more *records* and zero or more child *rowsets*.
 - A row contains the records that make up that row. It may also contain child rowsets.
- A *rowset* contains one or more *rows*.
 - A rowset is a data structure that describes hierarchical data. For component buffers, think of a rowset as a scroll on a page that contains all of that scroll's data. A level 0 rowset contains all the data for the entire component.
 - You can use rowsets with application messages, file layouts, Business Interlinks, and other definitions besides components.
 - A level 0 rowset from a component buffer only contains one row, that is, the keys that the user specifies to initiate that component. A level 0 rowset from data that isn't a component, such as a message or a file layout, might contain more than one level 0 row.

The following is some basic PeopleCode that traverses through a two level Component Buffer using the new dot notation syntax. Level 0 is based on record QA_INVEST_HDR and level 1 is based on record QA_INVEST_LN.

```
Local Rowset &HDR_ROWSET, &LINE_ROWSET;
Local Record &HDR_REC, &LINE_REC;

&HDR_ROWSET = GetLevel0();

For &I = 1 to &HDR_ROWSET.RowCount
```

```

&HDR_REC = &HDR_ROWSET(&I).QA_INVEST_HDR;
&EMPLID = &HDR_REC.EMPLID.Value;
&LINE_ROWSET = &HDR_ROWSET(&I).GetRowset(1);
For &J = 1 to &LINE_ROWSET.RowCount
    &LINE_REC = &LINE_ROWSET(&J).QA_INVEST_LN;
    &LINE_SUM = &LINE_SUM + &LINE_REC.AMOUNT.Value;
End-For;
End-For;

```

Notice that each rowset is declared and instantiated. In general, your code is easier to read and maintain if you follow this practice.

Understanding Data Buffer Classes Examples

In this section, we provide examples of the following data buffer classes:

- Object creation.
- Data buffer hierarchy.
- Rowset.
- Hidden work scroll.

Most of the examples in this section use the page EMPLOYEE_CHECKLIST.

Employee Checklist

Schumacher,Simon
ID: 8001

*Checklist Date: 08/11/2000
Checklist: 000003
Repatriation Checklist

Responsible ID: 6602
Peppen,Jacques

Comment:

*Chklist Seq	*Chklist Itm	*Briefing Status	*Status Date
100	000015 Briefing with Human Resources	Initiated	08/11/2000
200	000025 Repatriation Discussion	Initiated	08/11/2000
300	000029 Career/Placement discussion	Initiated	08/11/2000

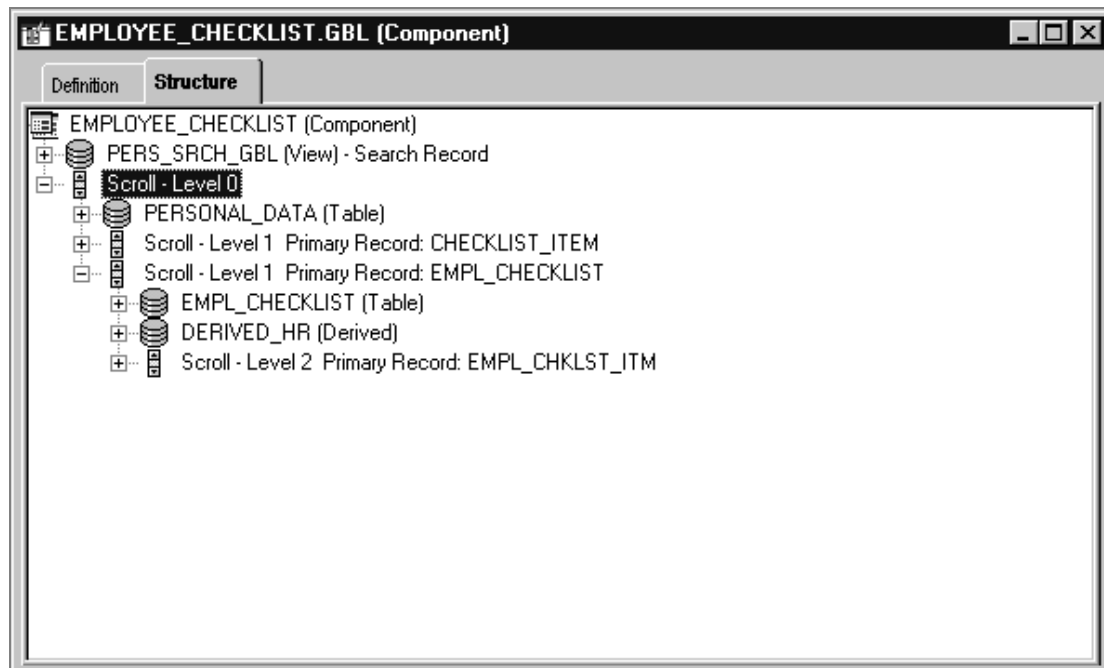
Save
Return to Search

Employee_Checklist page

This page has the following record structure:

Scroll Level	Associated Primary Record	Rowset and Variable Name
Level 0	PERSONAL_DATA	Level 0 rowset: &RS0
Level 1 scroll	EMPL_CHECKLIST	Level 1 rowset: &RS1
Level 1 hidden work scroll	CHECKLIST_ITEM	Level 1 rowset: &RS1H
Level 2 scroll	EMPL_CHKLST_ITM	Level 2 rowset: &RS2

Another way of looking at the structure of a component is to use the Structure View. All of the scrolls are labeled, and the primary record associated with each:



Component Structure view showing data hierarchy

In the following example the visible level 1 scroll also has only one row. That row is made up of the following records:

- EMPL_CHECKLIST
- DERIVED_HR
- CHECKLIST_TBL
- PERSONAL_DATA

You can see which records are associated with a scroll by looking at the Order view for a page:

EMPLOYEE_CHECKLIST.ENG (Page)							
Page Designer		Order					
	Lv	Label	Type	Field	Record	Display Control	Related Field
1	0	Frame	Frame			<input type="checkbox"/>	<input type="checkbox"/>
2	0	Frame	Frame			<input type="checkbox"/>	<input type="checkbox"/>
3	0	Frame	Frame			<input type="checkbox"/>	<input type="checkbox"/>
4	0	Employee Name	Edit Box	NAME	PERSONAL_DATA	<input type="checkbox"/>	<input type="checkbox"/>
5	0	ID	Edit Box	EMPLID	PERSONAL_DATA	<input type="checkbox"/>	<input type="checkbox"/>
6	1	Checklist Item Tbl	Scroll Bar			<input type="checkbox"/>	<input type="checkbox"/>
7	1	Checklist Sequen	Edit Box	CHECKLIST_SEQ	CHECKLIST_ITEM	<input type="checkbox"/>	<input type="checkbox"/>
8	1	Scroll Bar 1	Scroll Bar			<input type="checkbox"/>	<input type="checkbox"/>
9	1	Checklist Date	Edit Box	CHECKLIST_DT	EMPL_CHECKLIST	<input type="checkbox"/>	<input type="checkbox"/>
10	1	derived_hr.effdt	Edit Box	EFFDT	DERIVED_HR	<input type="checkbox"/>	<input type="checkbox"/>
11	1	Checklist	Edit Box	CHECKLIST_CD	EMPL_CHECKLIST	<input checked="" type="checkbox"/>	<input type="checkbox"/>
12	1	Checklist Descripti	Edit Box	DESCR	CHECKLIST_TBL	<input type="checkbox"/>	<input checked="" type="checkbox"/>
13	1	Responsible ID	Edit Box	RESPONSIBLE_I	EMPL_CHECKLIST	<input checked="" type="checkbox"/>	<input type="checkbox"/>
14	1	Responsible Nam	Edit Box	NAME	PERSONAL_DATA	<input type="checkbox"/>	<input checked="" type="checkbox"/>
15	1	Comment	Long Edit Box	COMMENTS	EMPL_CHECKLIST	<input type="checkbox"/>	<input type="checkbox"/>
16	2	Scroll Bar 2	Scroll Bar			<input type="checkbox"/>	<input type="checkbox"/>
17	2	Chklst Seq	Edit Box	CHECKLIST_SEQ	EMPL_CHKLST_ITM	<input type="checkbox"/>	<input type="checkbox"/>
18	2	Chklst Itm	Edit Box	CHKLST_ITEM_C	EMPL_CHKLST_ITM	<input checked="" type="checkbox"/>	<input type="checkbox"/>
19	2	Briefing Descriptio	Edit Box	DESCR	CHKLST_ITEM_TBL	<input type="checkbox"/>	<input checked="" type="checkbox"/>
20	2	Briefing Status	Drop Down List	BRIEFING_STAT	EMPL_CHKLST_ITM	<input type="checkbox"/>	<input type="checkbox"/>

EMPLOYEE_CHECKLIST page Order view showing records

The level 2 rowset has three rows. Each row is made up of two records: the primary record, EMPL_CHKLST_ITM, and CHKLST_ITM_TBL, the record associated with the related display field DESCR.

Employee Checklist

Schumacher,Simon

ID: 8001

*Checklist Date: 08/11/2000

Checklist: 000003 Repatriation Checklist

Responsible ID: 6602 Peppen,Jacques

Comment:

Level2 Rowset

*Chklst Seq

*Chklst Itm

*Briefing Status

*Status Date

100

000015

Briefing with Human Resources

Initiated

08/11/2000

200

000025

Repatriation Discussion

Initiated

08/11/2000

300

000029

Career/Placement discussion

Initiated

08/11/2000

Row2 for level2 rowset

Field

Save

Return to Search

Rowset and rows

Every record has fields associated with it, like NAME, EMPLID, CHECKLIST_SEQ, and so on. These are the fields associated with the record definitions, not the fields displayed on the page.

Object Creation Examples

Each data buffer access class has its own data type which is the same name as the class, that you should use when declaring your variables (Rowset objects should be declared as type Rowset, field objects as type Field, and so on.) Data buffer access class objects can be of type Local, Global, or Component.

The following declarations are assumed throughout the examples that follow:

```
Local Rowset &LEVEL0, &ROWSET;
Local Row &ROW;
Local Record &REC;
Local Field &FIELD;
```

Accessing Level 0

The following code instantiates a rowset object, from the Rowset class, that references the level 0 rowset, containing all the page data. It stores the object in the &LEVEL0 variable.

```
&LEVEL0 = GetLevel0();
```

The level 0 rowset contains all the rows, rowsets, records, and fields underneath it.

If the level 0 rowset is formed from Component Buffer data, the level 0 rowset has one row of data, and that row contains all the child rowsets, which in turn contain rows of data, that contain other child rowsets.

If the level 0 rowset is formed from buffer data, such as from an Application Message, the level 0 rowset may contain more than one row of data. Each row of the level 0 rowset contains all the child rowsets associated with that row, which in turn contain rows of data, that contain other child rowsets.

Use a level 0 rowset when you want an absolute path to a lower level object or when you want to do some processing on the entire data buffer. For example, suppose you load all new data into the Component Buffers, and want to redraw the page. You could use the following code:

```
/* Do processing to reload Component Buffers */
&LEVEL0 = GetLevel0();
&LEVEL0.Refresh();
```

See Also

PeopleTools PeopleBook: PeopleCode Reference, "PeopleCode Built-in Functions,"
GetRowset

Rowset Object

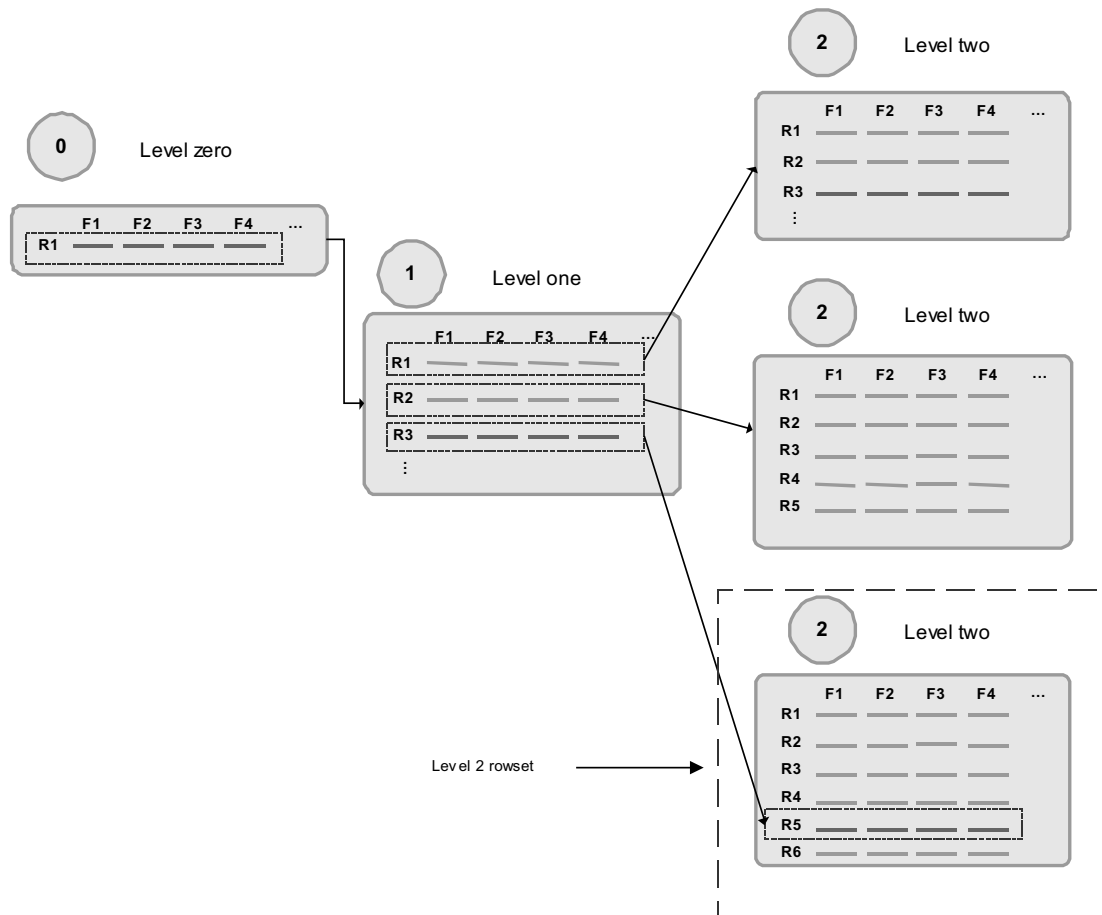
The following code instantiates a rowset object that references the rowset that contains the currently running PeopleCode program.

```
&ROWSET = GetRowset();
```

You might later use the `&ROWSET` variable and the `ActiveRowCount` property to iterate over all the rows of the rowset, or to access a specific row (using the `GetRow` method) or to hide a child rowset (by setting the `Visible` property.)

The level 1 rowset contains all the level 2 rowsets. However, the level 2 rowsets can only be accessed using the different rows of the level 1 rowset. From the level 0 or level 1 rowset, you can only access a level 2 rowset by using the level 1 rowset *and* the appropriate row.

For example, suppose your program is running on some field of row five of a level 2 scroll, which is on row three of its level 1 scroll. The resulting rowset contains all the rows of the level 2 scroll that are under the row three of the level 1 scroll. The rowset does *not* contain any data that is under any other level 2 scrolls.



Level 2 rowset from level 1 row

Let's illustrate this further with an example from the Employee Checklist page.

Suppose that one employee was associated with three different checklists: Foreign loan departure, Foreign loan arrival, and Foreign loan host. The checklist code field (`CHECKLIST_CD`) on the first level of the page drives the entries on the second level. Each row in the level 1 rowset produces a *different* level 2 rowset.

The Foreign Loan Departure checklist (000001) produces a checklist that contains such items as Briefing with Human Resources, Apply for Visas/Work permits, and so on.

Employee Checklist

Schumacher,Simon
ID: 8001

*Checklist Date: 08/11/2000
Checklist: 000001 Foreign Loan Departure Chcklst

Responsible ID: 6602 Peppen,Jacques

Comment:

<Previous + 1 of 3 - Next>

*Chklist Seq	*Chklist Itm	*Briefing Status	*Status Date
100	000015 Briefing with Human Resources	Initiated	08/11/2000
200	000030 Apply for Visas/Work Permits	Initiated	08/11/2000
300	000009 Reconfirm Relocation Package	Initiated	08/11/2000
400	000001 Select moving/storage company	Initiated	08/11/2000

Save
Return to Search

Foreign Loan Departure Checklist

The Foreign Loan Arrival Checklist (0000004) produces a checklist that contains items as Register at Consulate, Open new foreign bank accounts, and so on.

Employee Checklist

Schumacher, Simon ID: 8001

*Checklist Date: 08/11/2000 Checklist: 000004 Foreign Loan Arrival Checklist

Responsible ID: 7705 Holt, Susan

Comment:

Previous + 2 of 3 - Next

*Chklist Seq	*Chklist Itm	*Briefing Status	*Status Date
100	000022 Register at Consulate	Initiated	08/11/2000
200	000008 Open new foreign bank accounts	Initiated	08/11/2000
300	000018 Register children in school	Initiated	08/11/2000
400	000019 Join Newcomer's Club	Initiated	08/11/2000

Save Return to Search

Foreign Loan Arrival Checklist

The Foreign Loan Host Checklist (0000005) produces a checklist that contains items such as Arrange Meeting with Mentor, Clear Local tax filing reqmnts, and so on.

Employee Checklist

Schumacher, Simon ID: 8001

*Checklist Date: 08/11/2000 Checklist: 000005 Foreign Loan Host Checklist

Responsible ID: 7705 Holt, Susan

Comment:

<Previous + 3 of 3 - Next>

*Chklist Seq	*Chklist Item	*Briefing Status	*Status Date
100	000015 Briefing with Human Resources	Initiated	08/11/2000
200	000017 Arrange meeting with Mentor	Initiated	08/11/2000
300	000023 Clear local tax filing reqmnts	Initiated	08/11/2000
400	000027 Complete foreign tax forms	Initiated	08/11/2000

Save Return to Search

Foreign Loan Host Checklist

See Also

PeopleTools PeopleBook: PeopleCode Reference, "Rowset Class"

Row Object

When you create a page you put fields from different records onto the page. You can think of this as creating a type of pseudo-SQL join. The row returned from this pseudo join is a *row object*.

For example, the first level scroll of the EMPLOYEE_CHECKLIST page contains the following fields, associated with these records:

Field	Record
CHECKLIST_DT	EMPL_CHECKLIST
CHECKLIST_CD	EMPL_CHECKLIST
COMMENTS	EMPL_CHECKLIST
DESCR	CHECKLIST_TBL
NAME	PERSONAL_DATA
RESPONSIBLE_ID	EMPL_CHECKLIST

The pseudo-SQL join might look like the following:

```

JOIN A.CHECKLIST_DT, A.CHECKLIST_CD, A.COMMENTS, B.DESCR, C.NAME,
A.RESPONSIBLE_ID
FROM PS_EMPL_CHECKLIST A, PS_CHECKLIST_TBL B, PS_PERSONAL_DATA C, WHERE. . .

```

What goes into the WHERE clause is determined by the level 0 of the page. For our example, it's WHERE EMPLID=8001.

When the component is opened, data gets loaded into the Component Buffers. Any row returned by the pseudo-SQL statement is a level 1 *row object*.

CHECKLIST_DT	CHECKLIST_CD	COMMENTS	DESCR	NAME	RESPONSIBLE_ID
12/03/98	000001		Foreign Loan Department Checklist	Peppen, Jacque	6602

See Also

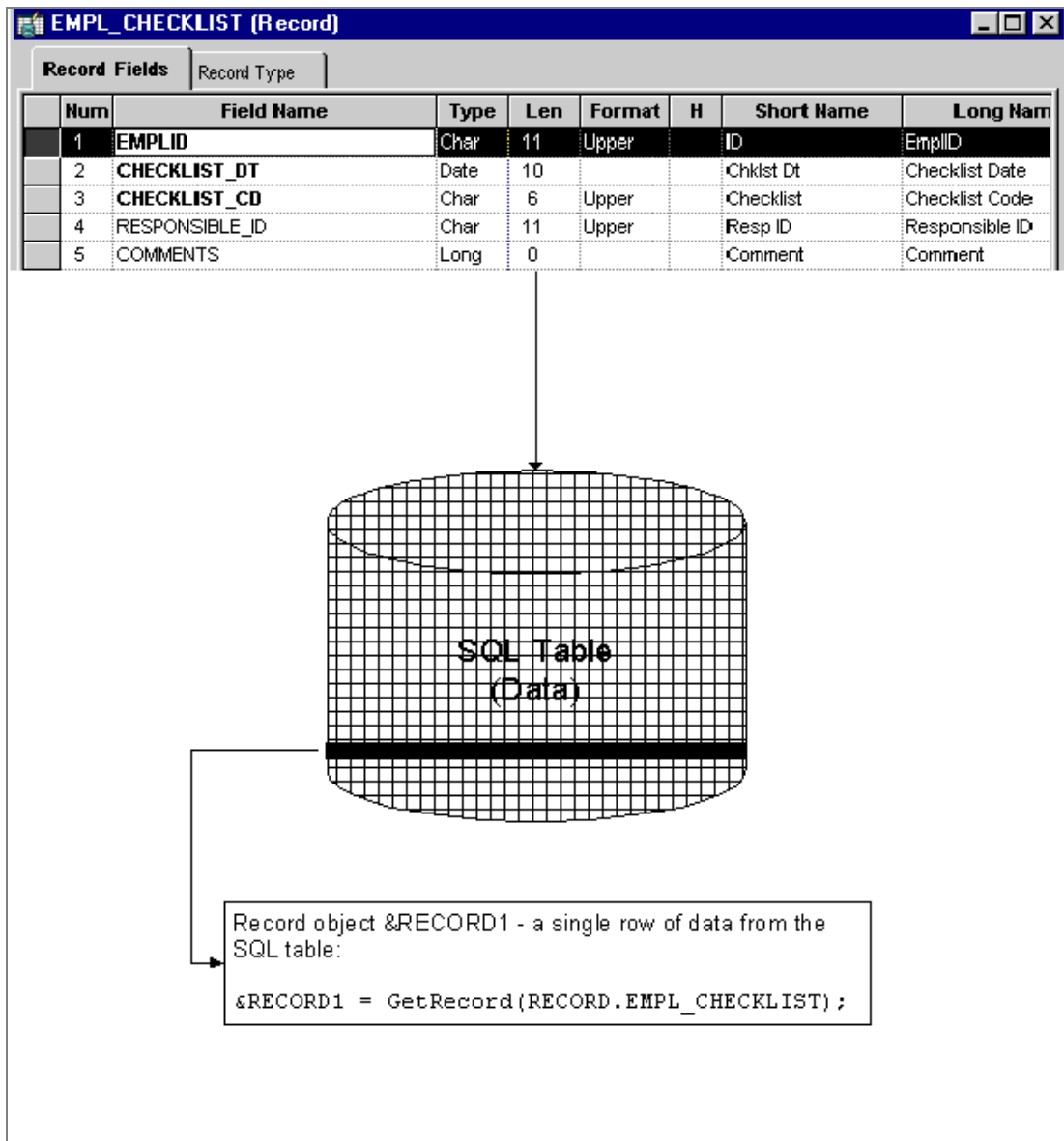
PeopleTools PeopleBook: PeopleCode Reference, "Row Class"

Record Object

A record *definition* is a definition of what your underlying SQL database tables look like, and how they process data. After you create your record definitions, you build the underlying SQL tables that contain the application data your users enter online in your production environment.

When you create a record object using the **CreateRecord** function, you're creating an area in the data buffers that has the same *structure* as the record definition, but no data.

When you instantiate a record object from the Record class using some variation of **GetRecord**, that record object references a single row of data in the SQL table.



Record object

Note. The data in the record that you retrieve is based on the *row*. This is analogous to setting keys to return a unique record.

The following code instantiates a record object for referencing the EMPL_CHECKLIST record of the specified row.

```
&REC = &ROW.GetRecord(RECORD.EMPL_CHECKLIST);
```

Using the "short" method, the following line of code is identical to the previous line:

```
&REC = &ROW.EMPL_CHECKLIST;
```

You might later use the &REC variable and the CopyFieldsTo property to copy all like-named fields from one record to another. In the following example, two row objects are created, the copy from row (COPYFRMROW) and the copy to row (COPYTROW). Using these rows, like-named fields are copied from CHECKLIST_ITEM to EMPL_CHKLIST_ITM.

```
For &I = 1 To &ROWSET1.ActiveRowCount
    &COPYFRMROW = &ROWSET1.GetRow(&I);
    &COPYTROW = &RS2.GetRow(&I);
    &COPYFRMROW.CHECKLIST_ITEM.CopyFieldsTo(&COPYTROW.EMPL_CHKLIST_ITM);
End-For;
```

A row may contain more than one record: in addition to the primary database record, you may have a related display record or a derived record. You can access these records as well. The level 1 rowset, &ROWSET1, is made up of many records. The following accesses two of them: EMPL_CHECKLIST and DERIVED_HR.

```
&REC1 = &ROW.EMPL_CHECKLIST;
&REC2 = &ROW.DERIVED_HR;
```

See Also

PeopleTools PeopleBook: PeopleCode Reference, "Record Class"

Field Object

The following instantiates a field object, from the Field class, that is used to access a specific field in the record.

```
&FIELD = &REC.GetField(FIELD.CHECKLIST_CD);
```

You might later use the &FIELD variable as a condition, like:

```
If ALL(&FIELD) Then
```

Or

```
If &FIELD.Value = "N" Then
```

Note. The data in the field that you retrieve is based on the *record*, which is in turn based on the *row*.

You can also set the value of a field. Remember, using **GetField** does not create a copy of the data from the Component Buffer. Setting the value or a property of the field object sets the actual Component Buffer field or property.

See Assigning Objects.

In the following example, the type of a field is checked, and the value is replaced with the tangent of that value if it is a number.

```
If &FIELD.Type <> "NUMBER" Then
    /* do error recording */
```

```

Else
    &FIELD.Value = Tan(&FIELD.Value);
End-If;

```

See Also

PeopleTools PeopleBook: PeopleCode Reference, "Field Class"

Traversing the Data Buffer Hierarchy Example

Suppose you want to access the BRIEFING_STATUS field at level 2 of the following page:

The screenshot shows the 'Employee Checklist' page. At the top, there's a header bar with 'Schumacher, Simon' and 'ID: 8001'. Below this is a form section with the following fields:

- *Checklist Date:** 08/11/2000
- Checklist:** 000003 (with a magnifying glass icon) Repatriation Checklist
- Responsible ID:** 6602 (with a magnifying glass icon) Peppen, Jacques
- Comment:** (empty text area with up/down arrows)

Below the form is a table with the following columns: *Chklist Seq, *Chklist Itm, *Briefing Status, and *Status Date.

*Chklist Seq	*Chklist Itm	*Briefing Status	*Status Date
100	000015 (with magnifying glass icon) Briefing with Human Resources	Initiated (dropdown arrow)	08/11/2000
200	000025 (with magnifying glass icon) Repatriation Discussion	Initiated (dropdown arrow)	08/11/2000
300	000029 (with magnifying glass icon) Career/Placement discussion	Initiated (dropdown arrow)	08/11/2000

At the bottom of the page are two buttons: 'Save' and 'Return to Search'.

Employee Checklist page

The first thing to ask is where is your code running? Where are you starting from? For this example, the code is starting at a field on a record at level 0. However, you do not always have to start at level 0.

After you start with level 0, you must traverse the data hierarchy, through the level 1 rowset, to the level 2 rowset, before you can access the record that contains the field. Here's the hierarchy again:

A rowset contains one or more rows, a row contains one or more records and zero or more child rowsets, and a record contains one or more fields.

Rowset

You first obtain the level 0 rowset, which is the `PERSONAL_DATA` rowset. However, you do not need to know the name of the level 0 rowset to access it.

```
&LEVEL0 = GetLevel0();
```

Rowsets Contain Rows

The next object to get is a row. As this code is working with data that is loaded from a page, there will only ever be one row at level 0. However, if you have rowsets that are populated with data that is not based on Component Buffers (for example, an Application Message) you may have more than one row at level 0.

```
&LEVEL0_ROW = &LEVEL0(1);
```

Rows Can Contain Child Rowsets

We need to get to the level 2 rowset. To do that, we need to traverse *through* the level 1 rowset first. Therefore, the next object we want to get is the level 1 rowset.

```
&LEVEL1 = &LEVEL0_ROW.GetRowset(SCROLL.EMPL_CHECKLIST);
```

Rowsets Contain Rows

If you're traversing a page, the first thing to always do after you get a rowset is to get the appropriate row. Because we want to process all the rows of the rowset, we set this up in a loop.

```
For &I = 1 to &LEVEL1.ActiveRowCount
    &LEVEL1_ROW = &LEVEL1(&I);
    . . .
End-For;
```

Rows Can Contain Child Rowsets, Rowsets Contain Rows

We need to traverse down another level in our page structure. This means accessing the second level rowset. Then we need to access the rows in the second level rowset, in another loop.

Because we're processing all the rows at the level 1, we're just adding code to the previous For loop. As we're processing through all the rows at level 2, we're adding a second For loop. The new code is in bold.

```
For &I = 1 to &LEVEL1.ActiveRowCount
    &LEVEL1_ROW = &LEVEL1(&I);
    &LEVEL2 = &LEVEL1_ROW.GetRowset(SCROLL.EMPL_CHKLST_ITM);
    For &J = 1 to &LEVEL2.ActiveRowCount
        &LEVEL2_ROW = &LEVEL2(&J);
        . . .
    End-For;
End-For;
```

Rows Contain Records

Rows also contain records. In fact, a row will always contain a record, and *may* contain only a child rowset, depending on how your page is set up. GetRecord is the default method for a row, so all you have to specify is the record name.

Because we're processing all the rows at the level 2, we're just adding code to the previous For loops. The new code is in bold.

```
For &I = 1 to &LEVEL1.ActiveRowCount
    &LEVEL1_ROW = &LEVEL1(&I);
    &LEVEL2 = &LEVEL1_ROW.GetRowset(ROLL.EMPL_CHKLIST_ITM);
    For &J = 1 to &LEVEL2.ActiveRowCount
        &LEVEL2_ROW = &LEVEL2(&J);
        &RECORD = &LEVEL2_ROW.EMPL_CHKLIST_ITM;
        . . .
    End-For;
End-For;
```

Records Contain Fields

Records are made up of fields. GetField is the default method for a record, so all you have to specify is the field name.

Because we're processing all the rows at the level 1, we're just adding code to the previous For loops. The new code is in bold.

```
For &I = 1 to &LEVEL1.ActiveRowCount
    &LEVEL1_ROW = &LEVEL1(&I);
    &LEVEL2 = &LEVEL1_ROW.GetRowset(ROLL.EMPL_CHKLIST_ITM);
    For &J = 1 to &LEVEL2.ActiveRowCount
        &LEVEL2_ROW = &LEVEL2(&J);
        &RECORD = &LEVEL2_ROW.EMPL_CHKLIST_ITM;
        &FIELD = &RECORD.BRIEFING_STATUS;
        /* Do processing */
    End-For;
End-For;
```

Using Shortcuts

The previous code is the long way of accessing this field. What if you wanted to use all the shortcuts, and access the field in one line of code? Here it is! The following code assumes all rows are 1.

Rowset
Row
Rowset
Row
Rowset
Row
Record
Field

```
&FIELD = GetLevel0() (1).EMPL_CHECKLIST(1).EMPL_CHKLIST_ITM(1).EMPL_CHKLIST_ITM.BRIEFING_STATUS;
```

Rowset example

Here's another method of expressing the code:

Object Type	Code
Rowset	&LEVEL0 = GetLevel0();
Row	&LEVEL0_ROW = &LEVEL0(1);
Rowset	&LEVEL1 = &LEVEL0_ROW.GetRowset(SCROLL.EMPL_CHECKLIST);
	For &I = 1 to &LEVEL1.ActiveRowCount
Row	&LEVEL1_ROW = &LEVEL1(&I);
Rowset	&LEVEL2 = &LEVEL1_ROW.GetRowset(SCROLL.EMPL_CHKLIST_ITM);
	For &J = 1 to &LEVEL2.ActiveRowCount
Row	&LEVEL2_ROW = &LEVEL2(&J);
Record	&RECORD = &LEVEL2_ROW.EMPL_CHKLIST_ITM;
Field	&FIELD = &RECORD.BRIEFING_STATUS;
	/* Do processing */
	End-For;
	End-For;

Traversing a Rowset Example

The following code example traverses up to four levels of rowsets and could easily be modified to do more. This example only processes the *first* record in every rowset. If you wanted to process every record, you must set up another For loop (For &R = 1 to &LEVELX.RECORDCOUNT, and so on.) Notice the use of ChildCount (to process all children rowsets within a rowset), ActiveRowCount, IsChanged, and dot notation.

‘...’ is where application specific code goes.

```

&Level0_ROWSET = GetLevel0();
For &A0 = 1 To &Level0_ROWSET.ActiveRowCount

    ...

    /*****

```



```

/* Process Level 1 Records */
/*-----*/
    If &Level0_ROWSET(&A0).ChildCount > 0 Then
    For &B1 = 1 To &Level0_ROWSET(&A0).ChildCount
        &LEVEL1_ROWSET = &Level0_ROWSET(&A0).GetRowset (&B1);
        For &A1 = 1 To &LEVEL1_ROWSET.ActiveRowCount
            If &LEVEL1_ROWSET(&A1).GetRecord(1).IsChanged Then

                ...

        /***** */
/* Process Level 2 Records */
/*-----*/
        If &LEVEL1_ROWSET(&A1).ChildCount > 0 Then
        For &B2 = 1 To &LEVEL1_ROWSET(&A1).ChildCount
            &LEVEL2_ROWSET = &LEVEL1_ROWSET(&A1).GetRowset (&B2);
            For &A2 = 1 To &LEVEL2_ROWSET.ActiveRowCount
                If &LEVEL2_ROWSET(&A2).GetRecord(1).IsChanged Then

                    ...

        /***** */
/* Process Level 3 Records */
/*-----*/
            If &LEVEL2_ROWSET(&A2).ChildCount > 0 Then
            For &B3 = 1 To &LEVEL1_ROWSET(&A2).ChildCount
                &LEVEL3_ROWSET =
&LEVEL2_ROWSET(&A2).GetRowset (&B3);
                For &A3 = 1 To
&LEVEL3_ROWSET.ActiveRowCount
                    If
&LEVEL3_ROWSET(&A3).GetRecord(1).IsChanged Then

                        ...

                        End-If; /* A3 - IsChanged */
                        End-For; /* A3 - Loop */
                    End-For; /* B3 - Loop */
                End-If; /* A2 - ChildCount > 0 */
            /***** */
/* End of Process Level 3 Records */
/***** */

            End-If; /* A2 - IsChanged */
            End-For; /* A2 - Loop */
            End-For; /* B2 - Loop */

```

```

                                End-If; /* A1 - ChildCount > 0 */
/*-----*/
/* End of Process Level 2 Records */
/*****/

                                End-If; /* A1 - IsChanged */
                                End-For; /* A1 - Loop */
                                End-For; /* B1 - Loop */
                                End-If; /* A0 - ChildCount > 0 */

/*-----*/
/* End of Process Level 1 Records */
/*****/

End-For; /* A0 - Loop */

```

Using a Hidden Work Scroll Example

In the FieldChange event for the CHECKLIST_CD field on the EMPL_CHECKLIST record, there is a program which does the following:

1. Flushes the rowset/hidden work scroll.
2. Selects into the hidden work scroll based on the value of the CHECKLIST_CD field and the effective date.
3. Clears out the second level scroll.
4. Copies like-fields from the hidden work scroll to the second level scroll.

The following is the code to do this using built-in functions.

```

&CURRENT_ROW_L1 = CurrentRowNumber(1);

&ACTIVE_ROW_L2 = ActiveRowCount(RECORD.EMPL_CHECKLIST, &CURRENT_ROW_L1,
RECORD.EMPL_CHKLIST_ITM);

If All(CHECKLIST_CD) Then
    ScrollFlush(RECORD.CHECKLIST_ITEM);
    ScrollSelect(1, RECORD.CHECKLIST_ITEM, RECORD.CHECKLIST_ITEM, "Where
Checklist_Cd = :1 and EffDt = (Select Max(EffDt) From PS_Checklist_Item Where
Checklist_Cd = :2)", CHECKLIST_CD, CHECKLIST_CD);

    &FOUNDDOC = FetchValue(CHECKLIST_ITEM.CHLST_ITEM_CD, 1);
    &SELECT_ROW = ActiveRowCount(RECORD.CHECKLIST_ITEM);

    For &I = 1 To &ACTIVE_ROW_L2

```

```

        DeleteRow(RECORD.EMPL_CHECKLIST, &CURRENT_ROW_L1,
RECORD.EMPL_CHKLIST_ITM, 1);
    End-For;

    If All(&FOUNDDOC) Then
        For &I = 1 To &SELECT_ROW
            CopyFields(1, RECORD.CHECKLIST_ITEM, &I, 2, RECORD.EMPL_CHECKLIST,
&CURRENT_ROW_L1, RECORD.EMPL_CHKLIST_ITM, &I);
            If &I <> &SELECT_ROW Then
                InsertRow(RECORD.EMPL_CHECKLIST, &CURRENT_ROW_L1,
RECORD.EMPL_CHKLIST_ITM, &I);
            End-If;
        End-For;
    End-If;
End-If;

```

This following program performs the same function as the previous code, only it uses the data buffer classes:

1. Flushes the rowset/hidden work scroll (&RS1H).
5. Selects into &RS1H based on the value of the CHECKLIST_CD field and the effective date.
6. Clears out the second level Rowset (&RS2).
7. Copies like fields from &RS1H to &RS1.

```

Local Rowset &RS0, &RS1, &RS2, &RS1H;

&RS0 = GetLevel0();
&RS1 = GetRowset();
&RS2 = GetRowset(SCROLL.EMPL_CHKLIST_ITM);
&RS1H = &RS0.GetRow(1).GetRowset(SCROLL.CHECKLIST_ITEM);

&MYFIELD = CHECKLIST_CD;

If All(&MYFIELD) Then
    &RS1H.Flush();
    &RS1H.Select(RECORD.CHECKLIST_ITEM, "where Checklist_CD = :1 and EffDt =
(Select Max(EffDt) from PS_CHECKLIST_ITEM Where CheckList_CD = :2)",
CHECKLIST_CD, CHECKLIST_CD);

    For &I = 1 To &RS2.ActiveRowCount
        &RS2.DeleteRow(1);
    End-For;

    &FOUND = &RS1H.GetCurrEffRow().CHECKLIST_ITEM. CHKLST_ITEM_CD.Value;

    If All(&FOUND) Then

```

```

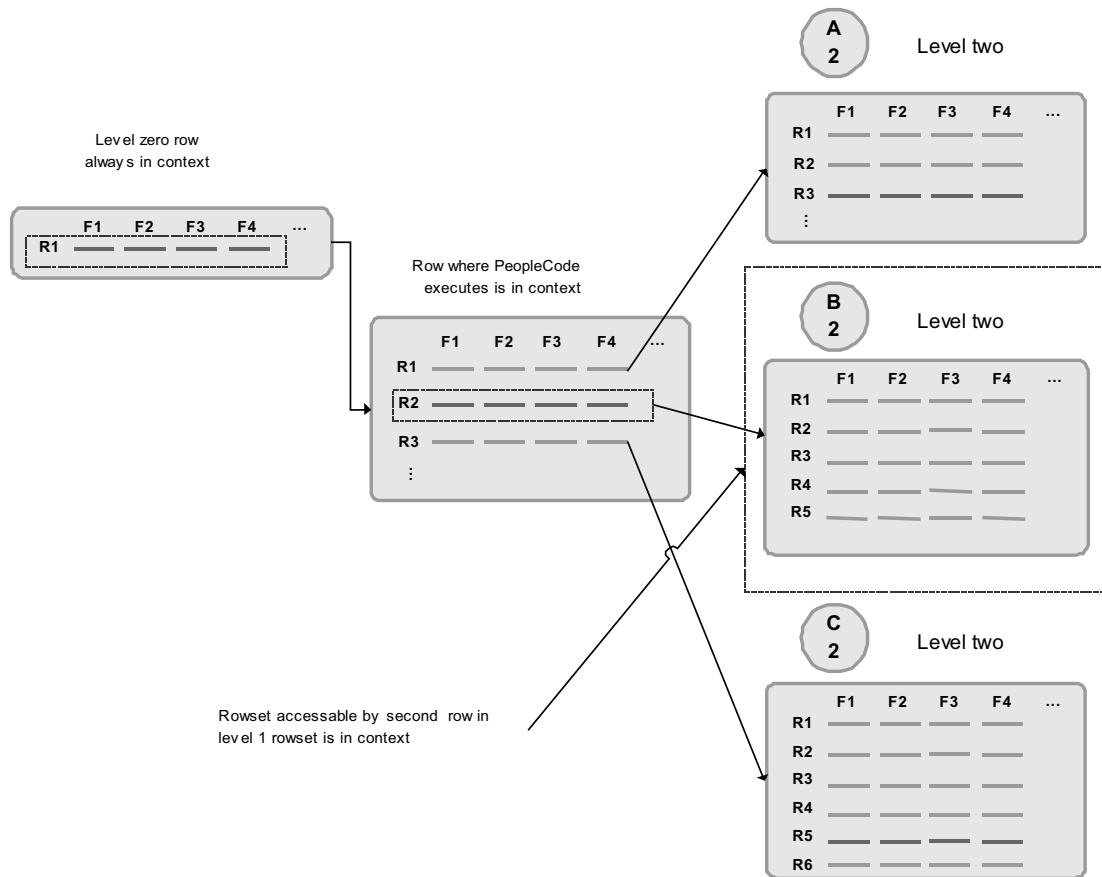
For &I = 1 To &RS1H.ActiveRowCount
    &COPYFRMROW = &RS1H.getrow(&I);
    &COPYTROW = &RS2.getrow(&I);
    &COPYFRMROW.CHECKLIST_ITEM.CopyFieldsTo(&COPYTROW.EMPL_CHKLIST_ITM);
    If &I <> &RS1H.ActiveRowCount Then
        &RS2.InsertRow(&I);
    End-If;
End-For;
End-If;
End-If;

```

Understanding Current Context

Most PeopleCode programs run in a *current context*. The current context determines which buffer fields can be contextually referenced from PeopleCode, and which row of data is the "current row" on each scroll level at the time a PeopleCode program is running.

The current context for the data buffer access classes is similar to the current context for accessing the Component Buffer. However, with the Rowset class, additional explanation is necessary.



Current context for rowsets

In this example a PeopleCode program is running in a buffer field on the second row of the level 1 rowset. The following code returns a row object for the second row of the level 1 rowset, because that is the row that is the *current context*.

```
Local Row &ROW

&ROW = GetRow();
```

The following code returns the B2 level 2 rowset, because of the current context:

```
Local Rowset &ROWSET2

&ROWSET2 = &ROW.GetRowset (SCROLL.EMPL_CHKLIST_ITM);
```

This code will *not* return either the C2 or the A2 rowsets. It returns only the rowset associated with the second row of the level 1 rowset.

See Also

"Referencing Data in the Component Buffer," Contextual References

Creating Records or Rowsets and Current Context

When you instantiate a record object using the CreateRecord function, you are only creating an area in the data buffers that has the same *structure* as the record definition. It does *not* contain any data. This record object will not have a parent rowset or be associated with a row. It is a free-standing record object, and therefore is not considered part of the current context.

The same concept applies when you instantiate a rowset object using the CreateRowset function. You are only creating an area in the data buffers that has the same *structure* as the records or rowset the new rowset is based on. It will *not* contain any data. This type of rowset will not have a parent rowset or row.

See Also

PeopleTools PeopleBook: PeopleCode Reference, "'PeopleCode Built-in Functions,' CreateRecord, CreateRowset

Accessing Secondary Component Buffer Data

When a secondary page is run, the data for its buffers is *copied* from the parent component to a buffer structure for the secondary page. This means there are two copies of this data. The data buffer classes give access to *both* of these copies of the data. Direct field references (recname.fieldname) will always use the current context to determine which value to access. So, in general, when using a secondary page, make sure that all your references are based on the secondary page.

Instantiating Rowsets Using Non-Component Buffer Data

Both the application message and the file layout technologies represent hierarchical data, and use the rowset, row, record, field hierarchy. Though you use different methods to instantiate a rowset object for this data, you still use the same rowset, row, record, and field methods and properties to manipulate the data. (Any exceptions are marked in the documentation.)

To instantiate a rowset for a message:

```
&MSG = CreateMessage (MESSAGE.EMPLOYEE_DATA) ;
&MYROWSET = &MSG.GetRowset () ;
```

To instantiate a rowset for a file layout:

```
&MYFILE = GetFile (&SOMENAME, "R") ;
&MYFILE.SetFileLayout (FILELAYOUT.SOMELAYOUT) ;
&MYROWSET = &MYFILE.ReadRowset () ;
```

In an Application Engine program, the default state record is considered the primary record, and the main record in context. You can access the default state record using the following:

```
&STATERECORD = GetRecord () ;
```

If you have more than one state record associated with an Application Engine program, you can access them the same way you would access other, nonprimary data records, by specifying the record name. For example:

```
&ALTSTATE = GetRecord (RECORD.AE_STATE_ALT) ;
```

See Also

"Using Methods and Built-In Functions," Using Standalone Rowsets

PeopleTools PeopleBook: PeopleSoft Integration Broker, "Defining Messages"

PeopleTools PeopleBook: Peoplesoft Application Engine, "Using Meta-SQL and PeopleCode"

CHAPTER 11

PeopleCode and the Component Processor

This chapter provides an overview of the Component Processor and its complex interaction with PeopleCode programs and discusses the following topics:

- Events outside the Component Processor flow.
- How PeopleCode programs are triggered.
- Component Processor behavior.
- Processing sequences.
- PeopleSoft Internet Architecture processing considerations.
- Deferred processing mode.
- PeopleCode events.
- PeopleCode execution in multiple scroll pages.

Understanding the Component Processor

The Component Processor is the PeopleTools runtime engine that controls processing of the application from the time the end-user requests a component from an application menu through the time that the database is updated and processing of the component is complete.

This chapter describes the PeopleCode events that are generated during the Component Processor's flow of execution at runtime, and how PeopleCode events trigger PeopleCode programs.

Events Outside the Component Processor Flow

Application Messages also have events associated with them (OnRouteSend, OnRouteReceive, OnRequest and Subscription.) However, these events are only associated with the message definition, and are not associated with any page. Therefore, they aren't considered part of the Component Processor flow.

A PeopleSoft Application Engine program can have a PeopleCode program as an action. Though the right-hand drop-down list box on the PeopleCode Editor window displays the text "OnExecute" this really isn't an event. Any PeopleCode contained in an Application Engine action will only be executed when the action is executed.

A Component Interface can have user-defined methods associated with it. These methods aren't part of any processor flow: they're called as needed by the program executing the Component Interface. In addition, the mobile events (OnConflict, OnSelect, OnValidate) are not part of the Component Processor flow, but part of the flow of mobile synchronization.

Security has a signon "event" during signon. This is actually PeopleCode programs on a record field that you've specified in setting up security.

Though Application packages have a right-hand drop-down list box on the PeopleCode Editor window that displays the text "OnExecute" this really isn't an event. Any PeopleCode contained in the Application Class is only executed when called explicitly in a PeopleCode program.

See Also

PeopleTools PeopleBook: PeopleCode Reference, "Component Interface Classes"

PeopleTools PeopleBook: PeopleCode Reference, "Application Classes"

PeopleTools PeopleBook: PeopleSoft Integration Broker, "Defining Messages"

PeopleTools PeopleBook: Peoplesoft Application Engine, "Creating Application Engine Programs," PeopleCode Actions

Security

PeopleTools PeopleBook: Peoplesoft Mobile Agent, "Introducing PeopleSoft Mobile Agent"

How PeopleCode Programs Are Triggered

PeopleCode can be associated with a PeopleCode record field, a component record, and many other items. PeopleCode events fire at particular times, in particular sequences, during the course of the Component Processor's flow of execution. When an event fires, it triggers PeopleCode programs on specific objects.

The following items have events that are part of the Component Processor flow:

<i>Items</i>	<i>Events trigger</i>
Menu Items	programs associated with the menu item
Component record fields	programs on specific rows of data
Component records	programs on specific rows of data
Components	programs associated with the component
Pages	programs associated with the page

<i>Items</i>	<i>Events trigger</i>
Record fields	programs on specific rows of data

Let's examine two examples.

Suppose the end-user changes the data in a page field, then presses TAB to move out of the field. This end-user action causes the FieldEdit PeopleCode event to fire. The FieldEdit event affects only the specific field and row where the change took place. If a FieldEdit PeopleCode program is associated with that record field, the program is executed. The program is executed just once, on the specific field and row of data.

If you have two FieldEdit PeopleCode programs, one associated with the record field and a second associated with the component record field, both programs are executed, but only on the specific field and row of data. The FieldEdit PeopleCode program associated with the first record field fires first, then the FieldEdit PeopleCode program associated with the first component record field fires.

By contrast, suppose the end-user has opened a component for updating. As part of building the component the Component Processor fires the RowInit event. This event triggers RowInit PeopleCode programs on every record field on every row of data in the component. In a scroll area with multiple rows of data, every RowInit PeopleCode program is executed once for each row.

In addition, if you have RowInit PeopleCode associated with both the record field and the component record, both programs are executed against every record field on every row of data in the component. The RowInit PeopleCode program associated with the first record field fires first, then the RowInit PeopleCode program associated with the first component record fires. This means if you've set the value of a field with the record field RowInit PeopleCode, then reset the field with the component record RowInit PeopleCode, the second value is the one that appears to the end-user.

When you develop with PeopleCode you must consider when and where your programs will be triggered during the Component Processor's flow of execution.

In this section, we discuss how to:

- Access PeopleCode programs.
- Understand the execution order of events and PeopleCode.

See Also

Execution Order of Events and PeopleCode

Accessing PeopleCode Programs

Every PeopleCode program is associated with a PeopleCode event, and is often referred to by that name, such as RowInit PeopleCode or FieldChange PeopleCode. These programs are accessible from, and associated with, different items. The following table lists which event, and hence which types of PeopleCode programs, are accessible from which item.

Note. The SearchInit and SearchSave events (under Component Record) are available only for the search record associated with a component.

<i>Record Field</i>	<i>Component Record Field</i>	<i>Component Record</i>	<i>Component</i>	<i>Page</i>	<i>Menu</i>
FieldChange	FieldChange	RowDelete	PostBuild	Activate	ItemSelected
FieldDefault	FieldDefault	RowInit	PreBuild		
FieldEdit	FieldEdit	RowInsert	SavePostChg		
FieldFormula	PrePopup	RowSelect	SavePreChg		
PrePopup		SaveEdit	Workflow		
RowDelete		SavePostChg			
RowInit		SavePreChg			
RowInsert		SearchInit			
RowSelect		SearchSave			
SaveEdit					
SavePostChg					
SavePreChg					
SearchInit					
SearchSave					
Workflow					

The following table lists the different types of PeopleCode programs and where they're accessible from in PeopleSoft Application Designer.

<i>PeopleCode Programs</i>	<i>In PeopleSoft Application Designer</i>
Record field	Record definitions and page definitions
Component record field, component record, and component	Component definitions
Menu item	Menu definitions

Understanding Execution Order of Events and PeopleCode

When you develop with PeopleCode you must consider when and where your programs are triggered during the Component Processor's flow of execution.

In PeopleSoft, the component is the representation of a transaction. Therefore, any PeopleCode that is associated with a transaction should be in events associated with some level of the component. If you have code that should be executed every time a field is edited, you should put it at the record field level. If you associate code with the correct transaction, you don't have to check for the component that's issuing it (such as, surrounding your code

with dozens of `If %Component =`). Records are more reusable and code is more maintainable.

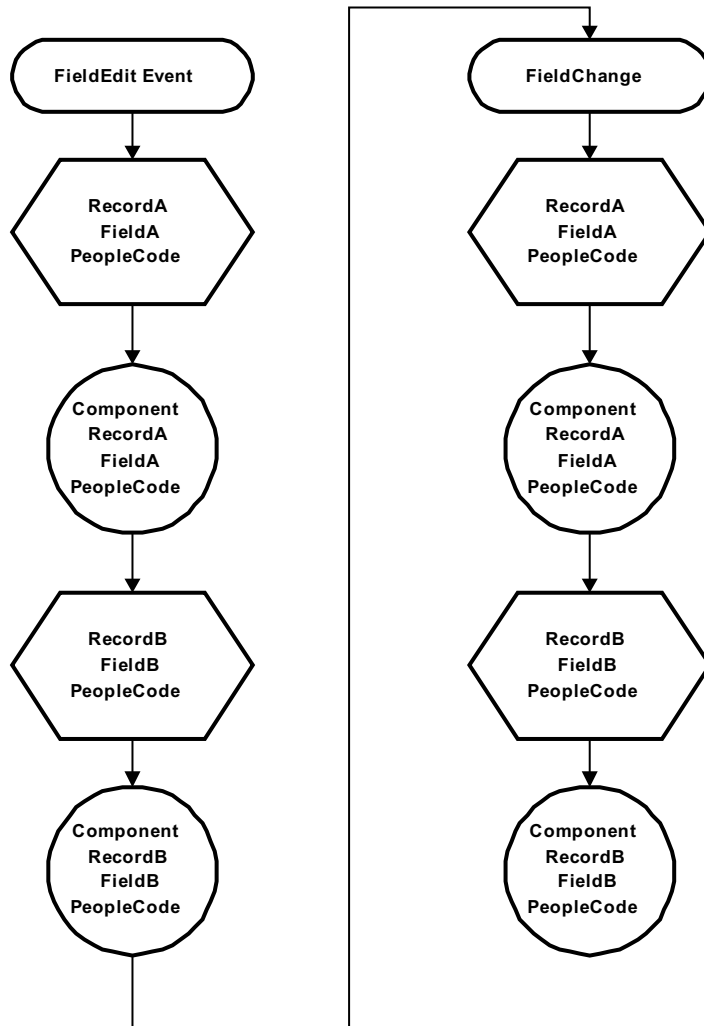
For example, if you have start and end dates for a course, you would always want to check and make sure the end date was after the start date. Therefore, your program to check the dates would go on the `SaveEdit` at the record field level.

All similarly named component events fire after the like-named record event. That is, the PeopleCode program associated with the record field event fires first, then the PeopleCode program associated with the like-named component event fires. This means that if you've set the value of a field with the record field PeopleCode, then reset the field with like-named component PeopleCode, the second value is the one that will be displayed to the end-user.

This is best illustrated with some examples:

Events After User Changes Field

```
Record.recordA.fieldA.FieldEdit -> Component.recordA.fieldA.FieldEdit ->
Record.recordB.fieldB.FieldEdit -> Component.recordB.fieldB.FieldEdit ->
Record.recordA.fieldA.FieldChange -> Component.recordA.fieldA.FieldChange ->
Record.recordB.fieldB.FieldChange -> Component.recordB.fieldB.FieldChange ->
```



Flow of events and PeopleCode programs after end-user changes a field

Events After User Saves

```
Record.recordA.fieldA.SaveEdit -> Record.recordA.fieldB.SaveEdit ->
Record.recordA.fieldC.SaveEdit -> Component.recordA.SaveEdit
```

```
Record.recordB.fieldA.SaveEdit -> Record.recordB.fieldB.SaveEdit ->
Record.recordB.fieldC.SaveEdit -> Component.recordB.SaveEdit
```

```
Record.recordA.fieldA.SavePreChange -> Record.recordA.fieldB.SavePreChange ->
Record.recordA.fieldC.SavePreChange -> Component.recordA.SavePreChange
```

```
Record.recordB.fieldA.SavePreChange -> Record.recordB.fieldB.SavePreChange ->
Record.recordB.fieldC.SavePreChange -> Component.recordB.SavePreChange
```

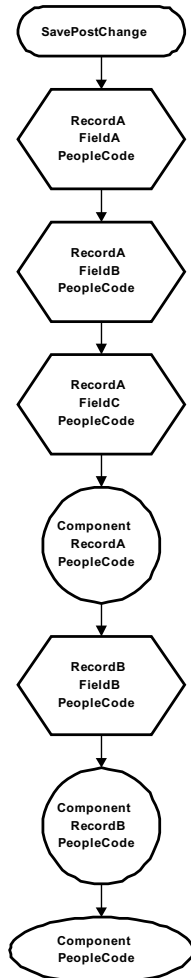
```
Record.recordA.fieldA.WorkFlow -> Record.recordB.fieldB.WorkFlow ->
Record.recordC.fieldC.WorkFlow
```

```
Component.Workflow
```

```
Record.recordA.fieldA.SavePostChange -> Record.recordA.fieldB.SavePostChange -
> Record.recordA.fieldC.SavePostChange -> Component.recordA.SavePostChange
```

```
Record.recordB.fieldA.SavePostChange -> Component.recordB.SavePostChange
```

```
Component.SavePostChange
```



Flow of PeopleCode programs after SavePostChange

Component Processor Behavior

This section takes a high-level look at the behaviors of the Component Processor from page startup to page display, and the processes initiated by end-user action after the page is displayed.

Note. Keep in mind that this description is for components *not* running in *deferred* mode. If your component or page is running in deferred mode, refer to Deferred Processing Mode.

The next section, Processing Sequences, examines these processes in greater detail, showing the flow of system actions and PeopleCode events.

In this section, we discuss the following topics:

- From page start to page display.
- End-user actions in the component.

From Page Start to Page Display

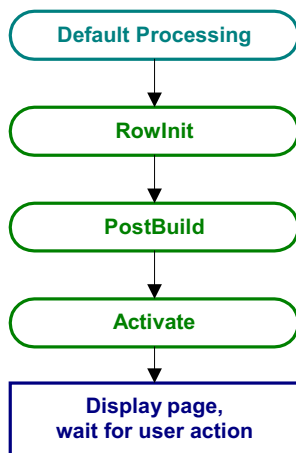
Before the end-user selects a component, the system is in reset state, in which no component is displayed. Component Processor's flow of execution begins when the end-user selects a component from a PeopleSoft menu. The Component Processor then:

1. Performs search processing, in which it obtains and saves search key values for the component.
2. Retrieves from the database server any data needed to build the component, then builds the component, creating buffers for the component data.
3. Does any additional processing for the component, the page, or any ActiveX controls.

Note. ActiveX control events are applicable only in the Windows client, not in the PeopleSoft Internet Architecture.

4. Displays the component and waits for end-user action.

The following flowchart shows the flow of execution at a high level.





Processing up to Page Display

End-User Actions in the Component

After the component is built and displayed, the Component Processor can respond to a number of possible end-user actions. The following table lists the end-user actions and briefly describes the resulting processing.





See Processing Sequences for details about the processing sequences that result from the end-user action, or follow the links in the table.

<i>End-User Action</i>	<i>Description</i>
Row Insert Processing	When the end-user requests a row insert, the Component Processor adds a row of data in the active scroll area, then displays the page again and waits for another action.
Row Delete Processing	When the end-user requests a row delete, the Component Processor flags the current row as deleted, then displays the page again and waits for another action.
Field Modification	If the end-user edits a page field, then leaves the field, the Component Processor performs standard edits (such as checking the data type and checking for values out of range). If the contents of the field do not pass the standard system edits, the Component Processor redisplay the page with an error or warning message and changes the field's color to the system color for field edit errors—usually red. Until the end-user corrects the error, the Component Processor will not let the end-user save changes or navigate to another field. If the contents of the field pass the standard system edits, the system redisplay the page and waits for further action.
Prompts	If the end-user clicks the prompt icon next to a field () a list of valid values for the prompt field displays. If the end-user clicks Return To Search or presses ALT+2, then presses ENTER, a search dialog appears, enabling them to enter an alternate search key or partial value. If the end-user clicks the detail button next to a date field (XXX) a calendar displays.
Pop-up Menu Display	If the end-user clicks the pop-up icon next to a field () a pop-up menu appears. This can be a default pop-up menu or one that has been defined by the developer. If the end-user clicks the pop-up icon at the bottom of the page, the pop-up menu for the page displays.
ItemSelected Processing	The end-user can select an item from a pop-up menu to execute a command.
PushButtons	The end-user can click a command pushbutton to execute a command.

End-User Action	Description
Save Processing	<p>The end-user can direct the system to save a component by clicking Save or by pressing ALT+1, then pressing ENTER. If any component data has been modified, PeopleSoft also prompts the end-user to save a component when the Next or List button is clicked, or when a new action or component is selected.</p> <p>The Component Processor first validates the data in the component then updates the database with the changed component data. After the update a SQL Commit command finalizes the changes.</p> <p>If a new component or new key has been requested, the Component Processor goes through Reset State to display a new component. If the end-user has not requested a new component, the Component Processor displays the page and waits for another end-user action.</p>
Exit Component	If the end-user clicks EXIT, the component goes into Reset State until the end-user requests another component.

Processing Sequences

This section examines possible sequences of actions and PeopleCode events that can occur within the Component Processor's flow of execution. The logic of each sequence of actions is presented in a flow diagram with the following elements:

Symbol	Description
	Blue rectangles represent actions taken by the system.
	Dark red rhomboids represent branches (decision points) in the logic.
	Green ellipses represent PeopleCode Events.
	Teal ellipses are subprocesses.

Most of the sequences described in the following sections correspond to the high-level behaviors defined in Component Processor Behavior. However, two of the sections describe subsequences that occur only in the context of a larger sequence. These are Default Processing which occurs in a number of different contexts, and Row Select Processing which most commonly occurs as a part of component build in any of the Update action modes; RowSelect Processing also occurs when a ScrollSelect or related function is executed to load data into a scroll.

Note. Keep in mind that the sequences described here are generalized, and that under some circumstances variations may occur, particularly when a PeopleCode function within a processing sequence initiates another processing sequence. For example, if a row of data is inserted or deleted programmatically during the Component Build sequence, this sets off a Row Insert or Row Delete sequence. **Note.** Also keep in mind that this description is for components *not* running in *deferred* mode. If your component or page is running in deferred mode, refer to Deferred Processing Mode.

In this section, we discuss the following topics:

- Default processing.
- Search processing in update mode.
- Search processing in add mode.
- Component build processing in update mode.
- Row select processing.
- Component build processing in add mode.
- Field modification.
- Row insert processing.
- Row delete processing.
- Pushbuttons.
- Prompts.
- Pop-up menu display.
- ItemSelected processing.
- Save processing.
- Exit component.

Default Processing

In default processing, any blank fields in the component are set to their default value (if one is specified). You can specify the default value either in the Record Field Properties, or in FieldDefault PeopleCode. If no default value is specified, the field is left blank.

Note. In the PeopleSoft Internet Architecture, if an end-user changes a field, but there is nothing to cause a trip to the server on that field, default processing and FieldFormula PeopleCode do not run. They only run when some other event causes a trip to the server.

Default processing is relatively complex. For the sake of clarity, the following two sections describe (1) how default processing works on the level of the individual field, and (2) how default processing works in the broader context of the component.

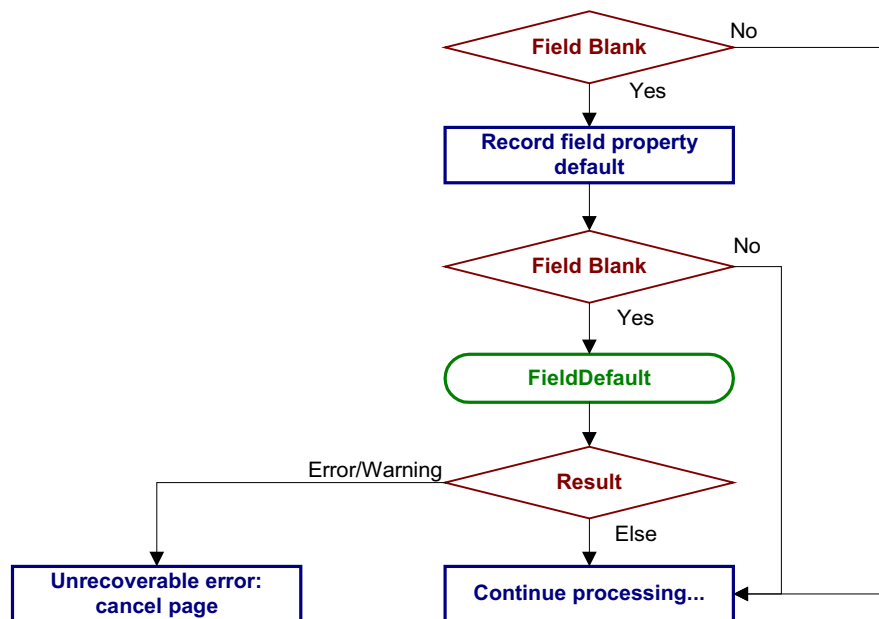
Field-Level Default Processing

During default processing the Component Processor examines all fields in all rows of the component. On each field, it does the following:

1. If the field is set to NULL (blank) for a character field or set to 0 for a numeric field the Component Processor sets the field to any default value specified in the record field properties for that field.
5. If no default value for the field is defined in the record field properties, then the Component Processor fires the FieldDefault event, which triggers any FieldDefault PeopleCode associated with the record field or the component record field.
6. If an **Error** or **Warning** executes in any FieldDefault PeopleCode a runtime error occurs that forces the end-user to cancel the page.

Important! Avoid using **Error** and **Warning** statements in FieldDefault PeopleCode.

The following flowchart shows this logic:



Field-Level default sequence flow

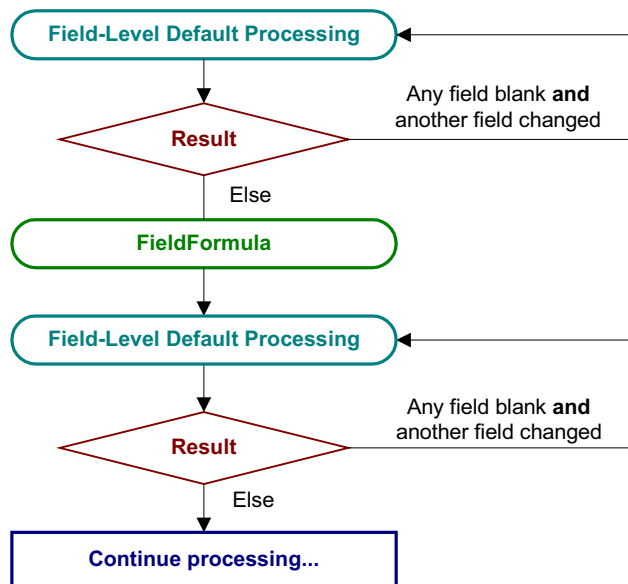
Default Processing on Component Level

Under normal circumstances, default processing in a component is relatively simple: each field on each row of data undergoes Field-Level Default Processing. For typical development

tasks, this is all you need to be concerned with. However, the complete context of default processing is somewhat more complex.

1. Field-Level Default Processing is done on all fields on all rows of data in the component.
7. If any field is still blank *and* any other field in the component has changed, Field-Level Default Processing may be repeated, in case a condition changed that causes default processing to now assign a value to something that was previously left blank.
8. The FieldFormula Event fires on all fields on all rows of data in the component. This PeopleCode event is now often used for FUNCLIB_ (function library) record definitions to store shared functions, so normally no PeopleCode programs execute.
9. *If* the FieldFormula Event changed anything, then Field-Level Default Processing is done again, in case FieldFormula PeopleCode blanked out a field or changed something that causes default processing to now assign a value to something that was previously left blank. Because there shouldn't be any FieldFormula PeopleCode, this is unlikely to affect the development process or performance.
10. Once again, if any field is still blank *and* any other field in the component has changed, Field-Level Default Processing is repeated.

The following flowchart shows this logic:



Default processing on Component Level

Search Processing in Update Modes

If the end-user selects any of the Update action modes (Update, Update/Display All, or Correction), the Component Processor begins Update Mode search processing.

1. The SearchInit PeopleCode event fires, which triggers any SearchInit PeopleCode associated with the record field or the component search record, on the keys or alternate search keys in the component search record. This enables you to control the search dialog field values or the search dialog appearance programmatically, or to perform other processing prior to the search dialog display.

Note. Set the search record for the component in the Component Properties.

For example, the following program in SearchInit PeopleCode on the component search key record field EMPLID sets the search key page field to the user's employee ID, makes the page field unavailable for entry, and enables the user to modify the user's own data in the component:

```
EMPLID = %EmployeeId;
Gray (EMPLID);
AllowEmplIdChg(true);
```

Note. Effects of SetSearchDialogBehavior: Normally a search dialog box is displayed. However, Use SetSearchDialogBehavior to set the behavior of the search dialog box before it is displayed. If SetSearchDialogBehavior is set to Force display, the dialog box is displayed even if all required keys have been provided. You can also set SetSearchDialogBehavior to skip if possible. In addition, you can force Search Processing to always occur by selecting Force Search Processing in the Component Definition Properties in PeopleSoft Application Designer.

11. The search dialog box and prompt list appears, in which the user can enter search keys, or select an Advanced search, to enter alternate search keys.

Note. Effects of SetSearchDefault: Normally the values in the search dialog box are not set to default values. However, if the SetSearchDefault function was executed in SearchInit PeopleCode for any of the search key or alternate search fields, those specific field in the dialog box are set to their system default. No other default processing occurs (that is, the FieldDefault event does not fire).

12. The end-user enters a value or partial value in the search dialog box, then clicks on Search.
13. The SearchSave PeopleCode event fires, which triggers any SearchSave PeopleCode associated with the record field or the component search record, on the search keys or alternate search keys in the search Record. This enables you to validate the user entry in the Search Dialog by testing the value in the search record field in PeopleCode and, if necessary, issuing an error or warning. If an **Error** statement is executed in SearchSave, the end-user is sent back to the search dialog box. If a **Warning** is executed the end-user can click **OK** to continue or click cancel to return to the search dialog box and enter new values.

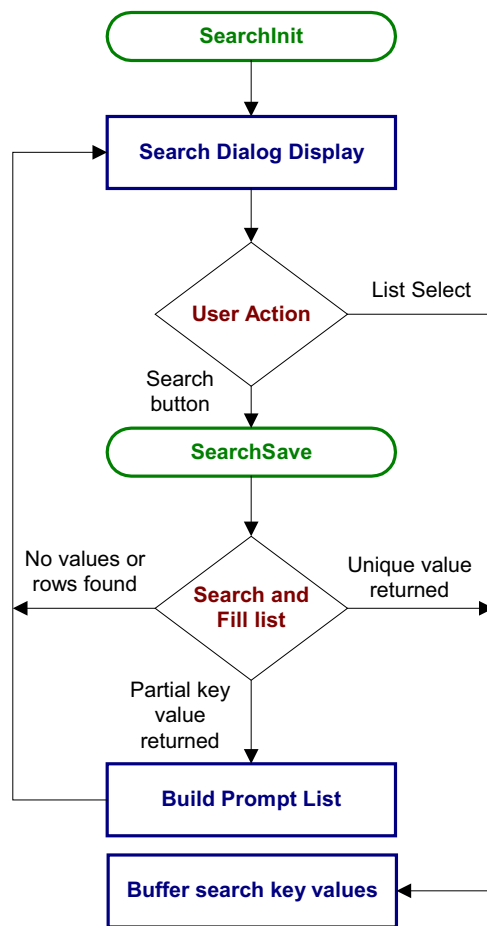
If partial values are entered, such that the Component Processor can select multiple rows, then the Prompt List dialog box is filled and the end-user can select a value. If key values

from the search dialog box are blank or if the system can't select any data based on the end-user entry in the search dialog box, the system displays a message and redisplay the search dialog box. If the values entered produce a unique value, the Prompt List isn't filled; instead, the end-user is taken directly to the page.

Note. Effects of SetSearchEdit: Normally no system edits are applied when the end-user changes a field in the search dialog box. However, if SetSeachEdit is executed for specific search dialog fields in SearchInit PeopleCode, the system edits are applied to those fields after the end-user changes a field and either leaves the field or clicks Search. If the end-user entry in the field fails the system edits, the system displays a message, highlights the field in question, and returns the end-user to the dialog box. The FieldEdit and SaveEdit PeopleCode events do not fire. **Note.** SearchSave does *not* fire after values are selected from the search list. If you need to validate data entered in the search dialog box, use the Component PreBuild event to do so.

14. The Component Processor buffers the search key values. If the end-user then opens another component while this component is active, the Component Processor uses the same search key values and bypasses the search dialog box.

The following flowchart shows this logic. (It does not show the effects of executing the SetSearchDefault and SetSearchEdit functions.)



Search processing logic in update modes

Note. You can use the `IsSearchDialog` function to create PeopleCode that runs only during search processing. To create processes that run only in a specific action mode, use the `%Mode` system variable. This could be useful in code that is part of a library function and that is invoked in places other than from the search dialog box. It could also be used in PeopleCode associated with a record field that appears in pages and in the search dialog.

Search Processing in Add Modesx

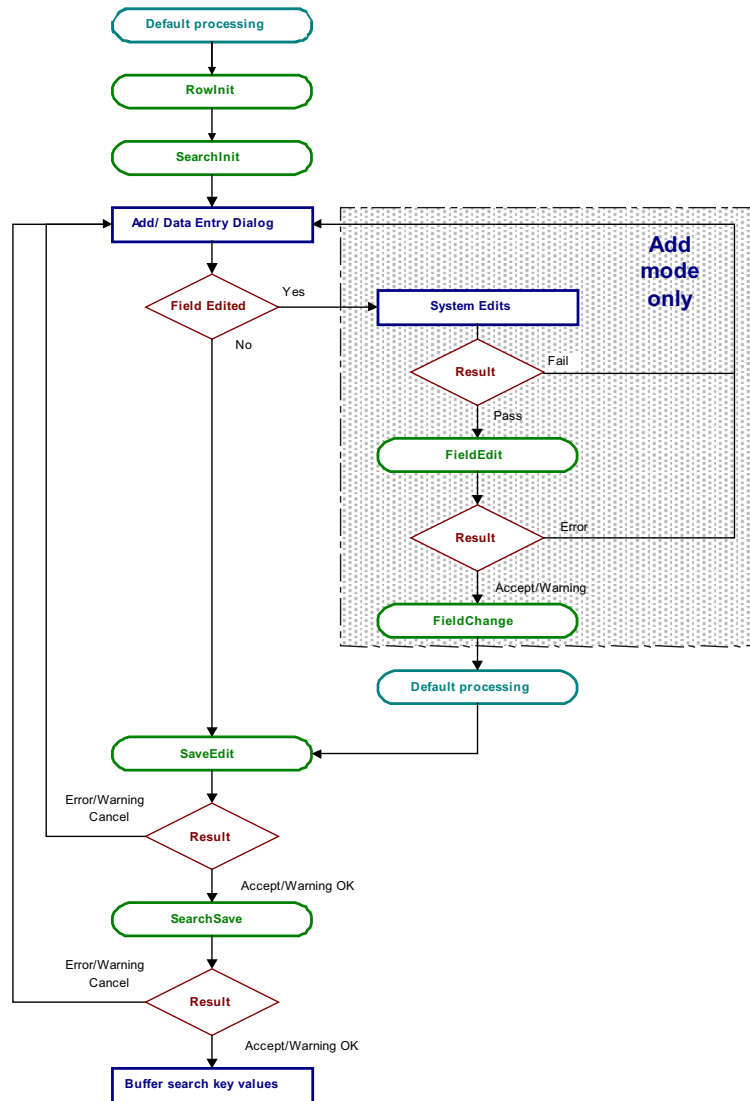
When the end-user starts up a component in Add or Data Entry mode, the Component Processor:

1. Runs default processing on the high-level keys to be displayed in the Add or Data Entry dialog box.
2. Fires the `RowInit` event, which triggers any `RowInit` PeopleCode associated with the record field or the component record, on the Add or Data Entry dialog box fields.

3. Fires the SearchInit event on dialog fields, which triggers any SearchInit PeopleCode associated with the record field or the component search record. This enables you to execute PeopleCode programs before the dialog box displays.
4. Displays the Add or Data Entry dialog.
5. If the end-user changes a dialog field then leaves the field or clicks **OK**:
 - *In Add Mode only* (not Data Entry mode) a Field Modification processing sequence occurs.
 - Default processing is run on the Add or Data Entry dialog field(s). Normally this won't have any effect, because the field(s) will have a value.
6. When the end-user clicks **OK** on the dialog box, this fires the SaveEdit event, which triggers any PeopleCode associated with the record field or the component record.
7. Fires the SearchSave event, which triggers any SearchSave PeopleCode associated with the record field or the component search record. This enables you to validate the end-user entry in the dialog box. If an **Error** statement is executed in SearchSave, the end-user is sent back to the Add or Data Entry dialog box. If a **Warning** is executed the end-user can click **OK** to continue or click **Cancel** to return to the dialog box and enter new values.
8. Buffers the search key values and continues processing.

Note. The flow is easier to understand than it seems. If you compare the following diagram with Search Processing in Update Modes, notice that the add modes are considerably more complex and involve more PeopleCode events. However, in practice, PeopleCode development is similar in both cases: PeopleCode that runs before the dialog appears (for example to control dialog appearance or set values in the dialog fields) generally goes in the SearchInit event; PeopleCode that validates end-user entry in the dialog goes in the SearchSave event.

The following flowchart shows this logic.



Search processing logic in add and data entry modes

Note. You can use the `IsSearchDialog` function to create PeopleCode that runs only during search processing. To create processes that run only in a specific action mode, use the `%Mode` system variable. This could be useful in code that is part of a library function and that is invoked in places other than from the search dialog. It could also be used in PeopleCode associated with a record field that appears in pages and in the search dialog box.

Component Build Processing in Update Modes

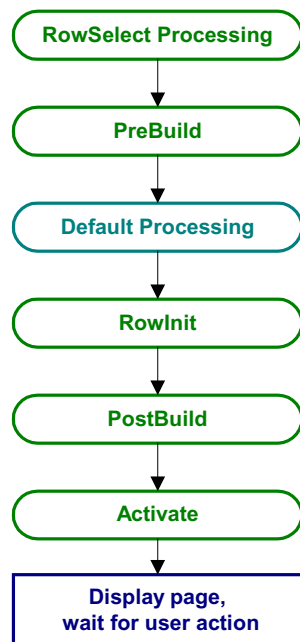
After the Component Processor has saved the search keys values for the component, it uses the search key values to select rows of data from the database server using a SQL Select statement. After the rows are retrieved, the Component Processor:

1. Performs Row Select Processing, in which rows of data that have already been selected from the database server can be filtered before they are added to the component buffer.
2. Fires the PreBuild event, which triggers any PreBuild PeopleCode associated with the component record, enabling you to set global or component scope variables that can be used later by PeopleCode located in other events. PreBuild is also used to validate data entered in the search dialog box, after a prompt list is displayed.

Note. If a PreBuild PeopleCode program issues an error or warning, the end-user is returned to the search page. If there is no search page, that is, the search record has no keys, a blank component page displays.

3. Performs Default Processing on all the rows and fields in the component.
4. Fires the RowInit event, which triggers any RowInit PeopleCode associated with the record field or the component record. The RowInit event enables you to programmatically initialize the values of non-blank fields in the component.
5. Fires the PostBuild event, which triggers any PostBuild PeopleCode associated with the component record, enabling you to set global or component scope variables that can be used later by PeopleCode located in other events.
6. Fires the Activate event, which triggers any Activate PeopleCode associated with the page about to be displayed, enabling you to programmatically control the display of that page.
7. Displays the component and waits for end-user action.

The following flowchart shows this logic.



Component build processing in update modes

Row Select Processing

Row Select processing enables PeopleCode to filter out rows of data after they have been retrieved from the database server using a SQL Select statement and before they are copied to the component buffers.

Row Select processing is a subprocess of Component Build Processing in Add Modes. It also occurs after a ScrollSelect or related function is executed.

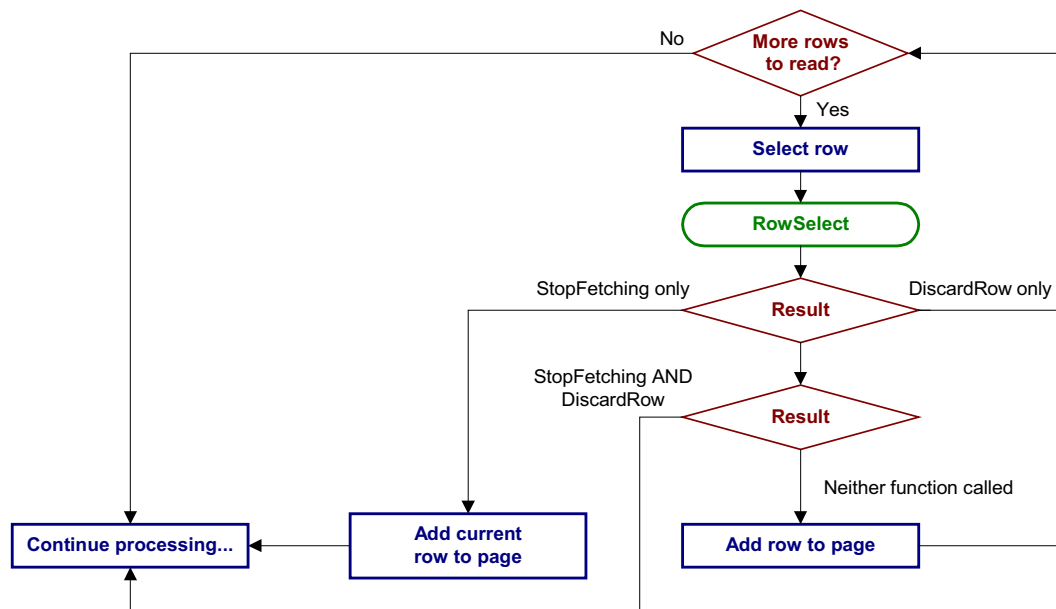
Note. This technique is not often used in recent applications, because it is far more efficient to filter out the rows using a search view, an effective-dated record, the ScrollSelect or a related function, or the Select method, *before* they are sent to the browser.

In Row Select processing the Component Processor:

1. Checks whether there are any more rows to add to the component.
2. Fires the RowSelect event, which triggers any RowSelect PeopleCode associated with the record field or component record. This enables PeopleCode to filter rows using the StopFetching and DiscardRow functions. StopFetching causes the system to add the current row to the component, then stop adding rows to the component. DiscardRow filters out a current row, then continues the Row Select process.
3. If neither StopFetching nor DiscardRow is called, the Component Processor adds the rows to the page and checks for the next row. The process continues until there are no more rows to add to the component buffers. If both StopFetching and DiscardRow are called, the current row is not added to the page, and no more rows are added to the page.

Note. In RowSelect PeopleCode, you can refer only to record fields on the record that is currently being processed because the buffers are in the process of being populated. This means the data might not be present.

The following flowchart shows this logic:



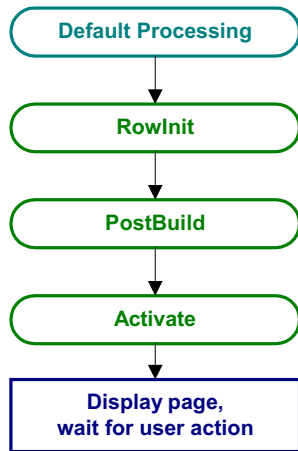
Row select processing logic

Component Build Processing in Add Modes

After search processing in Add or Data Entry modes, the Component Processor:

1. Runs default processing on all page fields. This enables you to set default fields programmatically using FieldDefault PeopleCode.
2. Fires the RowInit event on all fields in the component, which triggers any RowInit PeopleCode associated with the record field or component record. This enables you to initialize the state of page controls using RowInit PeopleCode before they are displayed. (RowInit enables you to set the values of non-blank fields programmatically, whereas default processing is used to set blank fields to their default values.)
3. Fires the PostBuild event, which triggers any PostBuild PeopleCode associated with the component record, enabling you to set global or component scope variables that can be used later by PeopleCode located in other events.
4. Fires the Activate event, which triggers any Activate PeopleCode associated with the page about to be displayed, enabling you to programmatically control the display of that page.
5. Displays a new component using the search keys obtained from the Add or Data Entry dialog with other fields set to their default values.

The following flowchart shows the logic:



Logic of component build processing in add modes

Field Modification

The Field Modification processing sequence occurs after the end-user does any of the following:

- Changes the contents of a field, then leaves the field.
- Changes the state of a radio button or check box.
- Clicks on a command pushbutton.

In this sequence the Component Processor:

1. Performs standard system edits.

To reduce trips to the server, some processing must be done locally on the machine where the browser is located, while some is performed on the server.

Standard system edits can be done either on the browser utilizing local JavaScript code or on the application server. The following chart outlines where these system edits are done.

System Edits	Where Executed
checking data type	Browser
formatting	Application server / browser
updating current or history record	Browser
effective date	Browser
effective date/sequence	Browser
new effective date in range	Browser
duplicate key	Application server

System Edits	Where Executed
current level is not effective-dated but one of its child scrolls is	Browser
required field	Browser
date range	Browser
prompt table	Application server
translate table	Browser
yes/no table	Depends on field type. Browser if the field is a check box. Application server if the field is an edit box and the values are Y or N.

Note. Default processing for the field can be done on the browser *only* if the default value is specified as a constant in the Record Field Properties. If the field contains a default, these defaults occur only upon Component initialization. Then, if a user then blanks out a default value, it is not reinitialized.

Note. The required fields check is *not* performed on derived work fields when you press TAB to move out of a field.

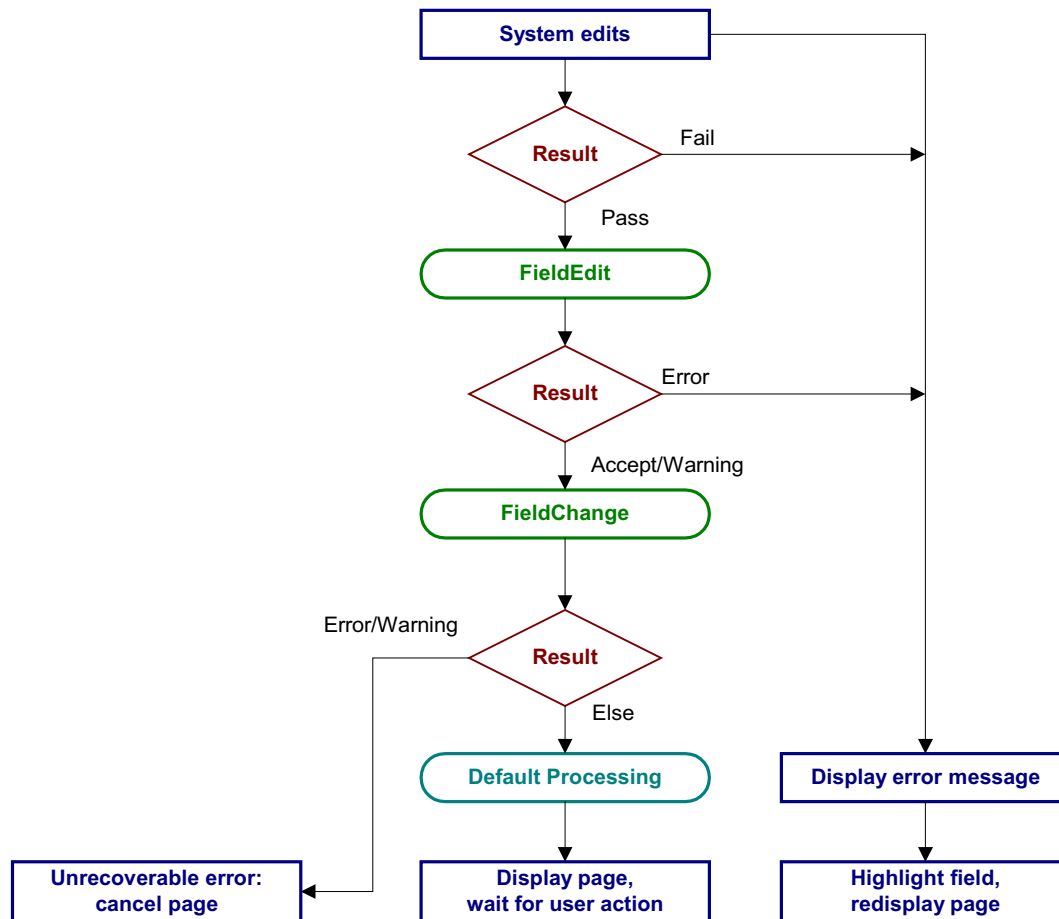
If the data fails the system edits, the Component Processor displays an error message and highlights the field in the system color for errors (usually red).

2. If the field passes the system edits, Component Processor fires the FieldEdit PeopleCode event, which triggers any FieldEdit PeopleCode associated with the record field or the component record field. This enables you to perform additional data validation in PeopleCode. If an **Error** statement is called in any FieldEdit PeopleCode, Component Processor treats the error as it does a system edit failure: a message is displayed and the offending field is highlighted. If a **Warning** statement is executed in any FieldEdit PeopleCode, a warning message appears alerting the end-user to a possible problem, but the system accepts the change to the field.
3. If the field change is accepted, the Component Processor writes the change to the component buffer, then fires the FieldChange event, which triggers any FieldChange PeopleCode associated with the record field or the component record field. This event enables you to add processes other than validation initiated by the changed field value, such as changes to page appearance or recalculation of values in other page fields. An **Error** or **Warning** statement in any FieldChange PeopleCode causes an unrecoverable runtime error and forces cancellation of the page.

Important! PeopleSoft recommends against putting an **Error** or **Warning** statement in any FieldChange PeopleCode. All data validation should be performed in FieldEdit.

After FieldChange processing, Component Processor runs default processing on all page fields, then redisplay the page. If the end-user has blanked out the changed field, or if **SetDefault** or a related function is executed, and the changed field has a default value specified in the record field definition or any FieldDefault PeopleCode, the field is re-initialized to the default value.

The following flowchart shows this logic:



Logic of field modification processing

Row Insert Processing

Row Insert processing occurs when:

- The end-user requests a row insert in a scroll by pressing ALT+7 then pressing ENTER, clicking the Insert Row button, or the New button.
- A PeopleCode **RowInsert** function or a **InsertRow** method requests a row insert.

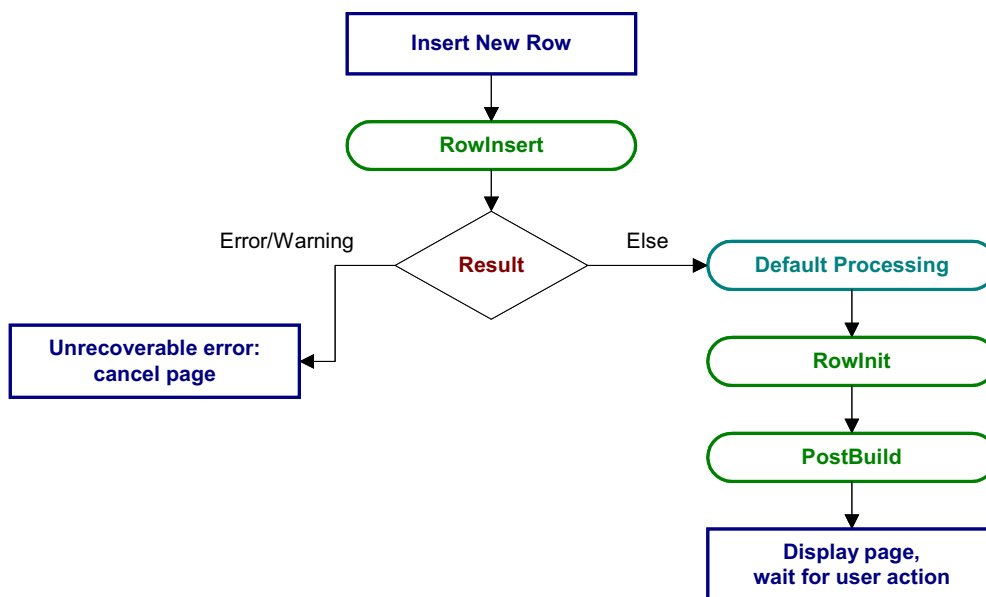
In either case the Component Processor:

1. Inserts a new row of data into the active scroll area. If the scroll area has a dependent scroll area, the system inserts a single new row into the blank scroll area—it continues until it reaches the lowest-level scroll area.
2. Fires the RowInsert PeopleCode event, which triggers any RowInsert PeopleCode associated with the record field or the component record. This event processes fields only on the inserted row, and any dependent rows that were inserted on lower-level scroll areas.

3. Runs default processing on all component fields. Normally this affects only the inserted row fields, and fields on its dependent rows, because other rows already have undergone default processing.
4. Fires the RowInit PeopleCode event, which triggers any RowInit PeopleCode associated with the record field or the component record. This event affects fields only on the inserted row and any dependent rows that were inserted.
5. Redisplays the page and waits for end-user action.

Important! PeopleSoft recommends against putting an **Error** or **Warning** statement in RowInsert PeopleCode. All data validation should be performed in FieldEdit or SaveEdit PeopleCode.

The following flowchart shows this logic:



Logic of row insert processing

Note. If none of the data fields in the new row are changed after the row has been inserted (either programmatically or by the end-user), when the page is saved, the new row isn't inserted into the database.

Row Delete Processing

Row Delete processing occurs when:

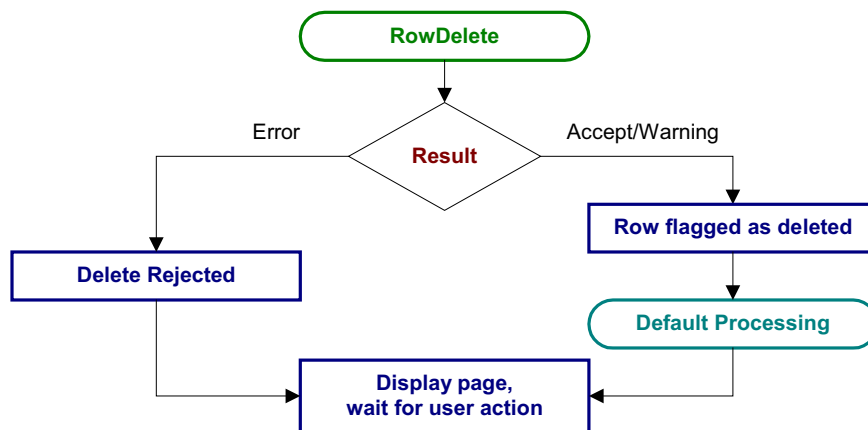
- The end-user requests a row delete in a scroll by pressing ALT+8 then pressing ENTER, clicking the Delete Row button, or clicking the Delete button.
- A PeopleCode **RowDelete** function or a **DeleteRow** method requests a row delete.

In either case the Component Processor:

1. Fires the RowDelete PeopleCode event, which triggers RowDelete PeopleCode associated with the record field or the component record. This event processes fields on the deleted row and any dependent child scrolls. RowDelete PeopleCode enables you to check for conditions and control whether the end-user can delete the row. An **Error** statement displays a message and prevents the end-user from deleting the row. A **Warning** statement displays a message alerting the end-user about possible consequences of the deletion, but permits deletion of the row.
2. If the deletion is rejected the page is redisplayed after the error message.
3. If the deletion is accepted, the row, and any child scrolls dependent on the row, are flagged as deleted. It no longer appears in the page, but it is not physically deleted from the buffer and can be accessed by PeopleCode all the way through the SavePostChange event (note, however, that SaveEdit PeopleCode is not run on deleted rows).
4. Runs default processing on all component fields.
5. Redisplays the page and waits for end-user action

Note. PeopleCode programs are triggered on rows flagged as deleted in SavePreChange and SavePostChange PeopleCode. Use the RecordDeleted function, or the IsDeleted Row property, to test whether a row has been flagged as deleted. You can also access rows flagged as deleted by looping through the rows of a scroll area using a For loop delimited by the value returned by the TotalRowCount function or the RowCount Rowset property.

The following flowchart shows this logic:



Logic of row delete processing

Pushbuttons

When the end-user presses a pushbutton, this initiates the same processing as changing a field. Typically PeopleCode programs started by pushbuttons are placed in the FieldChange event.

See Also

Field Modification

Prompts

No PeopleCode event fires as a result of prompts, returning to the search dialog box or displaying a calendar. This process is controlled automatically by the system.

Pop-Up Menu Display

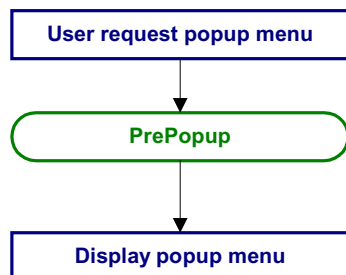
To display a pop-up menu, an end-user can click the pop-up button, either next to a field or at the bottom of a page (if the page has a pop-up menu associated with it.) The end-user can display a standard pop-up menu on a page field if no pop-up menu has been defined by an application developer for that page field.

The PrePopup PeopleCode event fires only if the end-user displays a pop-up menu defined by an application developer on a page field. It doesn't fire before a pop-up menu attached to the page background.

The PrePopup PeopleCode event enables you to disable, check, or hide menu items in the pop-up.

PrePopup PeopleCode menu item operations (such as HideMenuItem, EnableMenuItem, and so on) work with pop-up menus attached to a grid, not a field in a grid, *only* if the PrePopup PeopleCode meant to operate on that pop-up menu resides in the record field that is attached to the *first* column in the grid. It doesn't matter if the first field is visible or hidden.

The following flowchart shows this logic:

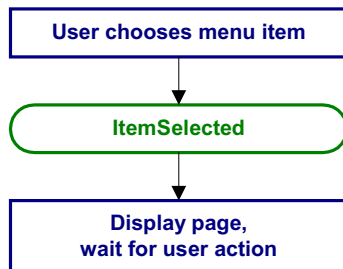


Logic of PrePopup processing

ItemSelected Processing

ItemSelected processing occurs when a end-user selects a menu item from a pop-up menu. This fires the ItemSelected PeopleCode event, which is a menu PeopleCode event.

The following flowchart shows this logic:



Logic of ItemSelected processing

Save Processing

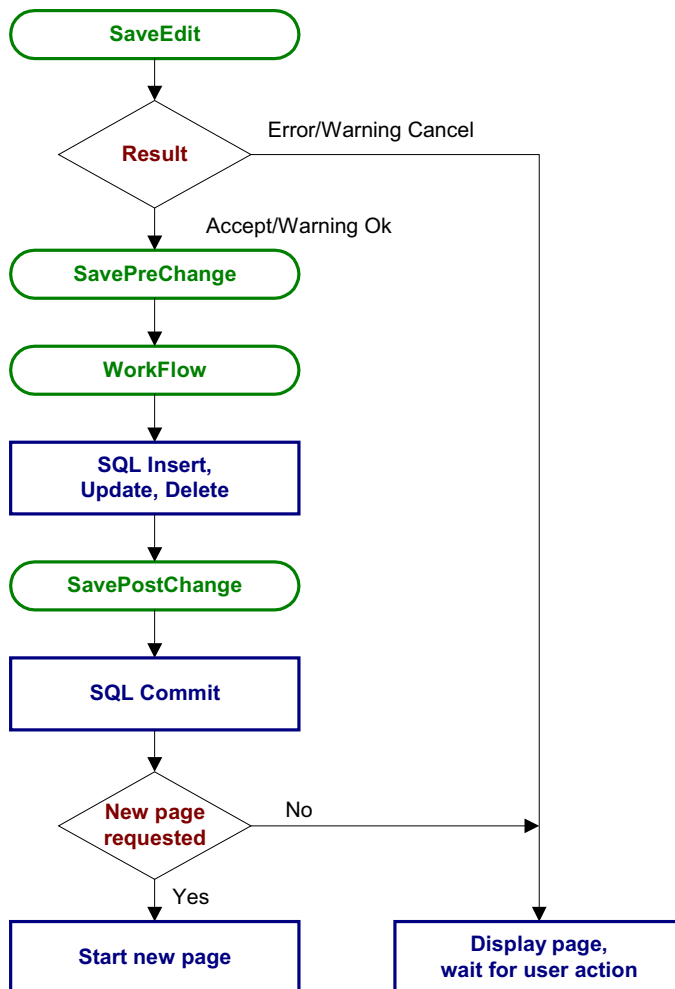
The end-user can direct the system to save a component by clicking Save or by pressing ALT+1, then ENTER. PeopleSoft also prompts the end-user to save a component when the Next or List button is selected, or when a new action or component is selected. In all cases the Component Processor:

1. Fires the SaveEdit PeopleCode event, which triggers any SaveEdit PeopleCode associated with a record field or a component record. This enables you to cross-validate the page fields before saving, checking consistency among the page field values. An **Error** statement in SaveEdit PeopleCode displays a message and then redisplay the page, stopping the save. A **Warning** statement enables the end-user to cancel save processing by clicking **Cancel**, or continue with save processing by clicking **OK**.
2. Fires the SavePreChange event, which triggers any SavePreChange PeopleCode associated with a record field, a component record, or a component. SavePreChange PeopleCode enables you to process data after validation and before the database is updated.
3. Fires the Workflow event, which triggers any Workflow PeopleCode associated with a record field or a component. Workflow PeopleCode should be used only for workflow-related processing (**TriggerBusinessEvent** and related functions).
4. Updates the database with the changed component data, performing any necessary SQL Inserts, Updates, and Deletes.
5. Fires the SavePostChange PeopleCode event, which triggers any SavePostChange PeopleCode associated with a record field, a component record, or a component. You can use SavePostChange PeopleCode for processing that must occur after the database update, such as updates to other database tables not in the component buffer.
6. Issues a SQL Commit to the database server.

7. Depending on whether the end-user has requested a new component, one of these actions occurs:

- re-displays component.
- starts a new component.

Important! Never use an **Error** or **Warning** statement in any Save Processing event other than SaveEdit. All component data validation should be performed in SaveEdit.



Logic of save processing

Exit Component

When the end-user clicks EXIT, the component is canceled, the component buffer is cleared, the Component Processor returns to reset state and the user must login again. No PeopleCode events are fired.

PeopleSoft Internet Architecture Processing Considerations

Keep the following points in mind concerning the PeopleSoft Internet Architecture:

- If an end-user changes a field, but there is nothing to cause a trip to the server on that field, default processing and FieldFormula PeopleCode don't run. They only run when some other event causes a trip to the server.

This means other fields that depend on the first field using FieldFormula or default PeopleCode are not updated until the next time there is a server trip.

- In applications that run on the PeopleSoft portal, external, dynamic hyperlink information *must* be placed in RowInit PeopleCode. If it's placed in FieldChange PeopleCode, it won't work.

Deferred Processing Mode

When a component is running in deferred processing mode, trips to the server are reduced. When deploying some pages in the browser, you may want the user to be able to input data with minimal interruption or trips to the server. Each trip to the server results in the page being refreshed on the browser, which may cause the display to flicker. It can also slow down your application. By specifying a component as Deferred Processing Mode, you can achieve better performance.

If you've specified Deferred Processing Mode for a component, you can then specify whether a page within a component, or a field on a page, will also do their processing in deferred mode. The default is for all pages and fields to allow deferred processing.

Specifying that a field or page allows deferred processing, then *not* setting the component to Deferred Processing Mode, will *not* start deferred processing mode. You must set the component first.

See Application Designer for details about how to set Deferred Processing Mode for components, pages, or fields.

The characteristics of this mode are:

1. Field modification processing is deferred.

No field modification processing is done on the browser. FieldEdit and FieldChange PeopleCode, as well as other edits, such as required field checks, formats, and so on, will not run until a specific user action occurs. Several actions cause field modification processing to execute, for example, clicking a push button or hyperlink, navigating to

another page in the component, and saving the page. The following actions will *not* cause field processing:

- starting an External Link
- clicking a List (Search)
- clicking a Process push button

Deferred processing mode affects the appearance of pages in significant ways. For example, related processing is not done when the user presses TAB to move off a field. Consequently, PeopleSoft recommends avoiding related fields for components that use this mode.

2. Drop-down list box values are static while the page is displayed on the browser.

Drop-down list box values are generated on the application server when generating the HTML for the page.

If translate values are used to populate the drop-down list box and the current record contains an effective date, that date is static while the page is displayed. This means the drop-down list box values may become out of date.

If prompt table values are used to populate the drop-down list box, the high-order key field values for the prompt table are static while the page is displayed. This means the drop-down list box values may become out of date.

PeopleSoft recommends that drop-down lists used on pages executed in deferred mode shouldn't have any interdependencies because the lists may quickly become out of date.

3. No field modification processing is done during prompt button processing.

When the user clicks a prompt button, a trip is made to the application server (if values weren't already downloaded) to select the search results from the database and to generate the HTML for the prompt dialog. During this trip to the application server, field modification processing for the field being prompted is *not* performed because this may cause an error message for another field on the page, and this may confuse the user. When there are deferred changes to other fields, the field modification processing for these fields is done before prompting. The field modification for the prompted field is done after returning from the prompt page. While the page displays the high-order key field values for the prompt table should be static or not require field modification processing. Display-only, drop-down list box, radio button, and check box fields do not require field modification processing. Field values that do not require field modification processing are temporarily written to the Component buffer, without any field modification processing being performed on them, including FieldEdit and FieldChange PeopleCode. The system restores the original state of the page processor before returning to the browser.

4. Field modification processing executes in field layout order.

The entire field modification processing sequence executes in field layout order for each field. If a field passes the system edits and FieldEdit PeopleCode, the field value is written to the component buffer. If an error occurs, field modification processing stops

and the system generates new HTML for the page with the field in error highlighted and sent to the browser.

5. PeopleCode dependencies between fields on the page will not work as expected.

PeopleSoft recommends avoiding PeopleCode dependencies between fields on pages displayed in deferred processing mode. Also, avoid FieldChange PeopleCode that changes the display.

The following are examples of PeopleCode dependencies between fields on the page and the application server's action. In the following examples, field A comes before field B, which comes before field C.

- Field A has FieldChange PeopleCode that hides or makes unavailable for entry field B. The value in field B of the page that was submitted from the browser will be discarded.
- Field B has FieldChange PeopleCode that hides or makes unavailable for entry field A. The change made by the user for field A, if any, remains in the Component buffer.
- Field A has FieldChange PeopleCode that changes the value in the Component buffer for field B. If the value in field B of the page that was submitted from the browser passes the system edits and FieldEdit PeopleCode, it will be written to the Component buffer, overriding the change made by field A's FieldChange PeopleCode.
- Field B has FieldChange PeopleCode that changes the value in the Component buffer for field A. The change made by field B's FieldChange PeopleCode overrides the change made by the user to field A, if any.
- Field A has FieldChange PeopleCode that un-hides or makes available for entry field B. Field B has the value that was already in the Component buffer. If the user requests a different page or finishes, they may not have the opportunity to enter a value into field B and therefore the value may not be correct.
- Field B has FieldChange PeopleCode that changes the value in the Component buffer for field A, but field C has FieldChange PeopleCode that hides or makes unavailable for entry field B. The change made by field B's FieldChange PeopleCode, a field that is now hidden or unavailable for entry, overrides the change made by the user to field A, if any.

There are other examples of PeopleCode dependencies between fields on the page. PeopleSoft recommends avoiding such dependencies by moving FieldChange PeopleCode logic from individual fields to save processing for the Component or FieldChange PeopleCode on a PeopleCode Command pushbutton.

6. Not all pushbuttons cause field modification processing to execute.

Specifically, External Link, List (Search), and Process pushbuttons do not cause field modification processing to execute.

7. You can use a PeopleCode Command pushbutton to cause field modification processing to execute.

An application can include a pushbutton for the sole purpose of causing field modification processing to execute. The result is a new page showing any display changes that resulted from field modification processing.

8. A scroll pushbutton (hyperlink) causes field modification processing to execute.

PeopleCode Events

The preceding sections discussed when sequences of PeopleCode events occur, and under what conditions they trigger PeopleCode programs. The following sections discuss the individual PeopleCode events:

- Activate event.
- FieldChange event.
- FieldDefault event.
- FieldEdit event.
- FieldFormula event.
- ItemSelected event.
- PostBuild event.
- PreBuild event.
- PrePopup event.
- RowDelete event.
- RowInit event.
- RowInsert event.
- RowSelect event.
- SaveEdit event.
- SavePostChange event.
- SavePreChange event.
- SearchInit event.
- SearchSave event.
- Workflow event.

Note. The term *PeopleCode type* is still frequently used, but it does not fit into the new PeopleTools object-based, event-driven metaphor. The term *PeopleCode event* should now be used instead. However, it's often convenient to qualify a class of PeopleCode programs triggered by a specific event with the event name; for example, PeopleCode programs associated with the RowInit events are collectively referred to as **RowInit PeopleCode**.

Activate Event

The Activate event is fired each time the page is activated. This means when the page is first displayed by the end-user, or if a end-user presses TAB between different pages in a component. Each page has its own Activate event.

The main purpose of the Activate event is to segregate the PeopleCode that is related to a specific page from the rest of the application's PeopleCode. PeopleCode related to page display or page processing, such as enabling a field or hiding a scroll, is best put in this event. Also, you can use this event for security validation: if an user doesn't have clearance to view a page in a component, you would put the code for hiding it in this event.

Note. PeopleSoft builds a page grid one row at a time. Because the Grid class applies to a complete grid, you cannot attach PeopleCode that uses the Grid class to events that occur before the grid is built; the earliest event you can use is the Activate event. **Note.** The Activate event isn't associated with a specific row and record at the point of execution. This means you cannot use functions such as GetRecord, GetRow, and so on, that rely on context, without specifying more context.

Activate PeopleCode can only be associated with pages.

This event is valid only for pages that are defined as Standard or Secondary. This event is *not* supported for subpages.

See Also

Component Build Processing in Update Modes

Component Build Processing in Add Modes

FieldChange Event

FieldChange PeopleCode is used to recalculate page field values, change the appearance of page controls, or perform other processing that results from a field change other than data validation. To validate the contents of the field, use FieldEdit Event.

The FieldChange event fires on the specific field and row that just changed.

Do not use **Error** or **Warning** statements in FieldChange PeopleCode: these statements cause a runtime error that forces the end-user to cancel the page without saving changes.

FieldChange PeopleCode is often paired with RowInit PeopleCode. In these RowInit/FieldChange pairs, the RowInit PeopleCode checks values in the component and initializes the state or value of page controls accordingly. FieldChange PeopleCode then rechecks the values in the component during page execution and resets the state or value of page controls.

To take a simple example, suppose you have a Derived/Work field called PRODUCT, the value of which is always the product of page field A and page field B. When the component is initialized you would use RowInit PeopleCode to initialize PRODUCT equal to A * B when the component starts up or when a new row is inserted. You could then attach FieldChange PeopleCode programs to both A and B which also set PRODUCT equal to A * B. Whenever the end-user changes the value of either A or B, PRODUCT would be recalculated.

FieldChange PeopleCode can be associated with record fields and component record fields.

See Also

Field Modification

FieldDefault Event

The FieldDefault PeopleCode event enables you to programmatically set fields to default values when they are initially displayed. This event is fired on all page fields as part of many different processes; however it triggers PeopleCode programs only when the following conditions are all true:

- The page field is still blank after applying any default specified in the record field properties. (This will be true if there is no default specified, if a null value is specified, or if a 0 is specified for a numeric field.)
- The field has a FieldDefault PeopleCode program.

In practice, FieldDefault PeopleCode normally sets fields by default when new data is being added to the component; that is, in Add mode and when a new row is inserted into a scroll.

If a field value is changed, whether through PeopleCode or by an end-user, the IsChanged property for the row is set to True. The exception to this is when a change is done in the FieldDefault or FieldFormula events, that is, if a value is set in FieldDefault or FieldFormula, the row is not marked as changed.

At save time, all newly inserted *and* changed rows are written to the database. All newly inserted but *not* changed rows are *not* written to the database.

You must attach FieldDefault PeopleCode to the specific field that is being defaulted.

Note. An **Error** or **Warning** issued from FieldDefault PeopleCode will cause a runtime error and force cancellation of the component.

FieldDefault PeopleCode can be associated with record fields and component record fields.

See Also

Default Processing

FieldEdit Event

FieldEdit PeopleCode is used to validate the contents of a field, supplementing the standard system edits. If the data does not pass the validation, the PeopleCode program should display a message using the **Error** statement, which redisplay the page, displaying an error message and turning the field red.

If you want to permit the field edit, but alert the end-user to a possible problem, use a **Warning** statement instead of **Error**. A **Warning** statement displays a warning dialog with **OK** and **Explain** buttons. It permits field contents to be changed and continues processing as usual after the end-user clicks **OK**.

If the validation must check the contents of more than one field—that is, if the validation is checking for consistency across page fields—then use SaveEdit PeopleCode instead of FieldEdit.

The FieldEdit event fires on the specific field and row that just changed.

FieldEdit PeopleCode can be associated with record fields and component record fields.

See Also

Field Modification

FieldFormula Event

The FieldFormula event is a vestige of much earlier versions of PeopleTools, and is not used in recent applications. Because FieldFormula PeopleCode fires in many different contexts and triggers PeopleCode on every field on every row in the component buffer, it can seriously degrade the performance of your application. In recent PeopleSoft applications, the RowInit and FieldChange events are used rather than FieldFormula.

If a field value is changed, whether through PeopleCode or by an end-user, the IsChanged property for the row is set to True. The exception to this is when a change is done in the FieldDefault or FieldFormula events, that is, if a value is set in FieldDefault or FieldFormula, the row is not marked as changed.

At save time, all newly inserted *and* changed rows are written to the database. All newly inserted but *not* changed rows are *not* written to the database.

Note. In the PeopleSoft Internet Architecture, if an end-user changes a field, but there is nothing to cause a trip to the server on that field, default processing and FieldFormula PeopleCode do not run. They only run when some other event causes a trip to the server.

As a matter of convention, FieldFormula is now often used in FUNCLIB_ (function library) record definitions to store shared functions. This is purely a matter of convention, and in fact you can store shared functions in any PeopleCode event.

FieldFormula PeopleCode is only associated with record fields.

Important! Do not use FieldFormula PeopleCode in your components. Use it only to store external PeopleCode functions in FUNLIB_ record definitions.

ItemSelected Event

The ItemSelected event fires whenever the end-user selects a menu item from a pop-up menu. In pop-up menus ItemSelected PeopleCode executes in the context of the page field from where the pop-up menu is attached, which means that you can freely reference and change page fields, just as you could from a pushbutton.

Note. This event, and all it's associated PeopleCode, will not fire if run from a Component Interface.

ItemSelected PeopleCode is only associated with pop-up menu items.

See Also

ItemSelected Processing

PostBuild Event

The PostBuild event fires after all the other component build events have fired. This event is often used to hide or unhide pages. It's also used to set component variables.

PostBuild PeopleCode is only associated with components.

PreBuild Event

The PreBuild event fires before the rest of the component build events. This event is often used to hide or unhide pages. It's also used to set component variables.

Note. If a PreBuild PeopleCode program issues an error or warning, the end-user is returned to the search page. If there is no search page, that is, the search record has no keys, a blank component page displays.

The PreBuild event is also used to validate data entered in the search dialog, after a prompt list is displayed. For example, after the end-user selects key values on the search, the PreBuild

PeopleCode program fires, which catches the error condition and issues an error message. The end-user receives and acknowledges an error message. The component is cancelled (because of the error) and the end-user is returned to the Search dialog box. PreBuild PeopleCode is only associated with components.

PrePopup Event

The PrePopup event fires just before the display of a pop-up menu.

You can use PrePopup PeopleCode to control the appearance of the Pop-up menu.

Note. This event, and all it's associated PeopleCode, will not fire if run from a Component Interface.

PrePopup PeopleCode can be associated with record fields and component record fields.

See Also

Pop-up Menu Display, DisableMenuItem, EnableMenuItem, HideMenuItem

RowDelete Event

The RowDelete event fires whenever a end-user attempts to delete a row of data from a page scroll. Use RowDelete PeopleCode to prevent the deletion of a row (using an **Error** or **Warning** statement) or to perform any other processing contingent on row deletion. For example, you could have a page field TOTAL on scroll level zero whose value is the sum of all the EXTENSION page fields on scroll level one. If the end-user deleted a row on scroll level one you could use RowDelete PeopleCode to recalculate the value of TOTAL.

The RowDelete event triggers PeopleCode on any field on the row of data that is being flagged as deleted.

Note. RowDelete does not trigger programs on Derived/Work records.

RowDelete PeopleCode can be associated with record fields and component records.

Considerations When Deleting all Rows From a Scroll

When the last row of a scroll is deleted, a new, dummy row is automatically added. As part of the RowInsert event, RowInit PeopleCode is run on this dummy row. If a field is changed by RowInit (even if it's left blank) the row is no longer New, and therefore won't be reused by any of the ScrollSelect functions or the Select method. In this case, you may want to move your initialization code from the RowInit event to FieldDefault.

See Also

Row Delete Processing

"Using Methods and Built-In Functions," Errors and Warnings in RowDelete

RowInit Event

The RowInit event fires the first time the Component Processor encounters a row of data. It is used for setting the initial state of component controls. This occurs during component build processing and row insert processing. It also occurs after a ScrollSelect or related function is executed.

Note. Generally, if none of the fields in the new row are changed after the row has been inserted (either by the end-user pressing ALT-7 and ENTER, or programmatically), when the page is saved, the new row isn't inserted into the database. However, if the ChangeOnInit property is set to False, you can set values for fields a new row in RowInsert or RowInit PeopleCode, and the row won't be saved.

RowInit is not field-specific: it triggers PeopleCode on all fields and on all rows in the component buffer.

Do not use **Error** or **Warning** statements in RowInit PeopleCode: these cause a runtime error and force the end-user to cancel the component without saving.

RowInit PeopleCode is often paired with FieldChange PeopleCode. In these RowInit/FieldChange pairs, the RowInit PeopleCode checks values in the component and initializes the state or value of page controls accordingly. FieldChange PeopleCode then rechecks the values in the component during page execution and resets the state or value of page controls.

To take a simple example, suppose you have a Derived/Work field called PRODUCT, the value of which is always the product of page field A and page field B. When the component is initialized you would use RowInit PeopleCode to initialize PRODUCT equal to $A * B$ when the component starts up or when a new row is inserted. You could then attach FieldChange PeopleCode programs to both A and B which also set PRODUCT equal to $A * B$. Whenever the end-user changes the value of either A or B, PRODUCT would be recalculated.

RowInit PeopleCode can be associated with record fields and component records.

See Component Build Processing in Add Modes, Component Build Processing in Update Modes.

Exception to RowInit Firing

In certain rare circumstances, the Component Processor doesn't run RowInit PeopleCode for some record fields. The Component Processor runs RowInit PeopleCode when it loads the record from the database. However, in some cases, the record can be initialized entirely from the keys for the component. When this happens, RowInit PeopleCode isn't run.

For RowInit to not run, the following must *all* be true. If any field of the record doesn't meet these conditions, the record must be read from the database, and so RowInit PeopleCode runs.

- The record is at level 0.
- Every record field that is present in the data buffers is also present in the keys for the component.
- Every record field that is present in the data buffers is display-only.

The qualification "that is present in the data buffers" means that the Component Processor determines the field is required by the component. In practice, this usually means that the field is associated with a page field, possibly hidden, for some page of the component. It could also mean that the field is referenced by some PeopleCode program that is attached to an event on some other field of the component.

RowInit not running isn't considered to be an error. The purpose of RowInit PeopleCode is to complete initialization of data on the row after it has been read *from the database*. Because the data in this special circumstance is coming from the keylist, it has already been initialized correctly by whatever processing produced the keylist. More general initialization of the component should be done in PostBuild PeopleCode, not RowInit.

RowInsert Event

When the end-user adds a row of data, the Component Processor generates a RowInsert event. You should use RowInsert PeopleCode for processing specific to the insertion of new rows. Do not put PeopleCode in RowInsert that already exists in RowInit, because a RowInit event always fires after the RowInsert event, which will cause your code to be run twice.

Note. Generally, if none of the fields in the new row are changed after the row has been inserted (either by the end-user pressing ALT-7 and ENTER, or programmatically), when the page is saved, the new row isn't inserted into the database. However, if the ChangeOnInit property is set to False, you can set values for fields a new row in RowInsert or RowInit PeopleCode, and the row won't be saved.

The RowInsert triggers PeopleCode on any field on the inserted row of data.

Do not use a **Warning** or **Error** in RowInsert: this will cause a runtime error and force cancellation of the component.

You can prevent the end-user from inserting rows into a scroll area by selecting the **No Row Insert** check box in the scroll bar's Page Field Properties; however, you can't prevent row insertion conditionally.

The screenshot shows the 'Page Field Properties' dialog box with the 'General' tab selected. The 'Scroll Attributes' section has 'Occurs Level' and 'Occurs Count' both set to 1. The 'Field Use Options' section has 'Invisible', 'No Auto Select', and 'No Row Insert' unchecked, while 'Default Width', 'No Auto Update', and 'No Row Delete' are checked. The 'Scroll Action Buttons' section has 'Previous Page', 'Row Insert', 'Top', and 'Show Row Counter' unchecked, while 'Next Page', 'Row Delete', and 'Bottom' are checked. The 'Popup Menu' is empty. The 'Field Help Context Number' is empty with an '< Auto Assign' button. The 'Allow Deferred Processing' checkbox is checked. 'OK' and 'Cancel' buttons are at the bottom.

No Row Insert in Properties of the Scroll Bar

Note. RowInsert does not trigger PeopleCode on Derived/Work fields.

RowInsert PeopleCode can be associated with record fields and component records.

See Also

Row Insert Processing

RowSelect Event

The RowSelect event fires at the beginning of the Component Build process in any of the Update action modes (Update, Update/Display All, Correction). RowSelect PeopleCode is used to filter out rows of data as they are being read into the component buffer. This event also occurs after a ScrollSelect or related function is executed.

A **DiscardRow** function in RowSelect PeopleCode causes the Component Processor to skip the current row of data and continue to process other rows. A **StopFetching** statement causes the Component Processor to accept the current row of data, then stop reading additional rows. If both statements are executed, the program skips the current row of data, then stops reading additional rows.

PeopleSoft applications rarely use RowSelect, because it's inefficient to filter out rows of data after they've already been selected. Recent applications screen out rows of data using search record views and effective-dated tables, which filter out the rows before they're selected. You could also use a ScrollSelect or related function to programmatically select rows of data into the component buffer.

In previous versions of PeopleTools the **Warning** and **Error** statements were used instead of **DiscardRow** and **StopFetching**. **Warning** and **Error** will still work as before in RowSelect, but their use is discouraged.

Note. In RowSelect PeopleCode, you can refer to record fields only on the record that is currently being processed. **Note.** This event, and all its associated PeopleCode, will not fire if run from a Component Interface.

RowSelect PeopleCode can be associated with record fields and component records.

See Also

Row Select Processing

SaveEdit Event

The SaveEdit event fires whenever the end-user attempts to save the component. You can use SaveEdit PeopleCode to validate the consistency of data in component fields. Whenever a validation involves more than one component field, you should use SaveEdit PeopleCode. If a validation involves only one page field, you should use FieldEdit PeopleCode.

SaveEdit is not field-specific: it triggers associated PeopleCode on every row of data in the component buffers, except rows flagged as deleted.

An **Error** statement in SaveEdit PeopleCode displays a message and redisplay the component without saving data. A **Warning** enables the end-user to click OK and save the data, or click Cancel and return to the component without saving.

Use the SetCursorPos function to set the cursor position to a specific page field following a **Warning** or **Error** in SaveEdit, to show the end-user the specific field (or at least one of the fields) that is causing the problem. Make sure to call **SetCursorPos** before the **Error** or **Warning** (because these may terminate the PeopleCode program).

SaveEdit PeopleCode can be associated with record fields and components.

See Also

Save Processing

SavePostChange Event

After the Component Processor updates the database, it fires the SavePostChange event. You can use SavePostChange PeopleCode to update tables not in your component using the SQLExec built-in function.

An **Error** or **Warning** in SavePostChange PeopleCode causes a runtime error, forcing the end-user to cancel the component without saving changes. Avoid **Errors** and **Warnings** in this event.

The system issues a SQL Commit after SavePostChange PeopleCode completes successfully.

If you are executing WorkFlow PeopleCode, keep in mind that if the WorkFlow PeopleCode fails, SavePostChange PeopleCode will not be executed. If your component has both WorkFlow and SavePostChange PeopleCode, consider moving the SavePostChange PeopleCode to SavePreChange or WorkFlow.

If you are doing application messaging, your **Publish()** PeopleCode should go into this event.

Important! Never issue a SQL Commit or a Rollback manually from within a **SQLExec** function. Let the Component Processor issue these SQL commands.

SavePostChange PeopleCode can be associated with record fields, components, and component records.

See Also

Save Processing

SavePreChange Event

The SavePreChange event fires after SaveEdit completes without errors. SavePreChange PeopleCode provides one final opportunity to manipulate data before the system updates the database; for instance, you could use SavePreChange PeopleCode to set sequential high-level keys. If SavePreChange runs successfully, a WorkFlow event is generated, then the Component Processor issues appropriate INSERT, UPDATE, and/or DELETE SQL commands.

SavePreChange PeopleCode is not field-specific: it triggers PeopleCode on all fields and on all rows of data in the component buffer.

SavePreChange PeopleCode can be associated with record fields, components, and component records.

See Also

Save Processing

SearchInit Event

The SearchInit event is generated just before a search dialog, add dialog, or data entry dialog box is displayed. SearchInit triggers associated PeopleCode in the search key fields of the search record. This enables you to control processing before the end-user enters values for search keys in the dialog box. In some cases, you may want to set the value of the search dialog fields programmatically. For example, the following program in SearchInit PeopleCode on the component search key record field EMPLID sets the search key page field to the user's employee ID, makes the page field unavailable for entry, and enables the user to modify the user's own data in the component:

```
EMPLID = %EmployeeId;
Gray (EMPLID);
AllowEmplIdChg(true);
```

You can activate system defaults and system edits during the search dialog by calling **SetSeachDefault** and **SetSearchEdit** in SearchInit PeopleCode. You can also control the behavior of the search dialog box, either forcing it to display even if all the required keys have been provided, or skipping it if possible, with the **SetSeachDialogBehavior** function. You can also force search processing to always occur by selecting the Force Search Processing check box on the component properties in PeopleSoft Application Designer.

Note. This event, and all it's associated PeopleCode, will not fire if run from a Component Interface.

SearchInit PeopleCode can be associated with record fields and component search records.

SearchInit PeopleCode Function Restrictions

You can't use the following functions in SearchInit PeopleCode:

- DoModal
- DoModalComponent
- Transfer
- TransferPage

See Also

SetSearchDefault, SetSearchEdit, SetSearchDialogBehavior

Search Processing in Update Modes

Search Processing in Add Modes

SearchSave Event

SearchSave PeopleCode is executed for all search key fields on a search dialog, add dialog, or data entry dialog box after the end-user clicks Search. This enables you to control processing after search key values are entered, but before the search based on these keys is executed. A typical use of this feature is to provide cross-field edits for selecting a minimum set of key information. It is also used to force the user to enter a value in at least one field, even if it's a partial value to help narrow a search for tables with many rows.

Note. SearchSave does *not* fire when values are selected from the search list. If you need to validate data entered in the search dialog box, use the Component PreBuild event to do so.

You can use **Error** and **Warning** statements in SearchSave PeopleCode to send the end-user back to the search dialog box if the end-user entry does not pass validations implemented in the PeopleCode.

Note. This event, and all its associated PeopleCode, will not fire if run from a Component Interface.

SearchSave PeopleCode can be associated with record fields and component search records.

See Also

Search Processing in Update Modes

Search Processing in Add Modes

Workflow Event

Workflow PeopleCode executes immediately after SavePreChange and before the database update that precedes SavePostChange. The main purpose of the Workflow event is to segregate PeopleCode related to Workflow from the rest of the application's PeopleCode. Only PeopleCode related to Workflow (such as **TriggerBusinessEvent**) should be in Workflow programs. Your program should deal with Workflow only after any SavePreChange processing is complete.

Workflow PeopleCode is not field-specific: it triggers PeopleCode on all fields and on all rows of data in the component buffer.

WorkFlow PeopleCode can be associated with record fields and components.

See Also

Save Processing

PeopleTools PeopleBook: Peoplesoft Workflow, "Defining Event Triggers," Writing Workflow PeopleCode

PeopleCode Execution in Multiple Scroll Pages

Components with multiple occurs levels can have multiple rows of data from multiple primary record definitions. You must know the order in which the system processes buffers for this data, because it applies PeopleCode in the same order.

The Component Processor uses a "depth-first" algorithm to process rows in multiple-scroll pages, starting with a row at level zero and drilling down to dependent rows on lower levels, then working its way up the hierarchy until it has processed all the dependent rows of the last row on the highest level.

Scroll Level One

When pages have only one scroll bar, the Component Processor processes buffers in a straightforward manner. It processes record definitions at scroll level zero, then all rows of data at scroll level one.

Data is retrieved for all rows with a single SELECT statement, and then merged with buffer structures.

Scroll Level Two

With scroll bars at multiple scroll levels, the system does the same for each scroll-level-one row. It processes a single row of data at scroll level one, then processes all its subordinate rows of data at scroll level two. After processing all subordinate data at scroll level two, it processes the next row for scroll level one, and all the subordinate data for that row. It continues in this fashion until it processes everything.

Scroll Level Three

The Component Processor uses the same method for processing subordinate data at scroll level three. Data is retrieved for all rows with a single SELECT statement, and then merged with buffer structures. When it processes a single row of data at scroll level two, it processes all subordinate data before processing the next scroll-level-two row.

See Also

"Referencing Data in the Component Buffer," Component Buffer Structure and Contents

CHAPTER 12

PeopleCode and PeopleSoft Internet Architecture

The chapter provides an overview of the PeopleCode considerations when writing PeopleSoft Internet Architecture applications and discusses how to:

- Use PeopleCode in the PeopleSoft Internet Architecture.
- Call dynamic link library functions on the application server.
- Update the Installation and PSOPTIONS tables.

Using PeopleCode in the PeopleSoft Internet Architecture

Take the following considerations into account when writing PeopleCode programs for the PeopleSoft Internet Architecture:

- To help your application run efficiently, avoid using field-level PeopleCode events (that is, **FieldEdit** and **FieldChange**).

If you must use field-level PeopleCode events, such as **FieldChange** and **FieldEdit** in your applications, keep the following in mind: each time you run a field-level PeopleCode program, it requires a trip to the application server to run the PeopleCode program.

However, the majority of PeopleCode programs naturally run on the application server as part of the Component build and save process. Do not hesitate to use PeopleCode for building and saving components.

- If an end-user changes a field, but there is nothing to cause a trip to the server on that field, default processing and FieldFormula PeopleCode do not run. They only run when some other event causes a trip to the server.

This means other fields that depend on the first field using FieldFormula or default PeopleCode are not updated until the next time there is a server trip.

- In applications that run on the PeopleSoft portal, external, dynamic hyperlink information *must* be placed in RowInit PeopleCode. If it's placed in FieldChange PeopleCode, it won't work.

- Trips to the server are reduced when a component is running in deferred processing mode. Each trip to the server results in the page being completely refreshed on the browser, which may cause the display to flicker. It can also slow down your application. By specifying a component as Deferred Processing Mode, you can achieve better performance.

See Also

"PeopleCode and the Component Processor," Deferred Processing Mode

Using PeopleCode to Populate Search Dialog Key Fields

In a PeopleSoft Internet Architecture application you typically want users to directly access their own data. To facilitate this, you may want to use SearchInit PeopleCode to populate and then make unavailable for entry standard search dialog key fields. You might assign the search key field a default value based on the user ID or alias the user entered at logon.

You must also call **AllowEmplIdChg**, allowing users to change their own data. This function takes a single Boolean parameter in which you pass True to allow employees to change their own data.

Here is a simple example of such a SearchInit program, using %EmployeeId to identify the user:

```
EMPLID = %EmployeeId;
Gray (EMPLID);
AllowEmplIdChg(true);
```

Calling Dynamic Link Library Functions on the Application Server

To support processes running on an application server, you can declare and call functions compiled in Windows dynamic link libraries and UNIX shared libraries (or shared objects, depending on the specific UNIX platform). You can do this either with a special PeopleCode declaration, or using the Business Interlink framework.

When you call out to a DLL using PeopleCode, on Windows NT application servers, the DLL file has to be on the path. On UNIX application servers, the shared library file must be on the *library path* (as defined for the specific UNIX platform).

The PeopleCode declaration and function call syntax remains unchanged. For example, the following PeopleCode could be used to declare and call a function **LogMsg** in an external library Testdll.dll on a Windows client or a Windows application server, or a libtestdll.so on an UNIX application server. The UNIX shared library's extension varies by the specific UNIX platform.

```
Declare Function LogMsg Library "testdll" (string, string)
    Returns integer;

&res = LogMsg("\temp\test.log", "This is a test");
```

See Also

PeopleTools PeopleBook: PeopleSoft Business Interlink Application Developer Guide

Sample Cross-Platform External Test Function

Following is the C source code for a sample cross-platform test file. It is a basic function that just opens a log file and appends a line to it. If you compile the code using a C++ compiler, the functions must be declared using external "C" to ensure C-language linkage.

This file contains an interface function required for non-Windows environments. This function is compiled conditionally: only if you are compiling for a non-Windows environment (for example, UNIX). The interface function references a provided header file, pcmext.h. The interface function is passed type codes that can be optionally used for parameter checking.

```

/*
 * Simple test function for calling from PeopleCode.
 * This is passed two strings, a file name and a message.
 * It creates the specified file and writes the message
 * to it.
 */
#include <stdlib.h>
#include <stdio.h>
#include <string.h>

#ifdef _WINDOWS
#define DLLEXPORT __declspec(dllexport)
#define LINKAGE __stdcall
#else
#define DLLEXPORT
#define LINKAGE
#endif

DLLEXPORT int LINKAGE LogMsg(char * fname, char * msg);

/*****
 * PeopleCode External call test function.
 *
 * Parameters are two strings (filename and message)
 * Result is 0 if error, 1 if OK
 *
 * To call this function, the following PeopleCode is
 * used
 *
 * Declare Function LogMsg Library "testdll"
 * (string, string)
 * Returns integer;
 *
 * &res = LogMsg("\temp\test.log", "This is a test");
 *****/

DLLEXPORT int LINKAGE LogMsg(char * fname, char * msg)
{
    FILE *fp;

    fp = fopen(fname, "a");          /* append */
    if (fp == NULL) return 0;

```



```

        fprintf(fp, "%s\n", msg);
        fclose(fp);
        return 1;
    }

#ifdef _WINDOWS

/*****
 *   Interface function.
 *
 *   This is not needed for Windows...
 *
 *****/

#include "pcmext.h"
#include "assert.h"

void LogMsg_intf(int nParam, void ** ppParams, EXTPARAMDESC * pDesc)
{
    int    rc;

    /* Some error checking */
    assert(nParam == 2);
    assert(pDesc[0].eExtType == EXTTYPE_STRING
           && pDesc[1].eExtType == EXTTYPE_STRING
           && pDesc[2].eExtType == EXTTYPE_INT);

    rc = LogMsg((char *)ppParams[0],
                (char *)ppParams[1]);
    *(int *)ppParams[2] = rc;
}

#endif

```

Updating the Installation and PSOPTIONS Tables

When an application updates either the PSOPTIONS or the Installation table it must call **UpdateSysVersion** from **SavePreChange** PeopleCode event. This way the updates take effect at the next page load. Otherwise, the change does not take effect at the client workstation until the user logs off and logs back on.

Important! Making changes to the Installation and PSOPTIONS tables is not a simple task. Only a database administrator or the equivalent person in your organization should make changes to these tables.

CHAPTER 13

Debugging Your Application

This chapter provides an overview of the PeopleCode Debugger and discusses how to:

- Access the PeopleCode Debugger.
- Understand the PeopleCode Debugger features.
- Understand the PeopleCode Debugger options.
- Set Up the debugging environment.
- Debug Subscription PeopleCode.
- Compile all PeopleCode programs.
- Set PeopleCode Debugger Log options.
- Interpret the PeopleCode Debugger Log file.
- Use the Find In feature.
- Use Cross Reference Reports.

Understanding the PeopleCode Debugger

The PeopleCode Debugger is an integrated part of PeopleSoft Application Designer. The interface to the Debugger has a visual indicator of breakpoints, an arrow indicating the current line and the ability to step through code. You can inspect the value of a variable by 'hovering' your cursor over it and reading the pop-up bubble help. The Debugger also provides variable inspection windows for Globals, Locals, Function Parameters, and Component scoped variables. It also enables PeopleCode objects to be "expanded", so you can inspect their component parts.

Accessing the PeopleCode Debugger

You access the Debugger through PeopleSoft Application Designer. You can start a debugging session either before or after you start a PeopleSoft component.

To start the PeopleCode Debugger:

1. Open PeopleSoft Application Designer.
2. Select Debug, PeopleCode Debugger Mode.

The Local Variables watch window opens. PeopleCode programs that had breakpoints set from your previous debugging session are opened also, and the breakpoints are restored.

Note. If you've already opened the debugger, then closed it, the menu may not morph correctly to enable you to access the Debugger a second time. If this occurs, click the Local Variables window, then try the debug menu again.

You must make some adjustments to the system for running the Debugger in PeopleSoft Internet Architecture or with Application Message Subscription PeopleCode.

Note. Your security administrator has options for allowing users to access different parts of PeopleSoft Application Designer, including the PeopleCode Debugger. If you're having problems accessing the Debugger, you may need to contact your system administrator concerning your security access.

See Also

Setting Up the Debugging Environment

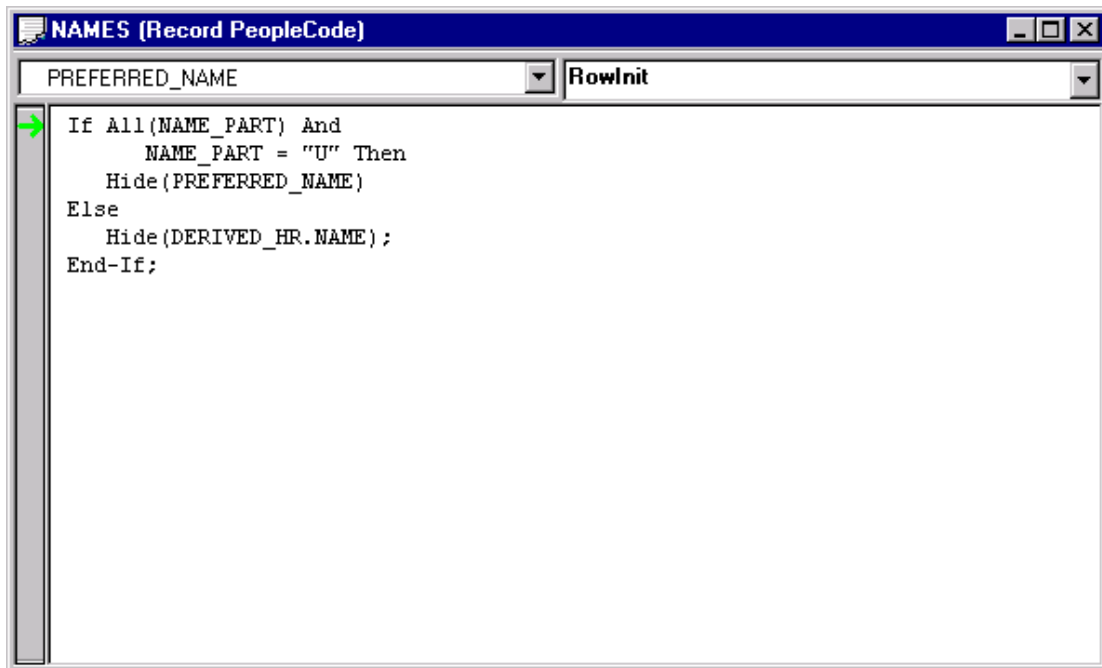
Understanding the PeopleCode Debugger Features

In this section, we discuss the following features of the PeopleCode Debugger:

- Visible current line of execution.
- Visible breakpoints.
- Hover inspect.
- Single debugger.
- Variable panes.

Visible Current Line of Execution

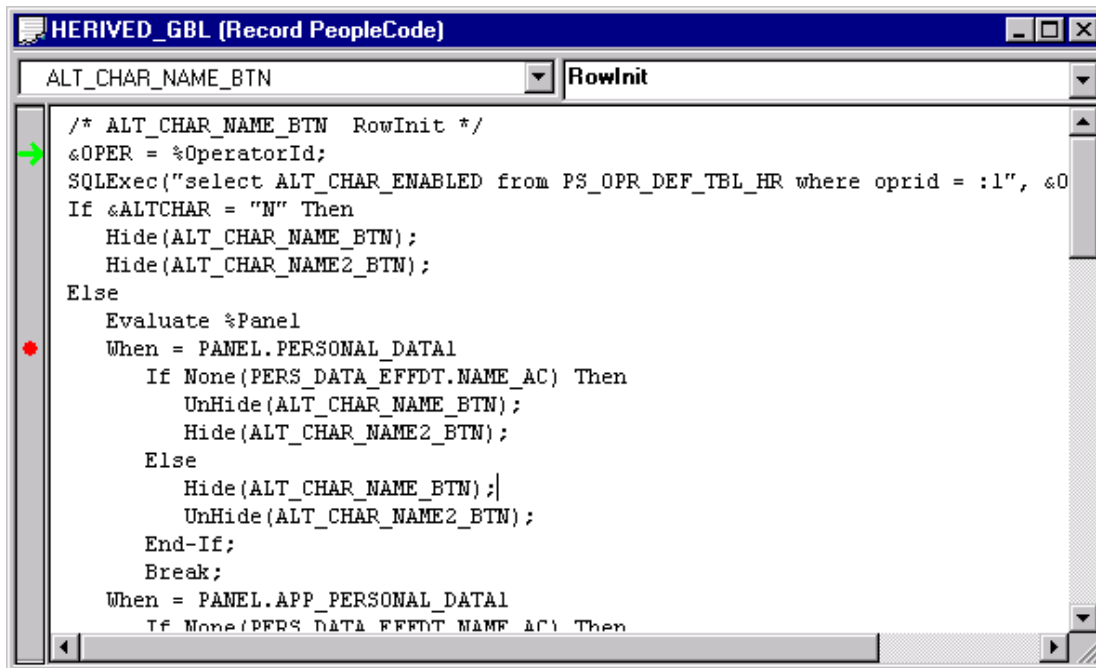
The current line indicator (green arrow displayed in left-hand gutter) is shown in the following illustration:



PeopleCode Debugger showing current line of execution

Visible Breakpoints

The PeopleCode Debugger supports visual indicators that signify breakpoint locations. In the following example the current line indicator (green arrow) is shown at the first line, and the breakpoint (red dot displayed in left-hand gutter) is on line 8:



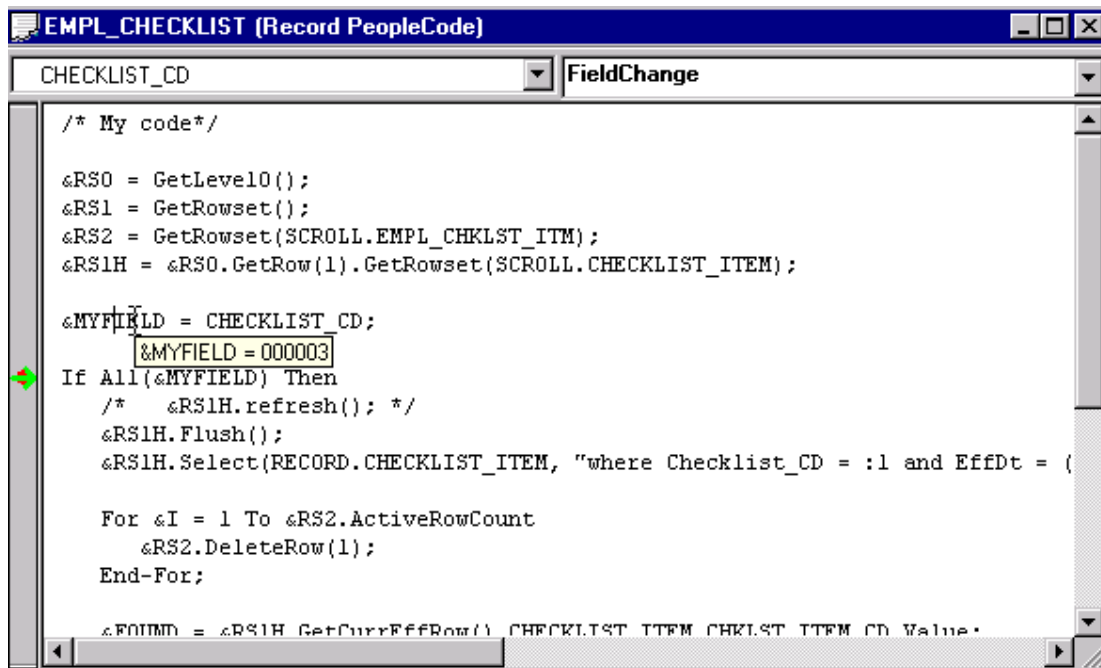
PeopleCode Debugger showing breakpoint and current line of execution

All breakpoints are saved when Exit Debug Mode is selected.

Note. You cannot set breakpoints on function declarations, variable declarations, or in comments.

Hover Inspect

If the program is already running, you can see the actual values for the variables by "hovering" the cursor over them. The current value is displayed in a pop-up window.



PeopleCode Debugger showing hover inspect

Hover inspect is implemented only for simple variables and fields.

Hover inspect is *not* implemented for object expressions (for example, rowset assignments, array assignments, and so on.)

Single Debugger

Each PeopleSoft session you run on a machine can have its own debugging session. However, there can only be one instance of the PeopleCode debugger *per session*. If more than one instance of PeopleSoft Application Designer is running for a session, only one may be the active debugger at a given time.

From within a running instance of PeopleSoft Application Designer, any component in the same session is also placed into debug mode.

After the session is in debug mode, any component that is started, that belongs to that session, automatically goes into debug mode.

Similarly, PeopleSoft Application Engine PeopleCode, Component Interface PeopleCode, and Message Subscription PeopleCode can be debugged.

After you exit debug mode, through the Debug, Exit Debug Mode menu or by exiting PeopleSoft Application Designer, all components in that session go out of debug mode.

If you exit a component, debugging continues with any remaining open and running components.

If more than one PeopleSoft Application Designer session is running, the Application Designer session that is used as a debugger is the first one that had been started.

If you're in debug mode, a PeopleCode Editor window is opened for every item (for example, record, component, page, and so on.) that has PeopleCode in it when that PeopleCode is executed. If a component has more than one event with a PeopleCode program, only one window is opened per item.

For example, if you have a record that has PeopleCode in both the SearchSave and RowInit events, only one PeopleCode Editor window is opened: first it contains the SearchSave PeopleCode program, then the RowInit program. If you have PeopleCode in the RowInit event for two different records that are part of the same component, two PeopleCode Editor windows are opened, one for each RowInit PeopleCode program.

Variables Panes

There are four types of variables panes:

- Local
- Global
- Component
- Parameter

The Local, Global, and Component variable panes show Local, Global, and Component variables, respectively. The Parameter variable pane shows the value of parameters passed in function declarations.

From the variables pane you can check the value of the variables you have in the program. These values are updated as the code runs.

Variables	
Local Name	Local Value
+ &RS0	Rowset
+ &RS1	Rowset
+ &RS2	Rowset
+ &RS1H	Rowset
&MYFIELD	000003
&I	2
&FOUND	000015
©FRMROW	<no value>
©TROW	<no value>

Local Variables pane

In addition, you can "expand" any of the objects to see its properties by clicking on the plus sign next to the variable name. In the following example, a level 1 rowset has been expanded. You can see the properties that are part of the rowset (such as ActiveRowCount or DBRecordName).

Local Variables	
Local Name	Local Value
+ &RS0	Rowset
+ &RS1	Rowset
- RowCount	1
- ActiveRowCount	1
- DeleteEnabled	True
- InsertEnabled	True
- Level	1
- EffDt	
- EffSeq	0.00
+ ParentRowset	
+ ParentRow	
- Name	EMPL_CHECKLIST
- DBRecordName	EMPL_CHECKLIST
- IsEditError	False
- TopRowNumber	1
+ GetRow(...)	
+ &RS2	Rowset
+ &RS1H	Rowset
&MYFIELD	<no value>
&I	<no value>

Local Variables pane with rowset object expanded

In addition, some objects "contain" other objects, like a rowset contains rows, rows contain records or child rowsets, and records contain fields. You can "expand" these secondary

objects also, to see their properties. In the following example, the first row of a rowset has been expanded, as has the EMPL_CHECKLIST record.

Local Variables	
Local Name	Local Value
⊖ &RS1	Rowset
RowCount	1
ActiveRowCount	1
DeleteEnabled	True
InsertEnabled	True
Level	1
EffDt	
EffSeq	0.00
⊖ ParentRowset	
⊖ ParentRow	
Name	EMPL_CHECKLIST
DBRecordName	EMPL_CHECKLIST
IsEditError	False
TopRowNumber	1
⊖ GetRow(...)	
⊖ (1)	
RecordCount	4
ChildCount	1
RowNumber	1
Visible	True
Selected	False
IsChanged	True
IsDeleted	False
IsNew	True
⊖ ParentRowset	
IsEditError	False
Style	
⊖ GetRecord(...)	
⊖ EMPL_CHECKLIST	
IsDeleted	False
IsChanged	True
Name	EMPL_CHECKLIST
FieldCount	5

Variable pane with rowset, row and record expanded (shown with condensed font)

Field Values

When you view a field object in the Debugger, the value of the field is listed under the Value column. Therefore, you do not have to navigate to the Value property to see the value of a field.

The following example shows the PERSONAL_DATA record, and the values of the fields.

Local Variables	
Local Name	
<input type="checkbox"/> PERSONAL_DATA	
<input type="checkbox"/> IsDeleted	False
<input type="checkbox"/> IsChanged	True
<input type="checkbox"/> Name	PERSONAL_DATA
<input type="checkbox"/> FieldCount	83
<input checked="" type="checkbox"/> ParentRow	
<input type="checkbox"/> RelLangRecName	
<input type="checkbox"/> IsEditError	False
<input type="checkbox"/> GetField(...)	
<input checked="" type="checkbox"/> EMPLID	8001
<input checked="" type="checkbox"/> NAME	Schumacher,Simom
<input checked="" type="checkbox"/> NAME_PREFIX	Mr
<input checked="" type="checkbox"/> NAME_SUFFIX	
<input checked="" type="checkbox"/> LAST_NAME_SRCH	SCHUMACHER
<input checked="" type="checkbox"/> FIRST_NAME_SRCH	SIMOM
<input checked="" type="checkbox"/> ADDRESS1	461 Haven Ct
<input checked="" type="checkbox"/> ADDRESS2	
<input checked="" type="checkbox"/> ADDRESS3	
<input checked="" type="checkbox"/> ADDRESS4	
<input checked="" type="checkbox"/> CITY	Moraga

PERSONAL_DATA record field values

In addition, the only fields that are displayed in the debugger are the fields that are actually in the Component Buffer. For example, suppose you have a derived work record, but don't access all the fields in the work record. Only the fields that you access, that are in the Component Buffer, are actually displayed in the debugger.

See Also

"Referencing Data in the Component Buffer", What Record Fields Are in the Component Buffer?

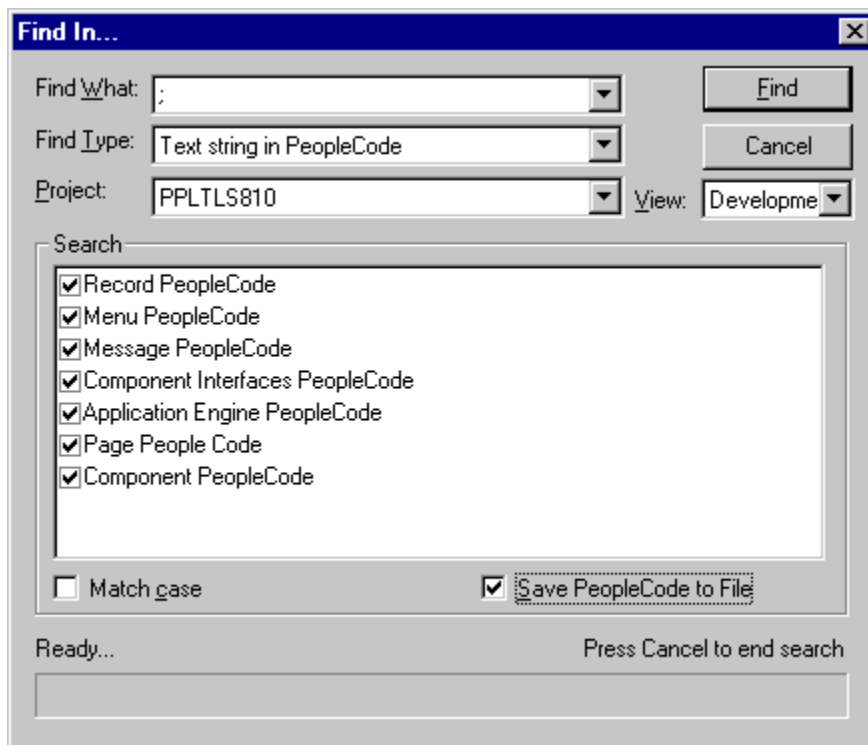
General Debugging Tips

- If you're having problems determining if the correct data is being loaded into the component buffers, use the View Component Buffers view window to see all the values currently in the component buffer. (This is equivalent to putting a **GetLevel0** function at the start of a program.)

Use the &LEVEL0 variable to navigate through all the levels of the rowset object, see the row, records, fields, and so on. This shows you everything that has been loaded into the component buffers for that component.

- While at a breakpoint, if you lose track of the window, or the location within the window, that is displaying the green execution location arrow, you can use the Execution Location Properties menu item's ViewCode button to find your current execution location again.

- Objects remain expanded in the variable windows as you move through PeopleCode. This enables easy and quick inspection of the state of an object as you step through the PeopleCode. However, there is a performance cost for using this feature. If you are finished examining an object, you may want to collapse it to improve the response speed.
- If a database transaction has been started (either for you by PeopleTools, or by you in PeopleCode) other users of that database are blocked from accessing that database until the transaction is complete. If you are stepping through PeopleCode while this transaction is open, you could potentially block other users for an extended period of time. You may want to use a private database for debugging to avoid blocking other users.
- To create a file that contains all the PeopleCode for a project (or database), use the Find In... utility, and search for ";". Be sure to select **Save PeopleCode to File**.



Find In... dialog box for finding all PeopleCode

DoModal Considerations

If you've set the PeopleCode Debugger to Break At Start, and you're using the DoModal PeopleCode function, the DoModal window may display *behind* the PeopleCode Debugger window. This makes it appear as though the Debugger has stopped, when it actually has not. Be sure to check that other windows have not opened while you're debugging the code.

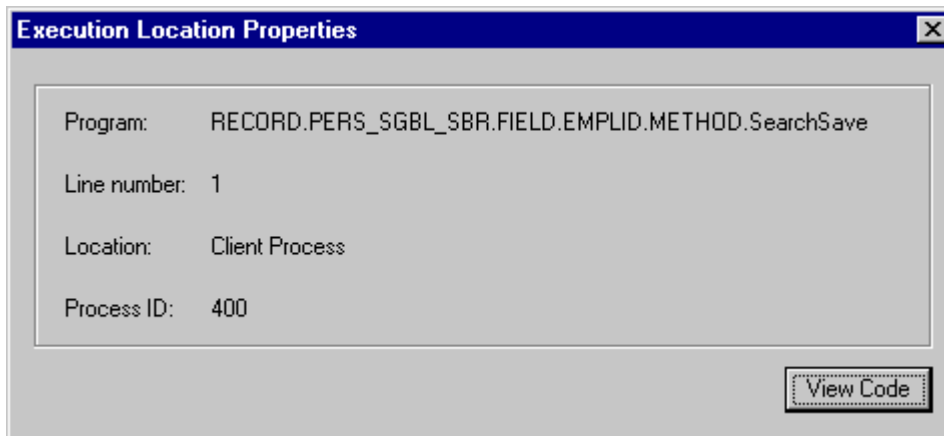
PeopleCode Debugger Options

After the debugger is running, if you select Debug again, you can select other options:



PeopleCode Debugger options

Exit Debug Mode	Exits debug mode. When you exit debug mode, all breakpoints are automatically saved. If you close PeopleSoft Application Designer, you automatically exit debug mode.
Abort Running Program	Stops running the PeopleCode program that is currently running.
Execution Location Properties	Displays the location of the running code in a dialog box. This includes the record name, field name, event name, and line number of the code. It also indicates if the code is executing on the client or server. You can also view the exact code by clicking View Code .



Execution Location Properties dialog box

Break at Start

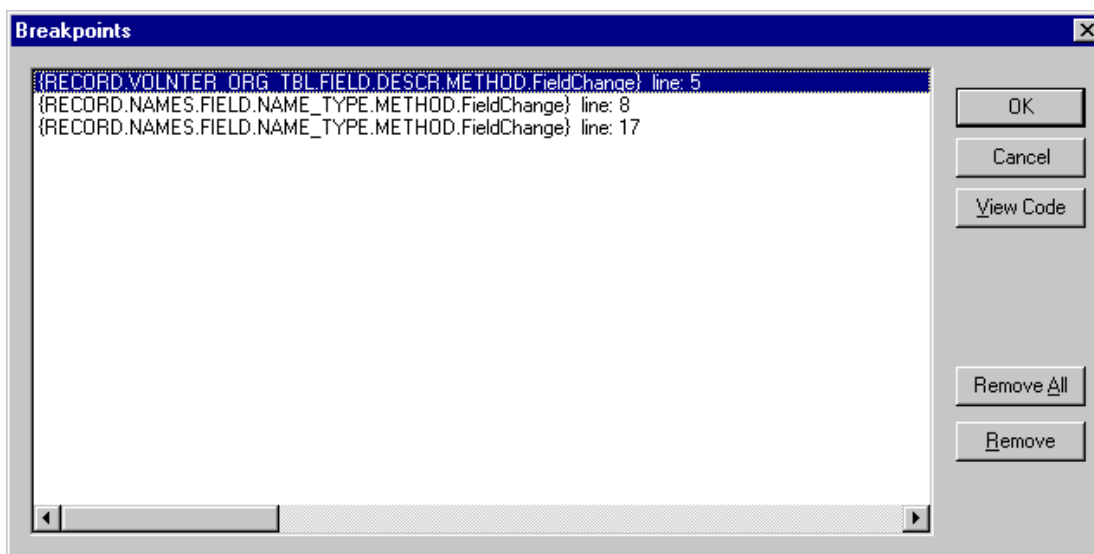
Pauses execution of the component on the first line of every PeopleCode program that executes in the component. If you've started a component with **Break at Start** selected, then start a second component, the PeopleCode associated with the second component will be stopped at the first line of the first PeopleCode program as well, as part of the same debugging session.

Toggle Break at Cursor

Removes the breakpoint if the line the cursor is currently on has a breakpoint. Adds a breakpoint if the line the cursor is currently on does not have a breakpoint.

Edit Breakpoints

Opens a menu that displays the lines that have breakpoints. From this menu you can display the code that contains the breakpoint, by clicking **View Code**. You can also remove one or all of your breakpoints.

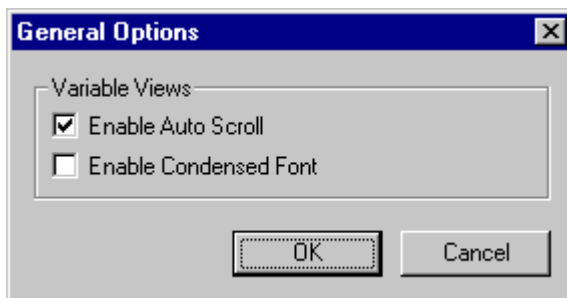


Edit Breakpoints menu

Go	Continues processing until the next breakpoint. If Break At Start is enabled, processing pauses at the next PeopleCode program.
Step	Executes the current line of the PeopleCode program, stepping into functions.
Step Over	Steps through each line of the PeopleCode program, one line at a time, but steps over the functions; the functions are executed, but not stepped into.
Run to Return	Processes past the return of the current function, then pauses.
Step Instruction	Processes low-level pseudo-machine code instructions internal to PeopleCode. This option is used in conjunction with Log Options .
View Global Variables	Opens a separate window for watching variables declared as Global.
View Component Variables	Opens a separate window for watching variables declared as Component.
View Local Variables	Opens a separate window for watching variables declared as Local.
View Function Parameters	Opens a separate window for watching user-specified parameters in Function calls.
View Component Buffers	Opens a separate window for viewing the current component buffers. This is equivalent to getting a level 0 rowset for the component.

Note. These five windows are continuously updated as the program executes.

Options Enables you to select between opening up a dialog box for general options or for specifying log options.



General Options dialog box

The General Options dialog box enables you to specify conditions of the View windows.

Enable Auto Scroll

If you select this check box, and you click a plus symbol next to a variable name in a View window, the variable you clicked “scrolls” to the top of the View window.

Enable Condensed Font

If you select this check box, all View windows are displayed with a smaller font.

The default is for both of these options to be selected.

Additional Features**Break at Termination**

After you are in debug mode, generally, any PeopleCode program in the session that terminates abnormally will first break in the debugger. In addition, the error message will be displayed in the PeopleCode log in the bottom window of PeopleSoft Application Designer.

See Also

Setting PeopleCode Debugger Log Options

Setting Up the Debugging Environment

You can use the PeopleCode debugger for two-tier and three-tier debugging. The database and application can reside on remote servers; they do not need to reside on the local machine.

Two-tier debugging works "out of the box." Setting up three-tier debugging requires you to make a few modifications in PSADMIN (PSAPPSRV.CFG) to enable debugging.

You can connect to an NT server domain that is not on your local machine. You don't have to configure a local domain to do this. You also don't have to have PeopleTools software installed locally for three-tier debugging.

See Also

PeopleTools PeopleBook: PeopleSoft Internet Architecture Administration, "Administration Considerations," Setting up the PeopleCode Debugger

Debugging Subscription PeopleCode

Use the following steps to debug Subscription PeopleCode.

To debug Subscription PeopleCode:

1. Start PeopleSoft Application Designer on the server machine that will be processing the subscription.

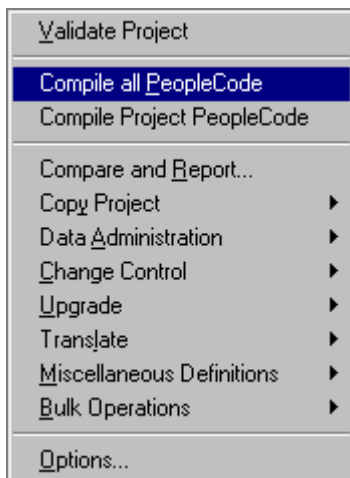
You must use the same User ID and database as the machine to which its Application Servers have been configured.

2. Start debug mode.

When the subscription server executes the subscription PeopleCode, the Debugger starts.

Compiling All PeopleCode Programs

In addition to checking individual programs, you can compile all PeopleCode programs either in a database or in a project to check for errors. This option opens and compiles every PeopleCode program. This option can be run after an upgrade to verify that all the programs were upgraded correctly. It could also be run on an as-needed basis to check for corruption in your programs.



Tools menu - Compile All PeopleCode options

To compile all PeopleCode programs:

1. Open PeopleSoft Application Designer while accessed to the database that contains the PeopleCode you want to check.
2. Select the compile option you want to use.

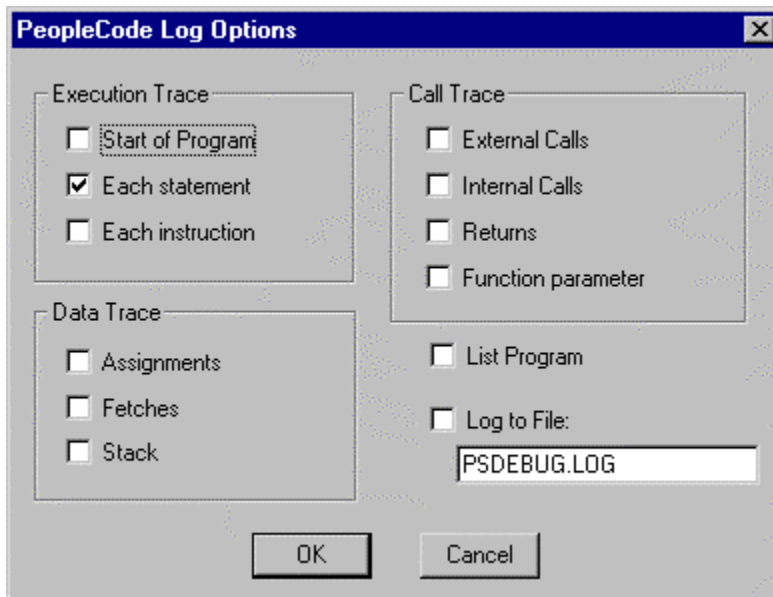
Select Tools, Compile All PeopleCode or Tools, Compile Project PeopleCode.

3. Click **Compile** on the Compile All PeopleCode dialog box.

If you have any errors, they appear in the PeopleCode log display window.

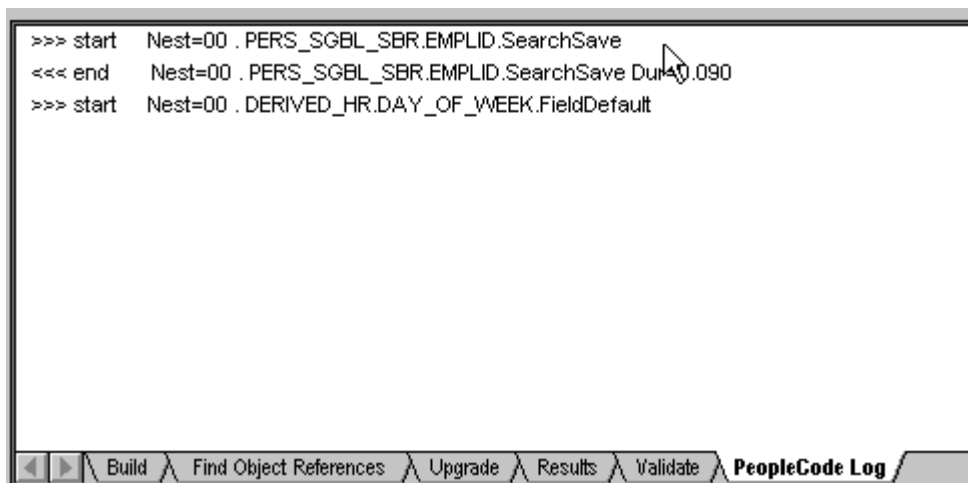
Note. If you've specified a log file in the debugger log options, all errors are written to the log file as well.

Setting PeopleCode Debugger Log Options



PeopleCode Log Options dialog box

Use the PeopleCode debugger to view PeopleCode that is executed while you're stepping through your application (use Debug, Log Options.) All log information appears in the PeopleCode Log window, at the bottom of PeopleSoft Application Designer.



PeopleCode Log window

You can also record what you see in a log file. You can tailor the log results to record a variety of online information—from as little to as much data as you require.

If you exit debug mode, but do *not* close PeopleSoft Application Designer, all the log options you specified are still there when you start debug mode again.

When you close the PeopleSoft Application Designer, all log options are clear. The next time you enter debugging mode, you must reselect your debug log options.

Some of the log options are described in the next few sections.

See Interpreting the PeopleCode Debugger Log File for more details.

All the options available in the Log Options dialog box are also available in PeopleSoft Configuration Manager, on the Trace tab, in the PeopleCode Trace section.

See Also

PeopleTools PeopleBook: Data Management, "Understanding Configuration Manager," Trace

Execution Trace Options

Execution Trace is set to trace each PeopleCode statement. You can also trace the start of each program or each program instruction.

Data Trace Options

The Data Trace options are:

Assignments	Records each assignment made to a field.
Fetches	Records the field values retrieved from a PeopleCode fetch.
Stack	Indicates the contents of the internal machine stack. Typically, only PeopleSoft staff developing PeopleCode language enhancements use this option.

Call Trace Options

The Call Trace options enable you to record the values of External calls, Internal calls, Returns, and Function parameters.

External Calls	Traces each call to external (PeopleCode) functions.
Internal Calls	Records each call to internal subroutines.
Returns	Logs the occurrence of program returns.
Function parameters	Logs the value of individual PeopleCode function parameters.

Log To File

When you select this option you must specify the name of a file: you receive an error and logging to file is disabled if you do not specify the name of a file.

If you do not specify a directory location, the file is placed in the same directory from which you're running PeopleTools.

If you specify the name of an existing file, a warning message appears, asking you whether to overwrite the file. At this point, you must go back into the Log Options and specify a different file name. If you do *not* specify a new log file name and start running an application, the log file is overwritten.

If you run more than one application and do *not* exit PeopleSoft Application Designer between times, each trace is appended to the specified log file.

Interpreting the PeopleCode Debugger Log File

You can produce a trace log using any of the following methods:

- Log File option in the PeopleCode Debugger
- PeopleSoft Configuration Manager Trace tab
- Built-in functions SetTracePC and SetTraceSQL
- PeopleTools Utilities (this is included for backward compatibility purposes only, and in general, should not be used.)

The first option, using the Log File, produces essentially the same trace file as using any of the other options. The only difference is that the other trace files contain timing information, that is, when each line started processing, and how long it took to execute.

The log file produced by the latter options is specified by the PeopleTools Trace File option on PeopleSoft Configuration Manager. All of these options write to the *same file*. The Log File option writes to the file you specify, that is, a different file.

Trace files are also produced by PeopleSoft Application Engine. These logs may contain more information.

In this section, we discuss the following topics:

- Log file contents.
- Sample trace file.
- Operations and Operands.
- Printed data values.
- Other items in the log file.

See Also

PeopleTools PeopleBook: Data Management, "PeopleTools Utilities," Debug Utilities

PeopleTools PeopleBook: Peoplesoft Application Engine, "Tracing Application Engine Programs"

Log File Contents

Rather than having to examine large program dumps, use the log file as it contains useful information for debugging PeopleCode.

You can view the log using any editor that displays ASCII text, such as Notepad. It has the following components.

Line Count	Specifies a line number in the file.
Internal Information	Contains reference numbers used for internal tracing. You can ignore this information.
Instruction Location	Address of an instruction processed in the program. You can follow programs and functions using this number.
Operation Code	The operation performed by the program.
Operation Operands	Contains information specific to each operation. The following table lists the possible operations and the operands that appear for the list and trace options.

Other Items in the Log File

There are several other items that can appear in a debugging trace. The following table describes those items.

<i>Trace Item</i>	<i>Description</i>
Store Field:record name.field name Value=xx	Issued when the assignments trace option is selected. It contains the record and field names and the value that is stored.
Fetch Field:recordname.fieldname Value=xx	Issued when the Fetch Fields option is selected. It contains the record and field name and the value that is retrieved.
Fetch Field:recordname.fieldname Contains Null Value	Issued when the Fetch Field option is selected and the selected record.field contains a null value.
Fetch Field:recordname.fieldname Does Not Exist	Issued when the Fetch Fields option is selected and when the field is not found.
Branch Taken	Displayed after a branch test when the branch is taken.
Field Not Found, Statement Skipped	Displayed whenever a "referenced field was not found" error causes the PeopleCode processor to skip to the next statement.
vvvvvv PeopleCode Program Listing	Issued when the List Program option is selected. It marks the beginning of a PeopleCode program listing.

<i>Trace Item</i>	<i>Description</i>
^^^^^ PeopleCode Program Listing End	Issued when the List Program option is selected. It marks the end of a PeopleCode program listing.
Error Return -> NNN	Issued when a fatal error condition terminates the PeopleCode program.

Find In . . .

Use the Find In feature of PeopleSoft Application Designer to search for strings, either in PeopleCode programs or in SQL definitions. This feature searches:

- All PeopleCode programs and all SQL statements
- Only PeopleCode programs
- Only SQL statements
- Only HTML definitions

You can further refine your search by specific project. If you're searching PeopleCode programs and SQL statements, you can further refine your search by item searched, that is, you can specify if you want record PeopleCode, page PeopleCode, menu PeopleCode, and so on.

All output from the search is placed in an output window. You can save these results to a file, copy them, clear them, or print them.

From the output window, you can immediately open any of the PeopleCode programs, SQL statements listed, or HTML.

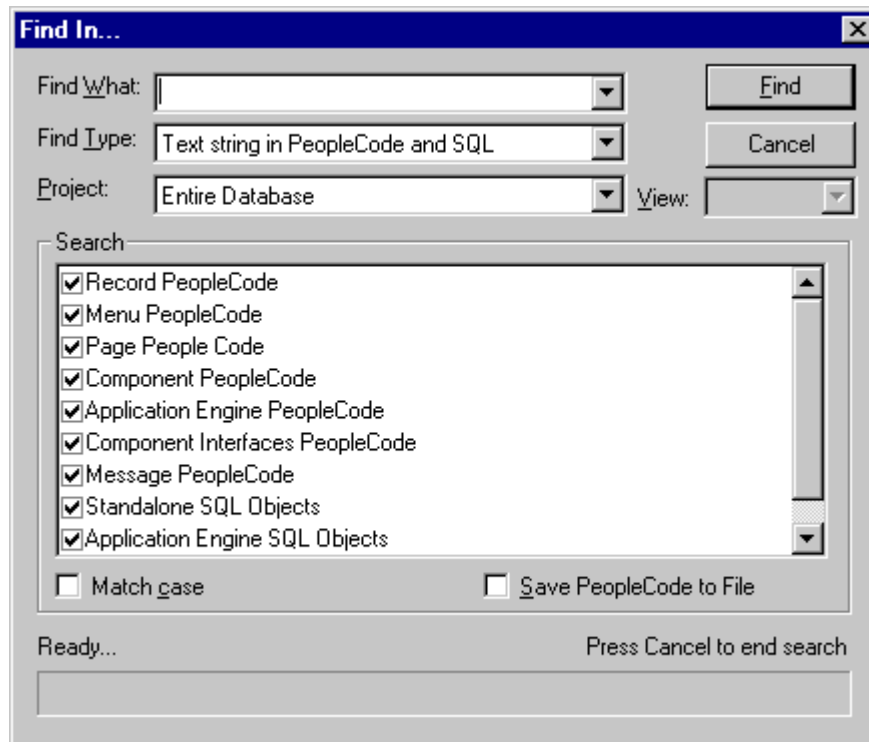
Also, from the output window, you can insert selected records into a project. Then, if you need to search those records again, you can search by project.

Note. If you want to create a file that contains all the PeopleCode for a project (or database) you can use the Find In. . .feature, and search for ";". Be sure to select **Save PeopleCode to File**.

To find a text string in PeopleCode or SQL:

1. In PeopleSoft Application Designer, select Edit, Find in.

This displays the Find in dialog box.



Find In dialog box

Use this dialog box to specify a string to be searched for.

2. Type the search string you want to find in the **Find What** edit box
 If you want returned only those items that match the case of what you entered, select the **Match Case** check box at the bottom of the dialog box.
3. Specify with the **Find Type** edit box whether you are searching in PeopleCode and SQL, just PeopleCode, just SQL, or HTML.
4. Select the project you want to search.
 You can search the entire data base, or any existing project.
5. (Optional) Select the view you want to search.
 If you decide to not search the entire database, you can specify if you want to search the Development view or the Upgrade view. The default is the Development view.
6. Select the items you want to search.
 You can search all items that contain either PeopleCode or SQL, or just a subset of items.

Note. The **Search** check box list for HTML is empty. The search is conducted against all HTML definitions.

7. (Optional) Save the search results to a file.

You can save the results of a PeopleCode search to a text file, which you can view or print using a text editor or word processor. The text file contains the entire PeopleCode program that contained the string.

If you want to save your results to a file, select the Save PeopleCode to File check box at the bottom of the dialog box. The results are saved to the file, and display in the PeopleSoft Application Designer Find In . . . output window.

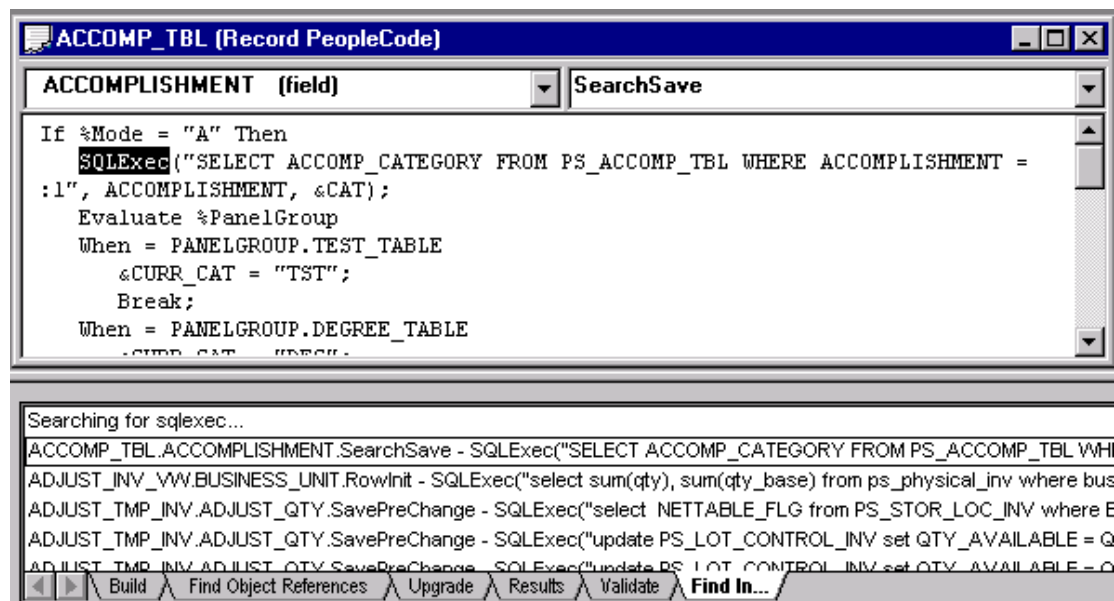
8. Click the **Find** button to start the search.

As Find In . . . searches the database, it displays a counter at the bottom of the Find In dialog box indicating the number of PeopleCode programs searched.

You can click the **Cancel** button to stop the process.

9. Check the Find in Output window for results.

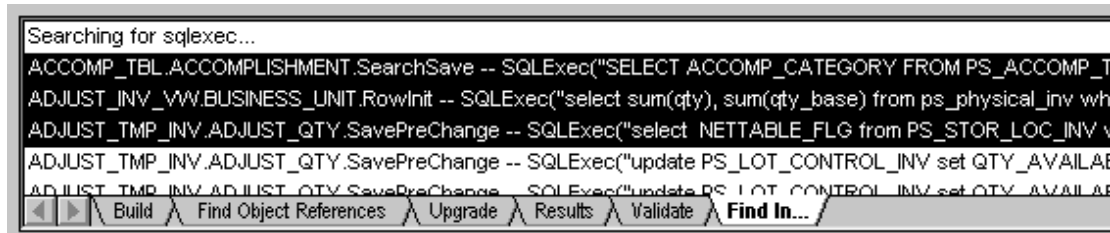
The results of the search appear in the Find In . . . tab of the output window. Each line shows where the string was found. You can open any of the programs listed by double-clicking a line in the output window.



Opening a PeopleCode program from the Find In tab

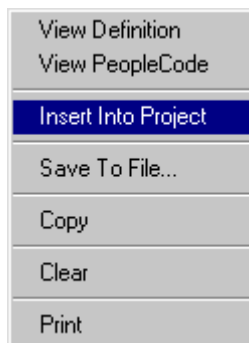
To save records in a project:

1. Run Find In to search for a string.
2. Highlight the references you want saved in the output window using click-shift.



Find In output window with records selected

3. Right-click the highlighted records and select Insert Into Project.



Find in pop-up menu

All the selected records are inserted into the current open project. Save your project.

The next time you search, you can search only your project (select a Project on the Find In dialog box) instead of searching the entire database.

Cross Reference Reports

You may find a situation where a field value changes, and you do not know how it changed. There are two ways of finding all references to a field.

- Find References in PeopleSoft Application Designer
- Cross reference reports

See Using Application Designer Projects for details about Find References.

PeopleTools is delivered with three PeopleCode cross reference reports:

- XRFFLPC.

Reports on all fields in the system referenced by other PeopleCode programs. The report sorts by record names, then field names. XRFFLPC shows the records, fields, and PeopleCode program types that reference each field.

- XRFLPCFL.

Reports on the fields that each program references. It sorts the report by record definition, field name, then PeopleCode type. It shows the records and fields referenced for each program. This report and XRFFLPC complement each other by using converse approaches to reporting the cross references.

- **XRFPNPC.**

Reports on pages with PeopleCode. This report show pages containing fields with PeopleCode attached to them.

You can run these using PeopleSoft Query and either view the reports online or print them. You can also download them an Excel spreadsheet.

	Panel Name	Record	Field Name	Prog Type
1	ACL_APPDES_MISC	DERIVED_APPDESM	ACCESS_CD	FieldChange
2	ACL_APPDES_MISC	DERIVED_APPDESM	ACCESS_CD	RowInit
3	ACL_APPDES_MISC	DERIVED_APPDESM	FULL_ACCESS_BTN	FieldChange
4	ACL_APPDES_MISC	DERIVED_APPDESM	ITEM_CHANGED	SavePreChange
5	ACL_APPDES_MISC	DERIVED_APPDESM	NO_ACCESS_BTN	FieldChange
6	ACL_APPDES_MISC	DERIVED_APPDESM	READ_ACCESS_BTN	FieldChange
7	ACL_APPDES_OBJECTS	DERIVED_APPDESO	ACCESS_CD	FieldChange
8	ACL_APPDES_OBJECTS	DERIVED_APPDESO	ACCESS_CD	RowInit
9	ACL_APPDES_OBJECTS	DERIVED_APPDESO	FULL_ACCESS_BTN	FieldChange

XRFPNPC Query results

See Also

PeopleTools PeopleBook: PeopleSoft Query, "Introduction to PeopleSoft Query"

PeopleTools PeopleBook: Peoplesoft Application Designer, "PeopleSoft PeopleTools Cross-Reference Reports"

Glossary

The terms in this glossary are used among multiple Financials and Supply Chain Management applications.

Numbers

401(a)(17) Limits

The limitations on the earnings that may be included in the calculation of benefits under qualified U.S. pension plans.

1st Year Amount

In PeopleSoft Workforce Analytics, 1st Year Amount is an employee-level compensation amount, totaling the calculations for the first calendar year's worth of accounting periods, in a compensation scenario.

A

Abend

Abnormal End (to a process).

ABM (Activity-Based Management)

See PeopleSoft Activity-Based Management.

ABPS (Activity-Based Planning and Simulation)

See Activity-Based Planning and Simulation.

Absence

An absence occurs when an employee is not at work (absent) during a normally scheduled work period. Absences may be scheduled or non-scheduled, compensated or uncompensated, excused or unexcused. An absence may occur for a variety of reasons like illness, family emergency, civic obligations (e.g. Military duty or jury duty), or vacation.

Absence Entitlement

Element which defines the 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

Element which defines the conditions that must be met before a payee is entitled to take paid time off.

Accepted Exception

An exception that has been reviewed and validated (see Time Management).

Accommodations

Accommodations are efforts your organization is able to make for employees or applicants with disabilities, such as purchasing special equipment or making structural changes to a work environment.

Account Management

In PeopleSoft Demand Planning, a feature that enables you to divide a centrally held corporate forecast into multiple subsections for easier maintenance and management. These subsections are separate databases that can be distributed to account managers for use and updates, then rejoined with the main database at a later date.

Account

A code for recording and summarizing financial transactions as expenditures, revenues, assets, or liabilities balances. This is a delivered PeopleSoft ChartField, specific use of which is typically defined by the organization during implementation of PeopleSoft General Ledger.

Account Type

A name for one of the different kinds of accounts used in a PeopleSoft General Ledger, such as Asset, Liability, Equity, Revenue, and Expense.

Accounting Class

In PeopleSoft Enterprise Performance Management, an attribute that defines how the particular resource would be treated for generally accepted accounting practices. Inventory denotes whether a Resource will become part of a balance sheet account such as inventory or fixed assets, while Non-inventory denotes that the Resource will be treated as an expense of the period during which it occurs.

Accounting Date

The date that a transaction is recognized as opposed to the date the transaction actually occurred—the **Transaction Date** (although the two dates 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. In PeopleSoft Asset Management, the difference between accounting date and transaction date determines whether prior period depreciation must be calculated, and how much. Accounting Date must be later than or equal to Transaction 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 = credits). Accounting entries are created to record accruals, payments, payment cancellations, manual closures, project activities in general ledger, and so forth (depending on the application).

Accounting Entry Template

A user-defined table that controls the use of system-generated accounting lines in the posting processes.

Accounting Split

Method indicating how expenses are allocated or divided among one or more sets of accounting ChartFields.

Accredited Education

Education above the high school level completed in a U.S. college, university, or other educational institution that has been credited by one of the accrediting agencies or associations recognized by the Secretary, U.S. Department of Education.

Accrual

Any hours that employees accumulate for use at another time in the form of earned vacation time or sick leave, for example.

Accrual Basis Accounting

Accounting that records the impact of a business event as it occurs, regardless of whether the transaction affected cash.

Accrual Class Codes

Classes or categories of accruals.

Accrual Type

Defines an accrual such as annual leave or sick leave.

Accumulate Demand

In PeopleSoft Demand Planning, a transfer process function that adds demand quantities for an item to any quantities that already exist for the period.

Accumulator

Element which allows you to combine several elements. 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. See also Time Administration.

Accumulator [Global Payroll]

Element which provides a means for storing the cumulative values of defined items as they are processed. As you make payments, take deductions, and perform calculations, you'll use accumulators to track accumulated amounts, or balances. You can accumulate a single value over time or multiple values over time, as your requirements specify. 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

In PeopleSoft Deduction Management, a task that you perform to obtain information required to resolve a deduction.

Action and Conditions

A process that defines actions and conditions independently of one another and then combines them to create a complete rule (see Rule Creation).

Action Code

In PeopleSoft Engineering, a user-defined code associated with an event/action triggered by the implementation of an engineering change order (ECO). Actions could include analyzing an item's existing quantity on hand, scrapping existing inventory, or modifying current documentation.

In PeopleSoft Product Configurator, a 2-character code that identifies rule types. For example, **FP** is the action code for the Finalize Price rule, and **CN** is the action code for the Condition rule. The rules control the processing path for configured items.

Action List

An online list of customers who meet predefined credit management criteria. The list also includes appropriate procedures for each action and contact information for the customer.

Action Owner

In PeopleSoft Deduction Management, the individual assigned a task to obtain information to resolve a deduction.

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 paygroup to another—and a ***reason*** for that action. Action Reason is used by PeopleSoft Human Resources, PeopleSoft Benefits Administration, PeopleSoft Stock Administration, and the COBRA Administration feature of the Base Benefits business process.

Active Control

A target control requiring that the user validate the budget against the planning targets before submitting it. If the budget totals are not within the tolerance levels, the system indicates that the status is invalid and the user cannot submit their budget until the budget is modified and the amount is within the tolerance range of the planning target.

Activity

In PeopleSoft Receivables and Deduction Management, an action taken on an item, such as creating an item, unposting an item, or writing off an item.

In PeopleSoft Projects, the unit of work that provides a further breakdown of projects—usually into specific tasks. Resources are assigned directly to activities within a project, not directly to projects.

A self-contained task that is part of one or more business processes. Business process maps display the activities that make up the process. An activity consists of steps representing the pages the user needs to complete and events representing the workflow routings triggered by the user's actions.

In PeopleSoft Enterprise Warehouse, the work of an organization and the aggregation of actions used for Activity-Based Costing.

Activity Attributes

Activity Attributes provide pieces of activity information. For example: capacity and performance, cost drivers, cycle time and performance measures.

Activity-Based Costing (ABC)

A methodology that measures the cost and performance of activities, resources and cost objects, assigns resources to activities and activities to cost objects based on their use and recognizes the causal relationships of cost drivers to activities.

Activity-Based Management (ABM)

See PeopleSoft Activity-Based Management (ABM).

Activity-Based Planning and Simulation (ABPS)

ABPS, a feature of PeopleSoft Activity-Based Management, calculates resource demands, new rates, costs, and activity volumes based on demand forecasts. It converts the new

resource demands into new cost requirements at the General Ledger item level to feed as input for budgeting.

Activity Driver

An Activity Driver indicates the amount of demand there is for a particular activity and it is used to assign cost to cost objects. In some instances, an activity driver may represent the yield of an activity.

Activity Fragmentation

The part of the Employee Profile feature that provides information about the number of employees that is involved in completing a particular activity on a full or part-time basis.

Activity ID

A unique 15-character alphanumeric identifier given to each activity within a project. Activity IDs need only be unique within a single project.

Activity List

In PeopleSoft Pension Administration, a checklist used to monitor pension-related activities.

Activity Type

A user-definable identifier for grouping activities.

Activity Type

Also known as Activity Code. A categorization of work effort. Typically work effort is categorized as productive or non-productive; Repair, Maintenance, Enhancement, or Improvement; or Development or Construction. Activity type is usually required to support cost accounting or financial accounting (recording) functions. It may also be required to support some organizational administration requirements such as organizational productivity goals, or employee performance measurement. In some companies, activity type is inferred from job function, work group affiliation, or organization.

Activity Use

An attribute used to describe the behavior of an Activity as defined within PeopleSoft Enterprise Performance Management. A Primary Activity is an activity that is performed for the purpose of directly generating revenue within the course of business. A Secondary Activity is generally performed in direct support of a Primary Activity such as activities related to human resources or MIS.

Actual Base Hours

This defines the number of hours that an employee is expected to work within a given period under analysis within PeopleSoft Enterprise Performance Management. Hours worked in excess of Actual Base Hours are generally considered overtime, while hours worked less than Actual Base Hours would illustrate that the employee is working part-time.

Actual Contribution Percentage (ACP)

The amount of an employee's after-tax or employer matching contributions made in a Section 401(m) plan on behalf of highly compensated plan participants, divided by the employee's annual compensation, or an amount determined in the same manner with respect to non-highly compensated employees. The Base Benefits business process is set up to perform ACP nondiscrimination tests for Section 401(m) plans. See Nondiscrimination Tests and Highly Compensated Employee.

Actual Date

Calendar date in which a punch occurred (see Time Reporting).

Actual Deferral Percentage (ADP)

The amount of salary reduction contributions made by an employee to a Section 401(k) plan for a year, divided by the employee's total compensation for that year. The Base Benefits business process is set up to perform ADP nondiscrimination tests for Section 401(k) plans. See Nondiscrimination Tests and Highly Compensated Employee.

Actual Demand

In PeopleSoft Demand Planning, an **Array** of demand by historical period imported from an external system. The demand figures are determined by imported values and typically include shipments, orders booked, orders booked by requested ship date, or shipments.

Actual Rates

An Actual Rate is the rate that your business currently uses for its business practice.

Actuarial Assumptions

Any assumptions used to calculate an equivalent benefit for an optional form of payment or an alternative retirement date.

Actuarial Valuation

A comparison of a pension plan's assets and liabilities.

Actuarial Valuation Extract

A PeopleSoft Pension Administration data extract containing data that a plan actuary needs in order to determine the plan's assets and liabilities.

Address Type

A high-level address classification that identifies addresses associated with a **Material Issue**. Examples include Ship To Address, Bill To Address, and Ship Notification Address.

Adjusted

In the Enterprise Planning and Simulation forecasting process, in addition to versions of the statistical forecast, there is an adjusted version of the forecast. Managers create this version by reviewing the forecasts and entering adjustments that cannot be inferred statistically. For example, there may be a promotional campaign next quarter that is expected to boost volume for certain products over several weeks.

Adjusted Demand

In PeopleSoft Demand Planning, an **Array** of demand after adjustments have been made to the actual demand values. The adjusted figures may include both manual and system-generated changes, such as demand filtering and depromotion. The system uses adjusted demand rather than actual demand in the Forecasting Reset process and in the recalculation of model components during period-end processing.

Adjusted Forecast

In PeopleSoft Demand Planning, a **Statistical Forecast** that has been adjusted using management overrides, proration, or summarization.

Adjustment

See **Bill Adjustment** or **Inventory Adjustment**.

Adjustment Voucher

A PeopleSoft Payables voucher that enables you to apply an adjustment to an existing voucher or to relate one voucher to another.

Advice

The Form that employees who choose direct payroll deposit receive in lieu of a check.

Affiliate

A control person of a corporation. Generally, an officer, director, or major shareholder that has the ability to influence the corporate management decisions.

After-tax Deductions

Deductions that reduce net pay. These deductions are subtracted from gross pay after taxes have been taken out. Also called “post-tax” deductions.

Agency

Any Department or independent establishment of the Federal Government, including a government-owned or -controlled corporation, that has the authority to hire employees in the competitive, excepted, and senior executive services.

Aggregated

In Enterprise Planning and Simulation, each period the statistical forecast is calculated automatically by the system. A forecast for each individual product can be computed using history for that product. Then these forecasts can be aggregated (that is, summarized) into forecasts for the product family.

Aggregate Reporting

The ability to report time as a collection or mass. In Time and Labor aggregate time reporting features include the ability to report time in a lump sum, as a pattern, in a range of dates, or for an entire crew.

Aging Data

Updating data from separate sources, and separate dates, to a common date using an annualized factor.

Aging ID

A code representing rules for aging open items.

Alias

Any of several PeopleSoft Pension Administration utilities that look up or calculate employee information.

Allocated

In Enterprise Planning and Simulation, the computed forecast and the summarized forecast are two different versions of the statistical forecast. In addition, the forecast at the product family level can be allocated down to the individual products. Usually this allocation is done in proportion to the calculated product forecasts at that level. This version of the (statistical) forecast is called the allocated or prorated statistical forecast.

Allocated Inventory

The inventory assigned to a specific stock request.

Allocation Manager

Perform allocations using the Allocation Manager. Allocations enable you to distribute revenue, expense, and statistical quantities across business units, departments, and so on. You can allocate budget planning to detail levels so that you may perform detailed budgeting. The type of allocation you select determines the output.

Allocation Manager Rules

In the PeopleSoft Enterprise Warehouse, Allocation Manager rules allow you to specify the basis as well as the target tables for moving, aggregating, or multidimensionalizing your output. Rules use Allocation Manager methods to enrich the PeopleSoft Enterprise Warehouse data. *See* Allocation Manager Methods.

Allocation Manager Methods

There are several methods: Arithmetic Operation, Prorata, and Spread Even. Each method enables you to move and/or enrich output.

Allocations

A process of distributing budget amounts to and from other Budget Centers. Budget amounts are allocated to cover, or offset, the costs in one Budget Center by charging them to another Budget Center. An allocation is also the budget amount that is distributed to or from a Budget Center. A budget amount that is charged to another Budget Center appears as a negative amount. This same budget amount appears as a positive amount in the other Budget Center receiving the allocation. PeopleSoft Budgeting-specific.

Allotment

This is a voluntary deduction from pay. Employees may elect up to two allotments from pay, transmitted to a financial institution to the employee's checking or savings account.

ALM (Asset Liability Management)

See PeopleSoft Asset Liability Management.

Allowances

The amount owed to an employee in addition to base salary and which is not defined as part of gross salary. For example, vacation can be considered an allowance. PeopleSoft Budgeting-specific.

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.

Alternate BOM

Identifies the multiple ways in which an item can be produced. The primary production BOM is designated as BOM code 1. By using BOM codes, you can associate up to 98 other alternate BOMs with the item.

Alternate Routing

A routing, usually less preferred than the primary routing, but resulting in an identical item. You can specify up to 98 alternate routings for production routing types by entering additional Routing Codes (greater than 1) for the same routing type.

Alternative Minimum Tax (AMT)

AMT is calculated by adjusting the taxpayer's regular taxable income with a number of tax preference items and adjustments. Tax preference items are positive items increasing

Alternative Minimum Taxable Income (AMTI) and are excluded from regular taxable income. Tax preference items include gain from the exercise of incentive stock options.

Amount Type

In PeopleSoft Workforce Analytics, the Amount Type specifies whether a benefits compensation amount is a value or expense, to the employee or the employer.

Analysis Base

Defined static, historical data used both to seed and compare against proposed budgets.

Analysis Group

A grouping of analysis types. Analysis groups can be used for project analysis and grouping or for mapping analysis types.

Analysis Template

A set of pre-defined reports that you can view and publish online. These templates access data in the Enterprise Warehouse tables, and organize it by function, role and industry. The templates allow you to pivot, sort, rank, drill and chart the data, for your analysis needs.

Analysis Type

A 3-character, user-definable identifier that enables you to label the different types of costs. For example, you might want to track budgeted costs (BUD), committed costs (COM), and actual costs (ACT).

Analytical Applications

See PeopleSoft Analytic Applications.

Analytic Forecasting

Analytic Forecasting is the part of the Planning and Simulation feature that creates forecasts for your business requirements.

Annual Amount

In PeopleSoft Workforce Analytics, Annual Amount is an employee-level compensation amount, totaling the calculations for a full fiscal year's worth of accounting periods, in a compensation scenario.

Annual Declaration Report

The French Annual Declaration report is a payroll report which checks establishment profiles to see whether an establishment has to produce the report, and then calculates the amount of all the social security contributions for this establishment.

Annual Leave

Annual leave is absence from work with pay and must be approved by the employee's supervisor in advance. This type of leave (Plan Type 51) is accrued based on years of service: Full-time Permanent/Full-time Seasonal employees ...0-3 years - 4 hours per biweekly pay period; 3-15 years - 6 hours per biweekly pay period (plus an additional 4 hours in the final pay period of the leave year); and 15+ years - 8 hours per biweekly pay period. Part-time Permanent/Part-time Seasonal employees...0-3 years - 1 hour for every 20 hours worked; 3-15 years - 1 hour for every 13 hours worked; 15+ years - 1 hour for every 10 hours worked. Generally, there is a leave year ceiling of 240 hours on accrual; amounts accrued in excess of the ceiling and not used prior to leave year-end are forfeited.

Annual Shareholders Meeting

A meeting of corporation's directors, officers, and shareholders held for the purpose of communicating the operating and financial results for the prior year, the prospects for the future and major decisions of management.

Annual Workforce Survey by Nationality and Professional Category (Enquête sur l'activité et les conditions d'emploi de la main d'oeuvre)

In France, companies are required to submit the Annual Workforce Survey by Nationality and Professional Category to the Ministry of Labor. This report provides an analysis of the company's foreign workforce, which includes any employee who does not have French citizenship.

Annualized Tax Method

A payroll tax calculation method that divides the tax on an annualized amount by the number of pay periods in the year to find withholding for a given pay period, based on the number of withholding allowances. Annualized is the most common tax method.

Annuitant Amount

The gross monthly annuity a federally retired employee receives.

Annuitant CSA Number

A unique number assigned by OPM for a retired employee.

Annuitant Indicator

A code used to indicate the status of an annuitant appointed to a position in the Federal civilian service. Text for the codes is as follows:

1. Reemployed annuitant - Civil Service/FERS
2. Retired military officer receiving pay
3. Retired military non-officer (enlisted) receiving pay
4. Retired military officer receiving pay and a reemployed annuitant - Civil Service

5. Retired military non-officer (enlisted) receiving pay and a reemployed annuitant - Civil Service
6. Not applicable (none of the above)

Annuitant Indicator (cont)

- A. Reemployed Annuitant – FERS
- B. Former Annuitant - FERS
- C. Retired Officer/Reemployed Annuitant - FERS
- D. Retired Officer/Former Annuitant - FERS
- E. Retired Enlisted/Reemployed Annuitant - FERS
- F. Retired Enlisted/Former Annuitant - FERS

Annuity

A series of periodic payments made to an individual. Under a pension plan, these payments are generally made monthly.

Anti-Dilutive

Typically, options or shares where the price is greater than the current fair market value of the security.

APE (Activité Principale Exercée) Codes

APE codes classify the type of industry or activity your French company is in, such as software, banking or insurance. The APE codes are a normalized set of codes that are required by law and are used in regulatory reporting.

API

An Application Programming Interface (API) is the technology that a software product supplies so you can control it or communicate with it from another application. PeopleSoft APIs enable the user to perform desired actions upon PeopleSoft data without having to know the internal logic or rules of the program.

Applicant Hire Process

The procedure of hiring an applicant who has been tracked and administered in the Recruitment pages. Once you assign an Employee ID, the system uses recruitment data to populate the fields in the Personal Data pages.

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.

Application Designer

The integrated development environment used to develop PeopleSoft applications.

Application Engine

PeopleTools batch processes consisting of a set of defined SQL statements. Application Engine processes is more efficient than COBOL or SQR, since they operate within the database system, and don't rely on external processing.

Application Journal Template

A set of rules and default values to control the creation of journals from accounting entries.

Application Processor

The Application Processor is the PeopleTools runtime engine that controls processing of the application from the time the user requests a panel group from an application menu through the time that the database is updated and processing of the panel group is complete.

Application Server

The application server is the centerpiece of PeopleSoft's three-tier architecture. It utilizes Tuxedo, BEA Systems' transaction monitor, to manage client transactions and provide the business rules and workflow capabilities of PeopleSoft's enterprise applications.

Application Server Domain

The collection of server processes and associated resource managers defined by a single PSTUXCFG configuration file. Each application server domain is configured to connect to a single database. Multiple application server domains can exist on the same server machine.

Appointing Authority

The basis that authorized the appointing officer to effect personnel actions on an employee.

Appointing Officer

Denotes if the employee has appointment authority based on laws and regulations.

Approve Time

The Time and Labor feature that approves all employee daily time before it can be sent to payroll for processing. You can approve time by group or by individual employee. You can also unapprove previously approved time.

Approving Official

Individual with the delegated authority responsible for signing the action(s) taken on an employee.

Array

An ordered grouping of data by period and year. PeopleSoft Demand Planning uses arrays in forecasting demand.

Array

Element which enables you to extract information based on a column value. One way of thinking of an array is that it is a SQL statement that retrieves data from an existing table.

Array Dimension

Determines which inventory-stocking possibilities are included in a **Cube View**. This standard one-level dimension consists of the key fields that include, for example, order quantity, safety stock, and turn rate.

Arrears Balance

An amount owed to either the employer or employee, usually the result of a deduction not fully taken.

Ask Price

The price at which someone who owns a security offers to sell it; also known as the asked price.

As-of-Dated

Refers to a snapshot of the data at a given point in time.

Asset Assignment

A streamlined means of associating project costs to assets or asset profiles within PeopleSoft Projects.

Asset Budgeting

Budget for planned asset acquisitions and the associated depreciation expense that can be associated with a Capital Acquisition Plan (CAP).

Asset Catalog

A list of asset profiles which includes information about that asset type, including Cost, Life, Salvage Value, Depreciation Method, Currency Code, and Asset and Depreciation Account.

Asset Category

A standard group of assets. Typical asset categories include Furniture and Fixtures, Machinery and Equipment, Land, Buildings, Leasehold Improvements, and the like. These generally correspond to General Ledger asset accounts. Assets in one category usually share some depreciation characteristics, such as estimated service life and depreciation limits.

Asset Class

An asset group used for reporting purposes. It can be used in conjunction with Category to refine asset classification.

Asset Liability Management

See PeopleSoft Asset Liability Management.

Asset Life

The number of years an asset will depreciate, after which time it might be kept or sold for its Salvage Value. Also *see* Useful Life.

Asset Profile

A template that contains standard depreciation criteria for an asset type and its corresponding asset books. You can use the information in asset profiles as default values when adding assets.

Assignment of Life Insurance

Effective 10/3/94, Federal employees can assign their Basic, Option A and Option B insurance to another person(s), firm(s), or trust(s); Option C is excluded. The assignment of benefits transfers ownership of the FEGLI coverage to the assignee(s). The insured no longer has control over his/her insurance coverage and can no longer designate beneficiaries.

Assignment is irrevocable. Either all or none of the insurance can be assigned. Assignment does not have to be to the same person or firm. Assignments must be made in percentages of total insurance versus an assignment of Basic Insurance to one person and Option A to another. Additionally, terminally ill employees can assign their insurance to a Viatical Settlement Firm in exchange for cash (approx. 60% - 85% of the face value of the coverage). Life Expectancy is usually 24 months or less for a Viatical Settlement Agreement.

Assignment Type

This defines the behavior of the object, (resource, activity, or cost object) within PeopleSoft Activity-Based Management. If the object is identified as a source then costs may be allocated from that object to another object, which must be identified as a target. If an object ID is identified as a target it may be allocated costs from another object ID but may not allocate costs. An object ID can be both a source and a target, thereby having the functionality of each.

Associated Primary BOM

With multiple outputs, it's possible that a given co-product can be created in more than one way – in other words, an item is a co-product on more than one items' primary BOM. By assigning an associated primary BOM to a co-product, you are telling the system which BOM to use in exploding the co-product to the next level.

AT Section

In France, this stands for Section Accident du Travail, or Work Accident Section. It is information needed to identify the establishment risk code for insurance purposes.

ATP Reserved Order

An order that has been promised against future supply. The user has an obligation to the customer to fulfill the order quantity by a certain date. ATP-reserved orders are also referred to as *promised orders*.

Attendance

A component of time reporting application whose purpose is to apply business rules related to Benefit Entitlement and Administration and Organizational Administration to time reported as worked or not worked, and to satisfy a variety of reporting needs.

Attendance Reporting

A Time and Labor report that indicates an employee's attendance record. It includes sick leave, vacation time, and other leaves taken.

Attribute

An attribute is an element within a dimension. For example, the element "Store" is an attribute of the dimension "Geography" for the retail industry. An attribute is also a column heading on an analysis and reporting template.

Audit Trail

See Drill-Back Calculation.

Auditor

Person designated to review expense sheets and cash advances before payment.

Automatic Revision Incrementing (Auto Rev)

The ability to automatically set up revision control and generate revisions for revision-controlled items at the business unit level. This includes setting up a revision scheme or a predetermined, ordered list of revision names.

Automatic Spouse Benefit

A joint and survivor pension benefit provided without any actuarial reduction to a pension benefit. The automatic benefit is a n% joint and survivor; the employee is still entitled to choose any optional form of payment and any beneficiary for the remainder of the benefit.

Availability Date

The date a lot becomes acceptable for fulfillment in PeopleSoft Inventory or for consumption in PeopleSoft Production Management. (Availability Date = Creation Date + Availability Lead Time)

Available to Promise (ATP)

The projected supply of a product less the actual demand, which informs the sales and marketing department of the products that can still be sold without modifying the master schedule. ATP isn't cumulative – it's calculated for each period.

Average Daily Balancing

A feature in PeopleSoft General Ledger that enables you to target the ChartFields on which you base average balance calculations, summarize amounts for selected ChartField values according to your reporting requirements, and define the periods for these calculations.

Used by the financial analytic applications in Enterprise Performance Management. For a reporting period (usually monthly) this refers to the average daily balance of an account as opposed to the month-end-balance, which is the balance as of the last day of the month.

Average Daily Balance Ledger (ADB_Ledger)

In the PeopleSoft Enterprise Warehouse, the Average Daily Balance Ledger table (PF_ADB_LEDGER_F00) is similar to the functionality of the PF Ledger table (PF_LEDGER_F00), in that it too supports reporting. However, the Average Daily Balance Ledger is used for average daily balances. It is a table that is used mostly for processes associated with the financial services industry.

Average Inventory

In PeopleSoft Inventory Planning, one half of the average lot size plus the safety stock when demand and lot sizes are expected to be relatively uniform over time. When demand and lot sizes are not uniform, the stock level versus time can be charted to determine the average.

Average Price

The average price derived from either the bid and ask prices (for bid/ask/average) or from the high and low prices (for high/low/average).

Average Static Calc Flag

In PeopleSoft Inventory Planning, a method used with static policies. The average method sets the static policy equal to the weighted-average, time-phased policy over the next argument periods.

Award

A special payment to an employee for certain prescribed kinds of activities or accomplishments.

B

Back Pay Interest

Under certain circumstances, an employee can be eligible to receive additional pay relative to a delayed receipt in salary caused by administrative error in processing a personnel action. The U.S. Office of Personnel Management has established guidelines for Federal agencies on when and how to make these calculations.

Background Process

Any task or process that is grouped with another and runs in the background. Background processes are usually scheduled to run on a regular basis. All background processes are executed through process-specific COBOL programs run outside the Windows environment.

Backlog Reason Code

An identifier indicating the reason an item could not be shipped. Example codes might include out of stock, discontinued, or seasonal.

BAD Forecast Ratio

In PeopleSoft Demand Planning, the maximum acceptable value of the ratio of the and the base component (Standard Deviation/Base Component). When this value is exceeded, the system automatically resets forecast model parameters. The higher the value, the less likely it is that the system will reset the parameters. In most organizations, a BAD ratio of 1.00 or lower is appropriate for most items.

Balance Segmentation

Balance Segmentation is used in Funds Transfer Pricing to divide balances in deposit accounts between core (stable) and non-core (volatile) segments. Core funds represent the minimum balances that are retained on a long-term basis, building a relatively reliable source of funding to the bank. Non-core funds are temporary in nature due to their volatility caused by customer preferences for liquidity, and cannot be utilized on a long-term basis.

Balance Type

Balance Type is a lookup code used to define the type of instrument balances that will be stored in the PeopleSoft Enterprise Warehouse and processed by the analytic applications. Examples of different Balance are Current Balance, Average Daily Balance, Period Ending Balance, or Commitment Balance.

Balanced Scorecard

See PeopleSoft Balanced Scorecard.

BAM

Business Analysis Model. XXX I think this term is incorrect because we use BAM to refer to the application. If we were referring to the business analysis model, we would say BAM model (that is, Business Analysis Modeler model.)

BAM Model

The BAM database published from the template. The model contains both the data and analytic structure used in the application. The BAM database is physically separate from the Enterprise Warehouse database. Data is sent to the model through migration processes.

BAM Template

A file created using BAM design tools, representing the model prior to its creation as a database. This file has an extension of .MDL. This file is published to a BAM database once the model design process is complete. Each application using BAM will deliver templates which the customer will review and publish to a database in their environment.

Bank Identification Number (BIN)

In PeopleSoft Payables, a part of the bank information that identifies business unit banks.

Base Budget

The initial budget defined by the Budget Coordinator. The base budget is distributed as a starting point for Budget to review and edit. The base budget can be zero-based or incremental.

Base Compensation

In PeopleSoft Workforce Analytics, Cash Compensation that is typically categorized as fixed. It includes base pay and shift differentials as well as associated merit, equity, and step increases.

Base Currency

Base Currency is used to consolidate and report financial results of a multinational company. When a company transacts its business operations in different transaction currencies, those currencies are translated to the base currency for reporting purposes.

Base Currency Equivalent (BCE) Amount

If the monetary amount is in a currency other than the base currency, either the Extract-Transform-Load (ETL) process or the Multi Currency Engine can be used to convert the monetary amount to the Base Currency Equivalent (BCE) Amount.

Base Factor

In PeopleSoft Demand Planning, an element of a smoothing constant simulation set that controls base component smoothing in the Model Reset Simulation process.

Base Metric

Metric found on a fact table. A base metric usually contains an aggregate operator, for example “sum” or “count”.

Base Pay

A pay component included in the job comp (job compensation rate) calculation. It is pay for a regularly assigned workweek. For example, you can set up a regular hourly rate plus a shift rate, a union-negotiated rate for hazardous work, and so on.

Base Pay Structure

A PeopleSoft Workforce Rewards module you use to create or revise pay structures, and to assess the cost and impact of implementing new structures.

Base Time Zone

Customer defined time zone used for converting reported time to a common time zone for ease of applying rules (see Time Administration).

Batch

Batch systems are used when realtime updates are not needed. Batch-oriented data collection applications, developed in-house or by a third-party vendor, produce transactions that are collected in an ASCII text file. The text file is fed to a PeopleSoft SQR program that loads the transactions into the database.

Batch Processes

Any of the background programs in the client/server environment of PeopleSoft applications. Batch processes perform operations—such as pay confirmation, deduction calculation, and so forth—on groups of records, and are usually scheduled to run on a regular basis. You run these processes from the Process Scheduler, and they are executed through process-specific COBOL programs.

Before-Tax Deduction

Deduction that reduces net pay and FWT taxable gross, applied prior to the calculation of federal and state/provincial withholding taxes. Also called “pre-tax” deductions.

Begin Calc Date

The date on which PeopleSoft Asset Management begins to deduct from an asset's life.

Begin Depr Date

The date on which PeopleSoft Asset Management begins to calculate depreciation for an asset. Begin Depr Date is calculated using In-Service Date and Prorate convention.

Benchmark Job

In PeopleSoft Workforce Analytics, this refers to a Job Code for which there is corresponding salary survey data from published, third party sources. Jobs for which there is no corresponding salary survey data are referred to as non-benchmark jobs.

Benefit Commencement Date (BCD)

The date on which a pension payee elects to begin receiving payments.

Benefit Deduction

Any amount taken from an employee's pay check to offset all or part of the cost of the employee's benefits.

Benefit Eligibility

The PeopleSoft Pension Administration function that determines if an employee is eligible for retirement or ancillary benefits. A plan may have several retirement types—normal, early, late, death, and disability—each with its own eligibility criteria.

Benefit Entitlement

Any rules governing the circumstances under which employees are entitled to receive certain benefits. Typically, entitlement to benefits is based on type of employee (for example, full time, part time, occasional), length of employment, and specific rules which apply thereto, i. e., work group affiliation, and compensation base. Other criteria may also apply, such as reasons-for-claiming or job performance.

Benefit Formula

The formula that determines a participant's pension benefit in a defined benefit plan, as well as the PeopleSoft Pension Administration function that calculates the benefit.

Benefit Group

Part of a group of defaults assigned to job codes. Benefit group may include medical, dental, and health benefits dependent on individual company parameters.

Benefit Plan

A specific benefit within a plan type. For example, your company's life plan type might include benefit plans of one times salary, two times salary, and three times salary.

Benefit Plan Type

Any category of benefit, such as health, life, or savings.

Benefit Program

A set of benefits and deductions valid for an employee or group of employees. A single company may have any number of programs. An individual employee may belong to only

one program; the deductions and benefits contained in that program are the only valid deductions and benefits for that employee.

Benefit Tables

Any of the tables that contain employee benefits information. These are often relevant to payroll processing.

Benefits Base

The salary used for benefit calculations. The benefits base will be either the employee Annual Rate or Annual Benefits Base Rate.

Benefits Compensation

In PeopleSoft Workforce Analytics, Benefits Compensation is value associated with employment benefits. It can include benefits types for Health and Welfare (Medical, Life Insurance), Retirement (annuities, savings plans, pensions), and Paid Time Off (Vacation Leave, Sick Leave). Benefits compensation is sometimes fixed, and sometimes variable, depending upon the benefit type.

Betriebszählung (Company Statistics Report)

Also called the OFIAMT report. This report provides statistics required by the Swiss Federal Department of Statistics (BFS).

Bias Signal Limit

In PeopleSoft Demand Planning, a number between one and six that indicates how many **Forecast Period** to test for bias. If the bias test is violated, the system records a **Tracking Signals** error in the period up to the number of periods determined by the bias signal limit.

Bias Test

In PeopleSoft Demand Planning, a forecasting test that sets the limit for tripping a **Tracking Signals**. The lower the value, the more likely it is that a tracking signal is set.

Bid Price

The price a prospective buyer is prepared to pay at a particular time for trading a unit of a given security.

BIF file

This is the bulk insert file (input.bif) used with the Verity search engine to specify the documents to be submitted to a collection (search index). It contains a unique key, document size (in bytes), field names and values, and document location in the file system.

Bilan Social Report

See Employee Survey Report.

Bill

In PeopleSoft Billing, any group of bill lines.

Bill Adjustment

The process of making credit or credit and rebill adjustments to an invoiced billing activity.

Bill By Identifier

The Bill By Identifier is used to define how billing activity is grouped when added to a bill through the billing interface or the Populate Billing process.

Bill Header

The record containing information that pertains to the bill as a whole. Each bill has a unique bill header that identifies it within the system.

Bill Inquiry Phone

Bill Inquiry Phone is the number printed on your invoices for your customers to call if they have any questions about their bill.

Bill Line

The basic unit of billing activity representing a billable charge, including the charge identifier, quantity, price, and any other information regarding an individual transaction. Every bill line is related to a bill header that may have one or more bill lines related to it.

Bill Search

A method of finding a bill or bill line when you don't have enough information to call up the bill directly. **Customer Bill Search** enables you to locate a bill by Customer Name. You can also choose other parameters to limit your search. With **Bill Line Search** you first search for a particular bill and then a line on that bill. Parameters for bill line search include Reference, Date, and Amount.

Bill Source

The point where billing activity originates. Bill sources may be external to the system (imported through the billing interface) or entered directly online. Examples of bill sources include order management, project costing, and contract administration.

Bill To Customer

A customer who receives an invoice.

Bill Type

A category of billing activity variety. Examples of Bill Types include standard and custom order activities.

Bill Update

The process that adjusts bills that have either been entered manually or generated within the system.

Billable Indicator

A status flag that identifies an item as eligible for billing to a customer.

Billback Discount (BB)

A per unit discount which typically requires a customer to perform one or more merchandising activities to receive the discount. A BB discount is not deducted from the customer invoice, but once the customer performs the merchandising activity, a sales representative or broker can approve payment for the discount amount. Billback discounts can originate from a National Allowance or Customer Promotion, and are passed to PeopleSoft Order Management for informational purposes only. Billback discounts are recognized as a liability when the product is shipped.

Billing Location

A number identifying a customer address. Each customer may have multiple locations, but must have one *Primary Location* at which you contact them.

Blackout Period

The period of time, determined by the company, which prohibits certain activity in the company stock. Blackout Periods can affect the trading of some key individuals or can be placed on the entire company.

Bonus Tax Method

Annualizes your year-to-date earnings by multiplying them by the number of pay periods in the year. This method is used for Canadian tax processing.

Book

In PeopleSoft Asset Management, a data location storing financial information—like cost, depreciation attributes, and retirement information—on assets.

Borrow/Loan

The temporary reassignment of an employee to other task reporting or compensation requirements to allow the business to meet unexpected, short-term, fluctuations in staffing or work load. Typically, this kind of reassignment is done informally at a local level, where HR isn't involved and a new job record isn't created. Companies may have specific rules about how long an employee may be borrowed/loaned, how and where productive, non-productive, and compensated absence time will be charged, and what business rules to apply to the borrowed employee's time for the purpose of compensation and benefit entitlement and administration. See also Casual work Assignment.

Bracket

Brackets are a way to look up and retrieve database table values. After you've defined a table, the system finds a corresponding row on that table and returns the value of the bracket. The result is then available for use in other items such as formulas.

Branch

A tree node that rolls up to nodes above it in the hierarchy, as defined in the Tree Manager.

Branch Of Military Service

Identifies, if any, military service in which the employee served.

Breadcrumbs

Breadcrumbs show the navigation path to the current web page location. As you drill down through the different levels of the registry, a “breadcrumb trail” appears that shows the path you’ve selected. Each registry level is separated by an angled brace (>), and you can select any level to navigate directly back to that level.

A typical Breadcrumb would look like this:

Home > HR > Administer Workforce > Benefits

Break Funding

Charges assessed for mortgages that are paid off before maturity. In the Funds Transfer Pricing (FTP) application, Break Funding charges are factored into the transfer price for a loan that may be prepaid.

Break in Service

A period of time for which an employee does not meet stated service requirements.

Break Price

The price used to determine which options are eligible for repricing. For example, if the break price is \$36, then all outstanding option with a grant price of \$36 and greater are eligible for repricing.

Break Punch

An in/out punch of when a time reporter takes a break.

Brokers

Individuals or organizations who buy and sell securities. Often they are account executives who work for firms registered with the Stock Exchanges and the SEC. Unlike Transfer Agents, (who are not responsible for sales) Brokers do not maintain records on all your company’s certificates. They maintain only sales records and stocks for their clients.

BSC (Balanced Scorecard)

See PeopleSoft Balanced Scorecard.

Budget Activity

A type of activity performed using PeopleSoft Budget Planning. Budget activities include Line Item Budgeting, Line Item Mass Adjustments, Budget Allocations, and Position Budgeting. PeopleSoft Budget Planning-specific.

Budget Amount Ledger

Stores budget amounts and is updated by posting budget entries, transfers, and adjustments.

Budget Analyst

A role within PeopleSoft Budgeting. Budget Analysts are typically people within an organization responsible for reviewing and analyzing a prepared budget before submitting it to the Budget Coordinator. PeopleSoft Budgeting-specific.

Budgetary Account Only

An account used by the system only and not by users; this type of account will not accept transactions. You can only budget with this account. Formerly called “system-maintained account.”

Budget Category

A set of related expenses that are accumulated for proposal budgets and reporting to a sponsor. The estimated cost for a set or class of accounts.

Budget Category

Numeric/alpha identification given to each category of positions.

Budget Center

In PeopleSoft Budgets, any entity responsible for producing or reviewing budget data. For example, a Budget Center might be the individual departments responsible for producing budgets.

Budget Center Dimension

In PeopleSoft Budgets, the dimension by which you distribute budget data. If you budget by department, your department dimension will be your Budget Center Dimension. You'll assign Budgets Users to the nodes and detail values on the tree you use to build your Budgets Center Dimension.

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 Check Override

Selective suspension of Budget Processing. With this feature you can override the controlled budget for a transaction that failed budget checking due to insufficient funds; or override the tolerance limits for a transaction rejected due to exceeded tolerance limits. When you push the Override button, the system flags the transaction to allow the Budget Processor to process successfully regardless of available funding. You can cancel the override any time before the Budget Processor is run by clicking the Cancel Override button.

Budget Control

In commitment control, it ensures that commitments and expenditures don't exceed budgets. It enables you to track transactions against corresponding budgets and abort 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 Coordinator

A role within PeopleSoft Budgeting. Budget coordinators are responsible for monitoring the budget process. The Budget Coordinator is typically located within an organization's central budget office and builds the budgeting model. PeopleSoft Budgeting-specific.

Budget Detail

A level of itemization that when combined makes up a major budget category.

Budgeted Rates

In PeopleSoft Activity-Based Management, the rate your organization uses based on the budget.

Budget Error Exception

A transaction that fails budget checking, causing an Error or Warning to be issued. See **Error Exception** and **Warning Exception**.

Budgeting Functions

PeopleSoft Budgeting's six main action categories, including: system administration, budgeting setup, budgeting preparation, budgeting analysis, data integration and my profile. Your user role determines how many of these functions display and are available.

Budgeting Model

The framework for an organization's budget development process. Business unit defines a Budgeting Model. The Budget Coordinator typically defines the model and includes the time period of a budget cycle, time period for phases within a budget cycle, the sources of data that will be available to budget users, the methods that will apply to line-item budgets, and other budget options and control parameters. PeopleSoft Budgeting-specific.

Budgeting Type

Associated with the budget ledger type set definition, a budget type is an indication of whether the organization uses a standard budget ledger, project budget ledger, or controlled budget ledger for budgeting.

Budget Justification

Written explanation further defining the what and why of a budget category.

Budget Period

The period in which you define plans to meet your organizations training requirements.

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.

Budget Phase

In PeopleSoft Budgets, a span of time during which a budget or portion of a budget is to be completed. You'll filter dimensions, assign alternate Budgets Users, enable Position and Asset budgeting, and specify Budgets User notification options at the Phase level.

Budget Plan

In PeopleSoft Workforce Rewards, when working with a Compensation Planning BAM model. A budget plan is a rollup of like compensation rules. For example, for base pay rules budget plans are a rollup of values for like Action Reasons. For variable pay rules budget plans are a rollup of the values for like Variable Compensation Plan IDs.

Budget Preparer

A role within PeopleSoft Budgeting. Budget preparers are typically people within an organization responsible for developing the detailed budget for a Budget Center and submitting it to a Budget Reviewer or Analyst for review and approval. PeopleSoft Budgeting-specific.

Budget Reviewer

A role within PeopleSoft Budgeting. Budget reviewers are typically people within an organization responsible for reviewing and approving a prepared budget submitted by a Budget Preparer. PeopleSoft Budgeting-specific.

Budget Seeding

Represents a new budget or forecast, such as historical data that is manipulated to develop a more current representation for a proposed budget. Uses detail data as the budget seed or basis to create the base budget that represent the level of detail in which budget numbers are prepared.

Budget Translation Trees

Trees translate (summarize) source transactions into the appropriate levels for processing against control budgets. This is because you usually budget above the level of your source transaction ChartFields on a tree.

Budget Type

Indicates whether a budget is for expenditures or revenues.

Budget Warning

See **Warning Exception**.

Budgets User

In PeopleSoft Budgets, any user who needs to gain access to the Budgets. You'll designate Budgets Users on the Budgets Users page through the Coordinate Budgets window. You'll also assign these users to the tree representing your Budget Center Dimension.

Budget View

A user-defined view where selected dimensions, columns and rows of data determine the layout of line-item budgets affecting the view or entry of data.

Budget Year

The institutionally defined, consecutive, 12-month period to which a financial transaction or summary applies.

Build Option

A detailed PeopleSoft Planning model that specifies a method of building an assembly item. This model specifies the routing, resources, and materials that are necessary to produce the item.

Built-in function

Prior to PeopleTools 8.0, there were only built-in functions, like FetchValue, ScrollSelect, etc. A built-in function, in your code, is on a line by itself, and doesn't (generally) have any dependencies. You don't have to instantiate anything before you can use a built-in.

Business Interlink Definition

A definition encapsulating an external Transaction or Query and providing a set of generically typed input/outputs that can be assigned to PeopleCode variable or Record Fields at runtime. A Business Interlink Definition is added to the Application Designer's objects at the same level as Fields, Records, Panels, etc.

Business Interlink Design-Time Plug-in

An XML file that, when coded for an external system, encapsulate that external system and provide a catalog of Transactions, Classes and Criteria specific and meaningful to that external system.

Business Interlink Framework

The framework for integrating any external system with PeopleTools application objects. It is composed of the following components:

1) An External System, 2) Generic definitions for a Transaction/Query command interfaces, 4) Business Interlink Definitions, 4) Business Interlink Plug-in.

Business Interlink Object

An instantiation based on a Business Interlink Definition. Actual data can be added to the inputs of the Business Interlink Objects once the appropriate bindings are provided. The Business Interlink Object can be executed to perform the external service. Once a Business Interlink Object is executed, the user of that object can retrieve the outputs of the external service. The Business Interlink Objects use buffers to receive input and send output. When a Business Interlink Object is executed, the transaction/query/class associated to the Business Interlink Object will be executed once per each row of the input buffers corresponding to the input Records. If there is only one row, after appropriate substitution by the driver, it is executed only once.

Business Interlink Runtime Plug-in

A set of C++, Visual Basic, or other high-level language methods that, when coded for an external system, encapsulate that external system and provide the execution methods to match the Business Interlink Design-Time Plug-in. (The catalog of Transactions, Classes and Criteria provided by the Design-Time Plug-in can also be provided by the Runtime Plug-in.)

Business Objects

A way of identifying those mass changes that have been designed to be referenced by a flexible formula and provide them with a shorter name to simplify the creation of flexible formulas.

Business Planning

The type of planning that focuses on elimination activities that are not needed by changing the drivers.

Business Rules

Rules that can process information differently depending on the values of data in the PeopleSoft Enterprise Warehouse.

Business Unit

A corporation or a subset of a corporation that is independent with regard to one or more operational or accounting functions. PeopleSoft General Ledger business units typically comprise individual entities for accounting purposes.

Business units in PeopleSoft Projects represent operational structures but not necessarily independent financial units.

PeopleSoft Payables business units are either *Vouching* (have payables accrued to them) or *Charge to* (have voucher expense distributions charged to them), and pass journals to general ledger units.

PeopleSoft Purchasing business units share vendor, purchase order, and receiving information with PeopleSoft Payables units in the same SetID.

A PeopleSoft Inventory business unit is a storage facility that maintains its own replenishment and costing methods, as well as its own definitions and guidelines.

The Manufacturing business unit must be identical to the Inventory business unit in order to link the manufacturing and inventory processes.

The Order Management business unit controls certain order processing parameters (tax and freight calculation methods, base currency, credit card hold options, and so on) for its associated PeopleSoft eStore and Mobile Order Management merchant variants.

Business Unit Audit List

One or more business units specifically targeted for expense report and cash advance audits.

Buying Agreement

You can structure flexible and easy-to-use buying agreements for customers or groups of customers. You can set up maximum amounts and specify the minimum dollar value per order placed against it. You can automatically generate sales orders or create sales orders online from buying agreements. Rebate and penalty calculations can be implemented for buying agreements.

C**Cafeteria-Style Benefits**

Any programs offering several benefit plans from which participants make elections. Cafeteria-style benefits may or may not include flexible credits.

Calculation

In PeopleSoft Pension Administration, the determination of a participant's pension benefit.

Calculation Rule

Criteria for calculating benefits, including as-of dates for age, service, premium, and coverage calculations; rounding rules; and minimum and maximum coverage amounts. Any number of program and plan combinations can use a single set of calculation rules.

Calculation Rule [Global Payroll]

Any rule you develop using combinations of elements to command the system to perform a type of calculation.

Calendar

In PeopleSoft Manufacturing, a list defining the days your enterprise is available and the hours of operation for each day. The system first looks to see whether you are using a work center specific calendar. If none is defined, it looks at the production calendar. If no production calendar is defined, planning and scheduling functions base start and due dates on a five-day workweek.

In PeopleSoft Demand Planning and Inventory Planning, a list defining the start and end dates for each time-phased period. It also contains daily weights for distributing raw data into different period buckets.

In PeopleSoft General Ledger, your accounting calendar defines the time periods to which you post transactions for different ledger group and business unit combinations. You can have multiple calendars, so you can keep a calendar for actuals, another for budget and forecast activity, and still others for special reporting or transitional needs.

Calendar Group ID

Allows you to group together multiple Calendars that you want to run together at the same time. It also controls the order in which the Calendars are processed. You can only group calendars together that are for the same country (based on pay entity country).

Calendar Scope

A time period type (Day-Factored, Month-Factored, or Week-Factored) for use in building your time period calendar.

Canada Academic Teaching Surveys

Statistics Canada requires that all Canadian universities (all degree granting institutions) produce full-time and part-time *Canada Academic Teaching Surveys*. These reports are a legislative requirement. PeopleSoft HRMS 8 provides you with the functionality to code HRMS information using Statistics Canada codes and create both the full-time and part-time Academic Teaching Surveys.

Canadian Industrial Sector

The Canadian industrial classification code with which employees are associated for Canadian employment equity reporting purposes.

Canadian National Occupational Classification (NOC) Codes

NOC codes are occupational classification codes for Canadian companies provided by the government.

Canadian Standard Occupational Classification (SOC) Codes

SOC codes are occupational classification codes for Canadian companies provided by the government.

Cancellation

A process that terminates stock fulfillment requests, allowing reserved and allocated items to be returned to inventory.

Cancellation

In the context of an employee stock plan, a transaction (usually triggered by a specific event, such as a termination of employment) in which outstanding securities are declared void and inactive and returned to the pool of securities reserved for issuance under the plan or retired.

Candidate Keys

In PeopleSoft Demand Planning, elements of data that can be used to construct the **Forecast Item** key field at different levels of the forecast.

Capacity Rate

A rate you assign to a capacity cost object. This enables you to track and report on excess capacity.

Capacity Fence

A time fence that indicates that date and time after which PeopleSoft Enterprise Planning or Production Planning solvers ignore capacity violations. The solvers do not use this date in processing capacity violations.

Capacity Multipliers

A multiple used in PeopleSoft Enterprise Planning and Production Planning to determine the available capacity on a resource. Since a capacity multiplier is effective-dated, you can use it to vary the resource's available capacity over time.

Capital Acquisition Plan (CAP)

A method of projecting and tracking capital expenditures for a project. Budgeted assets and actual expenditures can be associated with a CAP Plan so the owner can track planned against actual costs.

Capital Gain

The difference between an asset's purchase price and selling price, when the difference is positive. Capital gains can be either short-term (where the capital asset was held for 12 months or less) or long-term (where the capital asset was held for 12 months or more).

Capital Gains Tax

A tax on profits from appreciation in owned real property, recognized at the time the property is sold; real property includes owned company shares.

Capitalization

The total types and amount of the outstanding securities that have been issued by a corporation. Generally includes both equity and debt securities.

Capital Markets Instrument

In the financial services industry, Capital Market Instruments are assorted financial instruments issued by organizations to raise capital for funding operations. Participants are made up of interested parties that choose to supply or acquire the capital funding through such vehicles. Derivatives, debt instruments, equities and foreign exchange instruments that are traded in highly liquid markets represent the instruments. In the PeopleSoft financial analytic applications, Capital Market securities refer to instruments that are bought/sold by the institution for its own investment account. The capital markets set the product prices and interest rates.

CAP Sequence Number

The number that distinguishes a small project belonging to a CAP plan. Budgeted assets can be associated with an overall CAP Plan and a CAP Sequence, if that level of detailed tracking is desired.

Carry-Forward

Residual contributions that remain in a stock purchase participant's account after the purchase of shares that are used toward future purchases.

Carrying Cost

In PeopleSoft Inventory Planning, a value that shows the cost associated with holding a dollar of inventory for one year. The value is presented as a percentage.

Case Officer

In Germany employees in your company are designated as Case Officers, and have responsibilities for handling health and safety incidents.

Cash Balance Accounts

The PeopleSoft Pension Administration function that tracks the activity in an employee's hypothetical account under a cash balance plan.

Cash Balance Plan

A defined benefit plan designed to look like a defined contributory plan. The plan periodically credits a percentage of pay to each employee's hypothetical account.

Cash Compensation

In PeopleSoft Workforce Analytics, Cash Compensation is a component of direct compensation. Cash Compensation consists of direct cash payments made to an employee for base compensation and short-term variable compensation.

Cash Exercise

At the time of exercise, the optionee is required to pay in cash the total option price plus any withholding taxes due to the company.

Cash Flow Generator

This is a support module for the PeopleSoft financial services analytic applications. It generates actual and projected cash flows for financial instruments by using output from the other support modules, such as loan prepayment rates, deposit runoff rates, product pricing indices, discount rates, and product definitions (such as start and end dates, balance amount, interest rate, term, payment dates, repricing and compounding frequency, and accrual basis) to generate the cash flows. The Financial Performance Measures module accesses the cash flow results to calculate the required financial measures.

Casual Preparer

An additional user role at the lowest level of budget preparation for a budget center. This user performs the same activities as the Budget Preparer role when access is granted. The system does not, however, enable the Casual Preparer role to define their own private views for line-item budgeting.

Casual Work Assignment

The temporary assignment of an employee to a work position or location to meet the needs of the business. Typically, there is no Human Resource activity to support the work assignment (that is, a new Job record is NOT created). Often compensation rules that accrue to the temporary assignment override the compensation rules that apply to the employee's normal work assignment. See also Borrow/Loan.

Catalog

The list of transactions, classes, and queries used to interface to the external system. Integration users are presented with this list when they pick the type of Business Interlink Plug-in they are going to use. There are four types of catalogs: transaction, class, operator, and configuration parameter.

Catalog

A way of organizing your training courses into classifications for increased flexibility. Catalogs consist of categories and subcategories.

Category

Categories are the primary level of a two-tier structure of training courses. Categories can consist of subcategories that provide further course definition.

Category Tree

A hierarchical structure that groups products by category to control how they are displayed in PeopleSoft eStore web pages. Used also by Mobile Order Management to enable product information to be accessed by a wireless device.

CBM

See PeopleSoft Customer Behavior Modeling.

Census Metropolitan Area (CMA) Code

In Canada this code is prescribed by the government and refers to the area of an urbanized core with a population of at least 100,000.

Central Personnel Data File (CPDF)

Two types of reporting made by agencies to the OPM include the Dynamic and Status files (quarterly and monthly, respectively) covering a range of employee personnel/payroll data.

Certain and Continuous Payment Option

A form of pension payment where the benefit is paid out for the lifetime of the participant with a specified number of payments guaranteed so that a beneficiary will receive payments until the end of the guarantee period if the employee dies before the guaranteed payments are complete. For example, under a ten-year certain and continuous payment option, a retiree who lives less than ten years receives payments until death, then the retiree's beneficiary continues to receive payments for the remainder of the ten year period. A retiree who lives longer than ten years continues receiving payments after the ten year period until death. Also known as a "Term Certain and Continuous" payment option.

Certain Only Payment Option

A form of pension payment where the benefit is paid out entirely over a specified period of time—usually five, ten, or fifteen years—with no ongoing payments after the specified period. If the retiree dies before payment period is over, the remaining payments are made to a beneficiary. Also known as a "Term Certain" payment option.

Change To Lower Grade

- For positions under the General Schedule or under the same wage grade schedule, a change-to-lower grade changes the employee to a lower grade; and
- When both the old and new positions are under the same type ungraded wage schedule, or in different pay-method categories, a change-to-lower grade changes the employee to a position with a lower rate of basic pay.

Charge Out

A **Material Issue** used when the item is scheduled for future return.

ChartField

A field storing 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

PeopleSoft enables you to set up ChartFields and indicate that you want specific ChartFields to match (balance) on the debit and the credit side of a transaction. When you work with Controlled Budgets, the Fund and Budget Period are already set up in the system to balance (match). For example, suppose you want to balance by Class and Program. You indicate that these on a panel that these ChartFields are required, along with Fund and Budget Period which should already be selected. When you enter a transaction, you must enter the same Class, Program, Fund, and Budget Period ChartFields on both sides of the accounting entry. but you can modify any ChartFields, other than these four, on the user-defined line. The system always requires that total debits equal credits.

ChartField Combination Edit

Also called *Combo 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.

ChartViews

Charts of data in the model, presented through the Worksheet which retains the ability to drag dimensions on the chart as desired.

Check In/ Check Out

The process of retrieving planning activities from the BAM database (check out) and posting changes and results back into the database (check in).

Child

A node or detail on a tree linked to another, higher-level node (referred to as the parent). Child nodes—projects, customers, and so on—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.

Child

A node or detail of a tree linked to another, higher-level node referred to as the parent. Child nodes can be rolled up into their parent. A node can be a child and a parent at the same time depending on its location within the tree.

Chunking

Chunking is a PeopleSoft Enterprise Warehouse mechanism that makes voluminous processing easier through the use of multiple small parallel processes. By enabling chunking, multiple jobs are spawned from one Jobstream. These jobs run in parallel (behind the scenes) to process data efficiently.

Citizenship Code

Numeric indicator as to whether the employee is a U.S. citizen or a foreign national serving in the U.S. The codes are:

- citizen
- other

Civil Service Retirement System (CSRS)

A retirement plan available to employees of the federal government. CSRS covers all employees appointed to a position in the federal government before January 1, 1984. Coverage includes a basic annuity plan with employee contributions and the Medicare Hospital Insurance component (1.45%) of the Social Security tax.

Class catalog

Lists classes used to interface to an external system. A class contains data members of basic types and/or objects that are typed after another class. A Class can also contain lists of basic types or objects.

Class ChartField

A ChartField value that identifies a unique appropriation budget key when you combine it with a Fund, DeptID, and Program Code as well as a Budget Period. Formerly called “sub-classification.”

Classification Code

Need App A code that categorizes an engineering change. Example classification codes include the following: Mandatory, Optional, Upgrade, Quality, and Safety.

Clock Hour Reporting

Method of reporting time by recording actual times in and out (start and stop) (see Time Reporting).

Clone

To create a unique copy of an object. When used in PeopleCode, clone will always mean to make a unique copy. Copy, on the other hand, may or may not mean making a unique copy. Copy may mean making a new reference to an object, so if the underlying object is changed, both the copy and the original change.

Cloning

The process that enables you to copy run controls to create employee schedules from existing Run Control ID's that have already been executed and saved.

Close Date

The date in which time entry is no longer allowed for a given pay period. Defined as an offset number of days to the pay period end date.

Close Price

The price of the final trade for a security at the end of the trading day.

Closure Calendar

A calendar that establishes closure dates for shipping, receiving, and materials management operations for a specific **Business Unit**. Typically, application processes account for these closure dates when determining Lead Time and dates for anticipated fulfillment processing dates (scheduled shipment dates, scheduled arrival dates, and lot retest dates, for example).

CMA (Census Metropolitan Area) Code

In PeopleSoft Workforce Analytics, the CMA code is prescribed by Statistics Canada, and refers to the main labor market area of an urbanized core with a population of at least 100,000.

COBRA (Consolidated Omnibus Budget Reconciliation Act)

In PeopleSoft Workforce Analytics, this refers to legislation that requires employers to offer continued health care coverage to employees, and their dependents, who lose benefits coverage under certain defined conditions such as voluntary termination, divorce, becoming an overage dependent, or retirement. Any individual, whether employee or dependent, that is covered under a health plan at the time of a qualifying event, has the option to elect COBRA coverage.

Codepage

One character set.

Collection

To make a set of documents available for searching in Verity, you must first create one or more collections. 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 matching various search criteria. A collection is a set of statistics and pointers to the source documents, stored

in a proprietary format on a file server. Since a collection can only store information for a single locale, PeopleSoft maintains a set of collocations (one per language code) for each search index object.

Combined Federal Campaign (CFC)

A vehicle used by federal employees to contribute to a charity or charities of their choice.

Commercial-Off-The-Shelf (COTS)

Equipment or software that is currently sold commercially to at least one customer.

Commission Tax Method

A payroll tax calculation method that adds year-to-date earnings to earnings for this pay period and finds the annualized gross by multiplying by the number of pay periods in the year; the gross is then divided by the number of tax periods specified on the paysheet. This method is used for Canadian processing only.

Commitment Control

Commitment control includes budget control and commitment accounting functionality.

Common Shares Issued and Outstanding

Represents the residual ownership interests in the corporation. This is the composite number of shares available and tradable on the open market.

Community Background

In the United Kingdom Community Background refers to the religious category, such as Catholic or Protestant, of employees, job applicants or appointees. See the Northern Ireland Report for more information.

Compa-Ratio

In PeopleSoft Workforce Analytics, Compa-Ratio is most commonly defined as the relationship between current pay and the midpoint calculated as: $(\text{Incumbent Pay} / \text{Midpoint}) * 100$. Usually expressed in whole numbers, or in percentage form by dropping the multiplication operation. Much less common is the use of a compa-ratio calculation as: $\text{range midpoint} / \text{market rate}$.

Compensation Frequency

In PeopleSoft Workforce Analytics, this is the frequency at which a job is paid. This is the value you use for reporting or quoting pay. Examples include Annually, Monthly and Weekly.

Compensation Planning

In PeopleSoft Workforce Analytics, this is the process through which employee compensation plans are defined, and compensation budgets are allocated throughout an organization. Major components of compensation planning include designing pay structures, setting individual pay levels, and budgeting and forecasting compensation spending.

Compensation Rate

In PeopleSoft Workforce Analytics, this is the compensation rate for a job. This is the rate the company uses for quoting and reporting pay.

Comp time (compensatory time)

A PeopleSoft Time and Labor-managed employee benefit where time off is granted in exchange for time worked based on customer-defined criteria; is associated with an expiration and is used as reported time (see Attendance).

Compensation

The process by which a worker is remunerated for services rendered to, or work performed on behalf of a business entity.

Compensation Package

All of the base and non-base components on a job row.

Compensation Rules

Business methodology or logical process that is applied to reported time in order to determine payable time (see Time Administration).Competency

In PeopleSoft Workforce Analytics, Competency is a knowledge, ability, skill, accomplishment, or National Vocational Qualification (NVQ).

Competency Inventory

All of the roles, tasks, competencies and accomplishments possessed by the workers in the current workforce. This data is migrated from internal source systems into the data warehouse tables of the PeopleSoft Enterprise Warehouse.

Competency Strategy

The type and number of roles, tasks, competencies and accomplishments essential to accomplishing a business scenario based on your strategic business goals.

Competitive Appointment

An appointment to a position in the competitive service following open competitive examination or under direct-hire authority. The competitive examination, that is open to all applicants, may consist of a written test, an evaluation of an applicant's education and

experience, and/or an evaluation of other attributes necessary for successful performance in the position to be filled.

Competitive Service

All positions as defined by 5 USC 2102 in the executive branch of the Federal Government are in the competitive service unless they are specifically excluded from it. Positions in the legislative and judicial branches are outside of the competitive service unless they are specifically included.

Compress

The act of placing a Planning task as early as possible in the schedule without violating any constraints.

Compressed Split

In PeopleSoft Demand Planning, an optional function that allows a split database to be compressed so it can be transferred to an account manager's computer.

Concurrent Offerings

Multiple stock purchase offerings that are active and outstanding at the same time. The end date is measured from the employee's grant date.

Concurrent Processing

The situation in which you run multiple batch processes at a time. In PeopleSoft Benefits Administration, for example, simultaneous open enrollment and event maintenance qualifies as concurrent processing.

Configuration Code

A unique 50-character identification code that accurately tracks and costs inventory with the PeopleSoft Product Configurator. It corresponds to a lot number for a non-configured item.

Configuration Costing

The overall process of reviewing and evaluating anticipated cost data for a configured item.

Configuration parameter catalog

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

Consolidate Assets

In PeopleSoft Asset Management, the process of consolidating multiple load lines, usually coming from a separate application, into one asset.

Consolidate Depreciation

In PeopleSoft Asset Management, the process of summing all open Add and Adj transactions by transaction type, **Transaction Date**, and accounting date for all composite members reporting to one composite asset.

Consolidated Bill

A grouping of bills gathered together for invoice presentation. The bills belonging to a consolidated bill are invoiced and printed together, with a page summarizing the bills as a group.

Consolidations

The PeopleSoft Pension Administration functions that accumulate hours, earnings, and pension contributions based on payroll data.

Consolidations-Elimination Set

A related group of intercompany accounts that is processed during consolidations. Once eliminated, this group of accounts should normally net to zero.

Constraint

A limit to a schedule, that, when violated, must be repaired to produce a valid schedule. User-configurable Planning constraints include Missed Request Dates, Missed Promise Dates, BI Shortages, RM Shortages, Capacity Overloads, Missed Inventory Targets, Changeovers, and Excess Inventory. See also **Temporal Constraint**.

Constraints

In the PeopleSoft Enterprise Warehouse, a constraint can consist of one or more filters and is used to define complex business logic. Constraints are based on DataMaps.

Consumption Pattern

In PeopleSoft Activity-Based Management, an attribute used to describe how an activity interacts with the cost objects to which it has been assigned. A unit type activity can expect to be performed on a regular basis so that each time a product is produced. A batch type activity may only be performed periodically for a given range of transactions. For example, each time a machine is setup to produce another product type. Sustaining type activities generally occur to support the overall operation of a company unrelated to products produced or customers served.

Contact

A person associated with a Customer ID. Contacts can be internal contacts or external contacts. Internal contacts are your employees who manage the relationship with your customers, from handling billing inquiries to product/warranty questions, to basic product/service questions. Interactions with customers can be recorded via PeopleSoft Conversations. Self service interactions can be recorded through PeopleSoft Contact Us. External contacts are your customer's representatives who can access self-service transactions

and receive documents such as sales order acknowledgements. Contacts must have a User ID to access self-service transactions.

Contact Us

A method by which customers and unregistered guest users send email messages to specific addresses or members of the merchant's organization. Merchants can also define automatic response messages.

Container

An Inventory stock unit for receiving, putaway, bin to bin transfers, picking, shipping, adjustments, and physical accounting. Each container is associated with a unique container ID.

Content Reference

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

Contextual reference

PeopleCode refers to a row or buffer field determined by the current context; that is, the context in which the PeopleCode program is currently executing.

Contingent Beneficiary

In PeopleSoft Pension Administration, any non-spouse pension beneficiary, including a child, other relative, or a trust. Spousal consent is required in order for an employee to name a contingent beneficiary.

Contracting Officer (CO)

Individual who has the authority and the official responsibility to produce a sound acquisition document.

Contracting Officer's Technical Representative (COTR)

Individual responsible for monitoring a contract and its associated tasks and deliverables.

Contractor

Any individual or non-employee reporting time that will not be paid through the payroll system.

Contribution

Represents money a stock purchase participant elects to contribute to the plan. Contributions are deducted from the participant's paycheck and used to purchase stock pursuant to the offering and purchase period they are enrolled.

Contributory Plan

A Pension plan to which employees contribute. Contributions are typically a percentage of pay deducted from the employees' paychecks.

Control Budget

Commitment control enables you to establish budgets that provide extensive, active budgetary controls over transactions, rather than just passively recording transactions.

Control ChartField

A control ChartField is a key ChartField that you designate to be the control field. Designating a ChartField as the control allows you to set attributes for a specific value of the ChartField that are different from the attributes specified for the budget type in general. For example, if the tolerance for a Projects budget type is set to 10% in general, you can override this value, making it higher or lower for specific projects.

Control Group

A mechanism to relate vouchers together for the purpose of controlling voucher input into PeopleSoft Payables. Generally used for assigning vouchers to data entry personnel and for reviewing input.

A set of parameters that determines the major forecast process options. The Control Group code is assigned to a group of **Forecast Item** and controls the forecast development and tracking for each item in the group.

Control groups are used by the Analytic Forecast Component to govern particular properties of the forecast rule, such as what accuracy to expect and what statistical method to apply. Forecast elements are assigned to exactly one control group. They manage differences among forecasts within a set.

Control Hierarchy

The relationship between business units, origins, vendors, and control groups in PeopleSoft Payables that defines which processing data will be automatically entered on each voucher.

Control Number

A sequential identifying number used to identify an exercise.

Control Plan

In PeopleSoft Quality, a plan that brings together application, measurement, and control and response criteria for a specific product and process.

Conversation

Any notes, transcript, or detail of a telephone call between an employee and a customer. Conversations may be tied to items, payments, purchase orders, document references, or bills of lading.

Conversion data profile

A conversion data profile takes the values from a particular PeopleSoft database table (such as the table holding bank transaction codes) and specifies how that value appears in PeopleSoft Business Documents.

Conversion data profile

A conversion data profile takes the values from a particular PeopleSoft database table (such as the table holding bank transaction codes) and specifies how that value appears in PeopleSoft Business Documents.

Conversion Loader

A sample SQR delivered with PeopleSoft Asset Management that transfers data from multiple fixed-length ASCII files into sample, relational conversion tables.

Copy Bill

In PeopleSoft Billing, the online environment providing for the replication of a single bill, generating a new bill with its own unique invoice number.

Core Functionality

Core functionality is the set of information in PeopleSoft HRMS that is common to your entire global workforce tracking needs—and is always displayed on the primary page.

Core hours

The hours a workday, workweek or pay period in which a time reporter must be present for work in a flexible work schedule (see Scheduling).

Corporate Account

In PeopleSoft applications, this is equivalent to the Account (ACCOUNT) ChartField. The term is used to make a distinction 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.

Corporate Reporting

Companies with more than \$10 million in assets whose securities are held by more than 500 owners must file annual and other periodic reports. Publicly held companies are required to file documents with the SEC which include:

- Registration statements for newly-offered securities
- Annual and quarterly filings (Forms 10-K and 10-Q)
- Proxy materials sent to shareholders before an annual meeting

- Annual reports to shareholders
- Documents concerning tender offers (a tender offer is an offer to buy a large number of shares of a corporation, usually at a premium above the current market price)
- Filings related to mergers and acquisitions

Corporate Repurchase

When a corporation elects to repurchase some of its own securities. This reduces the Common Shares Issued and Outstanding. Typically, used to improve the valuation of the company's common securities outstanding as well as the Earnings Per Share (EPS).

Correction to IRR

An IRR type used when corrections need to be made to an original IRR that has already been submitted to the Office of Personnel Management (OPM). Federal employees covered by the CSRS retirement plan require SF-2806-1. Federal employees covered by the FERS retirement plan require SF-3101. A Correction IRR is also used if original retirement deductions were over-reported. See also Individual Retirement Record (IRR).

Correspondence Customer

A customer to whom all correspondence (statements) is addressed, often a corporate customer receiving correspondence for associated child customers.

Cost Accounting

A method where business costs are accumulated and distributed to products, processes, or discrete undertakings on an equitable basis. There are a variety of cost accounting methods, but they all share the same basic functions: classifying costs, recording costs, allocating costs to products or activities, summarizing and reporting costs to management. Cost accounting requirements and financial accounting requirements are not necessarily synonymous.

Cost Assignment

Resources assigned to cost objects or activities.

Cost Basis

Typically, this refers to the original price of an asset used in determining capital gains. However, in the case of death of an optionee, the appraised value of the asset at the time of death is the cost basis.

Cost Center

A Time and Labor Business Unit, in which all related costs attributable to some center within a business (such as an activity, an organization, or a program), are segregated for accounting or reimbursement purposes.

Cost Element

See **Inventory Cost Element** and **Manufacturing Cost Element**.

Cost Flow

Determines how depletions will occur for purposes of costing a transaction. Cost flows available include Specific Lot ID, Specific Serial ID, FIFO, and LIFO.

Cost Objects

Cost objects represent cost information about products, customers, and channels. They are the final results of the activities performed by your business, representing the focal point of costing and profitability analysis. Examples are products, customers and channels. They are the final results of the activities performed by your business. Your model's resources and activities are linked to the cost objects. They are often the focal point of profitability analysis.

Cost of Capital

An attribute used to describe the behavior of a particular cost object. A primary cost object is typically the main focus of the activity-based management analysis. This may be a product, customer or channel that you wish to calculate cost for. A support cost object may be used in a similar manner but may be further allocated to other support cost objects or primary cost objects.

Cost Of Living Allowance (COLA), Non-Foreign

A cost-of-living allowance payable to an employee at a location in a non-foreign area where living costs are substantially higher than those in the Washington, DC area.

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 Profile Group

A grouping of items for the purpose of costing transactions and valuing inventory for a given book. Assigning an item to a cost profile group determines the books used by the item when accounting for that item.

Cost Roll-up

A process for calculating item costs. Cost roll-up provides a summation of all of the costs associated with the bill of material structure and the routing used in producing the item.

Cost Row

A cost transaction and amount for a set of ChartFields.

Cost Template

A collection of cost components that you can apply to a group of purchased items.

Cost Type

A user-defined method of categorizing item costs in Manufacturing for simulations and what-if analysis. Examples of cost types include current costs (which reflect the item's current bill of material or routing), proposed costs (which could be used in preparation for the next standard cost period), or activity-based costs (which include costs for items that consume a given activity).

Cost Version Type

A combination of cost types and cost versions used in cost rollups. Valid values include production (rolls up only manufacturing data and uses only the primary BOM and routing, each with a code of 1), engineering (can roll up with either manufacturing or engineering data, with any combination of BOM/routing codes), or simulation (only rolls up with manufacturing data, but can use any combination of BOM/routing codes).

Count Grade

A user-defined evaluation of a counting event.

Count Point

A predefined step on a routing or operation list where you can gather operation completion information. You define the appropriate points on the routing, record completions at these count points, and the system automatically backflushes the prior operations. This is only used on production IDs.

Counts

Count elements allow you to count the number of days or hours from a specific period of time. Counts are used primarily during proration calculations, but can potentially be utilized in other situations as well.

Court-Ordered Benefits Coverage

As prescribed in Title 5, United States Code and Title 5, Code of Federal Regulations, court orders that stipulate that an employee must continue or begin the coverage features for all employee benefits must be enforced. Federal employees are mandated by court orders to continue covering or begin covering their former spouses and/or children under their federal employee benefit programs (health, life, and thrift savings).

Court-Ordered Garnishments

As prescribed in Title 5, United States Code and Title 5, Code of Federal Regulations, court orders enforcing child support, alimony, or collection of commercial indebtedness are served on the appropriate entity within the Federal agency and implemented as offsets against the employee's salary.

Coverage

An employee's chosen benefit plan and coverage level; that is, what sort of benefit is provided as well as the value.

CPAM (Caisse Primaire d'Assurance Maladie)

In France, CPAMs are the local social security offices that manage health coverage for French workers. CPAMs are regulated and established by the French government. If you're managing a French workforce you'll need to identify and track the CPAM offices that impact your enterprise.

CRAM (Caisse Régionale d'Assurance Maladie)

In France, CRAM is the regional social security body which oversees the running of CPAMs. CRAM offices work with companies to both prevent and compensate workers for industrial injury.

Create Date

The date that you extracted a deduction or offset to PeopleSoft Deduction Management or created a split deduction.

Create Missing Items

In PeopleSoft Demand Planning and Inventory Planning, a feature that enables automatic system generation of master records that don't exist in the system.

Created Time

Time collecting device time or elapsed time generated by the system based on the time reporter's schedule (see Time Administration)

Creating Time

The preliminary generation of time segments as close as possible to their likely values when you officially report time—so that the information on the time records is as fresh and current as possible. The system shows you time that has already been created, rather than you having to create it “on the fly” when you come in to report. The process fills in reporting day gaps as defined by work schedules.

Credit Analyst

A required field used in PeopleSoft Receivables, Billing, Order Management, and Deduction Management when working with items. Each item must be assigned to a credit analyst. If no credit analyst is assigned to an item, the credit analyst assigned to the customer is used as the default.

Credit Risk Spreads

In the financial services industry, the additional charge to a risk-free interest rate, based on a riskier credit rating.

Credits

See Flexible Credits.

CREF

Acronym for Content Reference.

Crew Reporting

A Time and Labor process that enables you to report the earnings which consist of one or several time reporting codes and associated quantities of hours, amounts, or units, and task information for one date under report for a user-defined crew. The system transforms the information into instances of daily time for each crew member for the entered date.

Critical Success Factors (CSFs)

In PeopleSoft Balanced Scorecard, things that an organization must do well or excel at to achieve its goals. One or more key factors or objectives that must be accomplished for a particular strategic thrust. Key Performance Indicators are attached to CSFs.

CRM Warehouse

See Warehouses.

Cross Border Walker

This term is used in Europe for an employee who lives near a border in one country and works in another country. Such employees are subject to different tax and social security rules.

Cross-Plan Validation

The process by which the PeopleSoft Benefits Administration determines enrollment prerequisites for benefit plans. You can define four types of cross-plan validation prerequisites: prerequisites based on plan types, benefit plans, dependent enrollments, and coverage percentage limits for Life and AD/D plans.

Cross-View Reconciliation

In PeopleSoft Demand Planning, a process that enables the balancing of forecasts between selected levels of related views with the same **Forecast Item** key. The process is used when adjustments have been made to a working view and are then required in a related view.

Cube

See **Multidimensional Database (MDDB)**.

Cube View

In PeopleSoft Demand Planning, defines the user's own view of a forecast. The parent working view and dimensions determine what forecast data is included and how aggregates are formed.

Cumulative Tax Method

A payroll tax calculation method that adds together year-to-date earnings and earnings for the current pay period, then annualizes the result before calculating tax. This method is useful when Payrolls vary greatly in amounts from pay period to pay period, such as in the case of sales commissions.

Currency Calendar

In the financial services industry, business calendars for markets outside the organization's domestic operations that reflect the foreign markets' holiday schedules.

Currency Conversion Engine

A PeopleSoft Enterprise Warehouse Engine that processes financial information in multiple currencies.

Current Period

The earliest pay period for which the close date has not passed (see Time Reporting).

Current Period (Time and Labor)

In Time and Labor, the employee's current pay period which will be determined via the employee's Pay Group affiliation. Although there can be only one definition of Current Period per installation, the user can change it manually.

Current View

A reporting screen in Time and Labor whose effective date is within the date boundaries of an employee's current pay period, and for which pay has not yet been confirmed. A *Future Time Reporting Transaction* is one that has an effective date after the last day of the employee's current pay period. An *Historical Time Reporting Transaction* is one that has an effective date before the first day of the employee's current pay period.

Current Year

A period for event maintenance processing.

Curve Generator

A supporting module (common to financial services industry applications) that enables you to construct curves used to determine appropriate interest rates for given maturities and / or time periods. You can import market data from outside sources such as Bloomberg, upload the data from a spreadsheet, or manually enter the data. You can then build configured curves from segments or combinations of other curves.

CUSIP Number

A nine digit alphanumeric number associated with issuers' securities. CUSIP (Committee on Uniform Securities Identification Procedures). A uniform numbering system widely used to identify specific securities and their issuers.

Custom Statement

A user-created logical or mathematical expression that determines information about an employee in PeopleSoft Pension Administration. Custom Statements commonly define employee groups and benefit formulas.

Customer Inquiry

A window containing options to review customer balances, aging, history, items, actions, and conversations.

Customer Scorecard

See PeopleSoft Customer Scorecard.

Customer Tree

A user-defined graphical representation of your current sales organization. A customer tree is used to establish and distribute funds and to determine authority levels for promotional activities.

Cut Session

Cut sessions are a means of dividing a course session. You use cut sessions where a course session does not run on consecutive days from start to finish, or if there are multiple instructors or locations. Each cut session has its own start/end date, location, and instructor. For example, if you have a course that runs for two days a week for a month, you would divide the course session into four cut sessions, each of which is two days long.

Cycle Count

A manual counting event that does not cover an entire inventory business unit. Usually includes every item (and lot, if applicable) in a location or family.

Cycle Interval

The number of days between cycle counts.

Cycle Procedures

Inventory planning tasks that need to be performed on a regular basis to ensure an up-to-date **Inventory Policy**. The tasks can be performed either at the end of a processing period or within the period, and should always be performed if the forecast or **Control Group** or **Policy Item** parameters change. Tasks include generating a policy and reviewing **Work Queue** messages.

D

DAT file

A text file (input.dat) used with the Verity search engine that contains all of the information from documents that will be searchable but not returned in the results list.

Data Elements

Data elements, at their most simple level, define a subset of data and the rules by which to group it.

For PeopleSoft Balanced Scorecard, data elements are used as the basis for key performance indicators, and as target values for Key Performance Indicator (KPI) objects.

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

Data Entry Access List

Used to present a concise list of often-performed data entry tasks to a user. You can assign multiple control plans to a single data entry access list.

Data Extract

A report that creates a file used to transmit data to a third party on magnetic media. There is no meaningful printed output for this type of report.

Data Loader

Data Loader is a PeopleSoft Enterprise Warehouse utility that moves data from the Operational Data Store staging area to either the ODS reporting area or the Data Warehouse. The Data Loader utility is made up of several pages that allow you to enter Metadata to define your source and target records and your transformation rules and then perform the load by running an Application Engine.

Data Loader Map

Defines how to extract data from the Operational Data Store (ODS), transform it, and load to a Target Table. The target table can reside in the warehouse or the ODS layer.

Data Manager

A PeopleSoft Enterprise Warehouse engine that distributes revenue, expense, analytical application engine results, statistical quantities and other measures across business units, departments, products, customers and channels—any field or logical group in the chart of accounts. You can define a number of types and options within this engine. It is also used as a means of posting to the Performance Ledger.

Data Manager Rules

In the PeopleSoft Enterprise Warehouse, Data Manager rules use Constraints to specify the source as well as the target tables for moving, aggregating, or multidimensionalizing your engine output. Rules use Data Manager methods to enrich the PeopleSoft Enterprise Warehouse data.

See Data Manager Methods.

Data Manager Methods

There are several methods: Copy, GL Mapper, Prorata, Spread Even, and Tree Aggregation. Each method enables you to move and/or enrich engine output.

Data Mart

A Data Mart is a data structure that uses a central fact table and related dimension tables to generate a “relational cube” or directly generate an Insight report.

Data Mart Builder

The Data Mart Builder is a multiple Application Engine (AE) process, that is, a framework of procedural programs, that creates a Data Mart.

DataMaps

Information that builds upon the data captured in the TableMap records. DataMaps enable you to define a logical view of the physical PeopleSoft Enterprise Warehouse tables. DataMaps bring together information from many different tables and fields and define it all as one entity or table.

Data Row

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

DataSet

DataSets are used as input for various engines and processes, for instance, the Analytic Forecasting component, the Data Manager, user defined functions, drivers in Activity-Based Management, and data elements in the Key Performance Indicator Manager. DataSets provide a user defined set of information to the engines. DataSets use Constraints to restrict used columns and restrict returned rows. Each DataSet is created by a process specific setup. However, the underlying logic is the same, enabling you to more easily understand the functional aspects of the process.

Data Warehouse

A large database containing data summarized from one or more transactional systems, optimized to support the analysis needs of the enterprise. An ideal data warehouse contains all the data necessary to make business decisions. Users analyze the data in the warehouse using Online Analytical Processing (OLAP) tools and ad hoc query/reporting tools. An increasing

number of organizations have "virtual" data warehouses, where the data warehouse is not one physical database, but rather a collection of specialized (and distributed) data marts.

See also PeopleSoft Enterprise Warehouse.

Data Warehouse Tables

Data Warehouse tables act as the portal for getting data into the PeopleSoft Enterprise Warehouse from PeopleSoft, OLTP applications or other "outside" sources. These tables are used:

- As targets for loading operational data.
- For error detection and handling
- For data validation.
- For aggregation.

Database Alias

The PeopleSoft Pension Administration utility that looks up employee data.

Dataset

A file containing data to be analyzed by the Quality Server program. The dataset is similar in content to a spreadsheet.

In PeopleSoft Planning, a file that stores schedule information such as tasks, resources, calendars, and so on.

Date

See **Accounting Date** **Transaction Date** or **Effective Date**.

Date

If you want to either include a date in a calculation, or determine a new date by taking a starting date and either adding or subtracting a period of time to come up with another date, you use a date element.

Date Classified

Date the Position Description is approved by Management/Position Management.

Date Eligible To Retire

Date an employee is eligible to optionally retire based on the combination of age and service that meets legal requirements.

Date Under Report

The date (day) in PeopleSoft Time and Labor for which time is being reported. The Date Under Report does not have to equal today's date.

Day Breaker

Customer defined time that is used to determine when one day becomes the next. It's used to determine the "logical" date of a punch. (See Understanding Workgroups.)

Days Supply

In PeopleSoft Inventory Planning, a method that can be used with several types of **Inventory Policy**. Using this method, a specific number of days of supply for an item should be used to calculate the item's inventory policy.

Deal Type

PeopleSoft Treasury has categorized deals into several basic deal types from which you can choose when defining an instrument.

Death Coverage

The PeopleSoft Pension Administration function that determines the factor used to reduce an employee's benefit when the plan charges for PRSA coverage.

Decompressed Split

In PeopleSoft Demand Planning, a function for returning a compressed split database to its original form. See also Compressed Split.

Deduction

Any amount taken from an employee's pay check each pay period. Deductions may include health or medical benefits, union dues, and so on. See also Benefit Deduction and General Deduction.

Deduction Date

The as of date for the deduction item in PeopleSoft Receivables.

Deduction Item

An individual item that you created in receivables and is an open receivable on the customers account due to a deduction that they took in a payment for a receivable item.

Deduction Reason

Code that describes the type of deduction. When assigned to a write-off resolution, it determines what accounting entries to create.

Deduction Specialist

The individual responsible for tracking and resolving deductions in PeopleSoft Deduction Management.

Deduction Subset

A group of deductions selected from a company's standard set of deductions. Deduction subsets minimize data entry time in special processing situations such as bonus check runs.

Default Mode (DM) model

In the financial services industry, an approach used by financial institutions to predict a decline in portfolio value. Only two outcomes are considered – default or non-default. If the debt does not default, there is no change in the value. If the debt does default, then the loss is calculated as the difference between what was contractually owed and the value of any collateral recovered.

Defection Analysis

In PeopleSoft Workforce Analytics, the identification of employees who are likely to leave the organization based on predefined assessment criteria.

Deferred Compensation

Compensation payments that are payable to an individual in the future such as pension plan payments, annuities, stock awards and profit sharing. Note: Profit sharing can be considered direct pay if paid out in cash on a periodic basis or deferred pay if cumulative with the intention of payment in the long-term future.

Deferred Vesting

The adjustment made to the original option's vesting schedule that pushes the vesting into the future.

Defined Benefit Plan (DB Plan)

A retirement income plan (usually called a pension plan) where the employee's benefit is definitely determinable based on a plan-specified benefit formula.

Definition or Function Definition

The parameters for any of PeopleSoft Pension Administration's nineteen core functions. A definition has to be explicitly associated with an employee Group Definition before it can be applied.

Dekit

The ability to return material issued in kits to inventory. This is used when entire kits need to be returned; individual components are handled through kit issues/returns.

Delete Non-Matching Items

In PeopleSoft Inventory Planning, an option used in the Generation process to delete Inventory Planning items that don't have corresponding items in Demand Planning. The item deletion occurs when the system generates the policy.

Delta

When retroactive processing occurs for a given payee, the system recalculates each element generated for the payee. The system compares the recalculated results to the original results. The difference between these results is typically referred to as the retro "delta." A retro delta can represent either an underpayment or an overpayment that results in an adjustment to the payee's earnings.

Demand

Collection of training requests. This could be an employee demand, a departmental one or a company-wide demand.

Demand Filter Width

In PeopleSoft Demand Planning, specifies the confidence interval within which demand is considered to be reasonable. Actual demand that is outside the confidence interval is automatically filtered and replaced by the value at the edge of the interval. The value is expressed as a percentage.

Demand Filtering

In PeopleSoft Demand Planning, provides a way to detect and highlight unusual demands and forecast errors. If the demand falls outside of a band that is considered reasonable, the system automatically adjusts it to the level of the boundary and logs a message to the **Work Queue**.

Demand Number

The configured product sub-component sequence number.

Demand Planning

In PeopleSoft Activity-Based Management, this type of planning focuses on studying the impact of cost objects and activity volumes.

Demand Priority

The placing of importance on independent demand. The Planning engine uses the demand priority value to determine the order in which you fulfill the demand. You can set a demand priority from 1 to 998 with 1 being the most important level. The priority value of 999 is reserved for the system.

Demand Priority Rules

In PeopleSoft Inventory, a set of rules that will sort demand so the most important demand will have the first opportunity to reserve available inventory. If demand priority rules have

been defined, the Material Reservations process (INPLDMND) sequences orders by priority rank, processing those with the lowest rank value first.

Deplete Cost Method

Determines how you cost a depletions transaction within a book. The deplete cost methods available include Actual, Non-Cost, Perpetual Weighted Average, Periodic Weighted Average, and Value at Current Standard.

Depreciate When in Service

A switch that indicates whether PeopleSoft Asset Management should allocate depreciation as of the date an asset was placed in service. This is valid only in the year the asset was acquired.

Depreciation - Declining Balance

Budgeting calculates this as: Cost minus Accumulated Depreciation divided by Life divided by number of periods per year. It results in a higher depreciation expense in the early years of an asset, which decreases as you near the end of its useful life.

Depreciation - Double Declining Balance

Budgeting calculates this as: Cost minus Accumulated Depreciation multiplied by 2 divided by Life divided by number of periods per year. It results in a higher depreciation expense in the early years of an asset, which decreases as you near the end of its useful life.

Depreciation Methods

The various methods of spreading the acquisition cost across the life of an asset rather than expense the full value of an asset at the time you acquire it. The value of the asset consequently decreases (or depreciates) through time. The four depreciation formulas delivered with PeopleSoft Budgeting include: declining balance, double declining balance, straight line, and sum of the years.

Depreciation - Straight Line

A method of depreciating asset value in equal amounts across the life of the asset. Per-Period Straight-Line depreciation is calculated as the cost of an item minus the salvage value divided by the number of periods to depreciate.

Depreciation - Sum of the Years

A depreciation method equal to the value of the remaining years of life divided by the sum of the years remaining is multiplied by the Net Book Value. This figure is then multiplied by the percent of years to depreciate. This results in a higher depreciation expense in the early years of an asset, which decreases as you near the end of its useful life.

Depromote

In PeopleSoft Demand Planning, the process of making an adjustment to actual demand data that removes the effect of a promotion during a defined period. As a promoted period moves into history, the system creates an adjusted demand entry that is equal to the **Prorated Forecast**.

DeptID

A ChartField that defines departments or administrative offices that have operational, fiscal and/or budgetary responsibility for specific sets of activities.

Derived Metric

The result of a calculation on a report of base metrics.

Detail

A temporary assignment to a different position for a specified period when the employee is expected to return to his/her regular duties at the end of the assignment. This employee is considered for pay and strength count purposes to be permanently occupying his/her regular position. Unless the agency chooses to use an SF50, a detail is documented with an SF52.

Detail Tree

A tree that employs ranges of detail values under each node; you must manually specify the detail values.

DFI ID (Depository Financial Institution ID)

A PeopleSoft Payables bank identifier, consisting of Transit Number, Swift ID, or CHIPS ID.

Dimension

A single element of a business model, such as product, department, or location. Cube Manager uses the term Conforming Dimension.

In terms of data analysis, dimensions can be thought of as criteria, such as time, product, and location, used to pinpoint a particular piece of data. For example, in the retail industry a set of dimensions could be geography, product, time, customer, and vendor. The geography dimension would include company, chain, region, district, and finally store attributes. A dimension is also a column heading on an analysis and reporting template which you can drill through or roll up to the multiple levels.

In PeopleSoft Budgeting, a view option that assists in summarizing the rows of data in line-item budgeting.

A single element of a budgeting model, such as account, product, project, department, or operating unit. In PeopleSoft Budgeting, these dimensions typically represent the ChartFields used by your organization during your budgeting process.

Dimension Table

In the PeopleSoft Enterprise Warehouse, Dimension Tables store additional attributes or data about Facts. Some example dimensions include Customer, Channel, Geography and Product.

Direct Compensation

In PeopleSoft Workforce Analytics, Direct Compensation is payment made to workers in exchange for their contributions to the organization. Direct Compensation is typically categorized as including Cash Compensation and Long-term Variable Compensation.

Cash payments made to workers in exchange for their contributions to the organization. Direct pay is typically categorized as fixed pay (for example, base pay, shift differentials) and variable pay (for example, profit sharing, incentive, bonus). Note: Profit sharing can be considered direct pay if paid out in cash on a periodic basis or deferred pay if cumulative with the intention of payment in the long-term future.

Direct Calculation

Calculate actual and directly assigned dollars.

Direct Cost

In PeopleSoft Workforce Analytics, a direct cost of an activity or a cost object. An example is the salary cost of employees working on a project.

Director

An affiliate of the company who holds a seat on the Board of Directors for the corporation. A Director, generally, is not an employee of the corporation.

Disability and Discrimination Act of 1995

In the United Kingdom this act makes it unlawful to discriminate against individuals on the basis of their disability in relation to recruitment, promotion, training, benefits, terms and conditions of employment, and dismissal.

Disability Rate Code

The desired percentage of disabled persons that should be employed by French employers, as mandated by the French government.

Disbursement View

In PeopleSoft Demand Planning, a **Forecast View** that allows the forecast from a working view to be reported on using an alternate key. Disbursement views are built directly from the working views and inherit many working view attributes, including time period and associated user data definitions, from the parent working view.

Discounted Stock Option

Rights to a stock option at a price less than 100 percent of fair market value at the time of grant.

Discretionary Plan

In PeopleSoft Workforce Analytics, this is a plan for distributing compensation awards that provide managers the ultimate discretion over a pool of money which is either funded based on company, group, or employee performance, or it's budgeted. The discretionary award determination is sometimes guided by a pre-determined percent of the participant's salary, expressed as an opportunity. This figure can then be modified based upon management's perception of actual value created by the group or employee.

Disqualifying Disposition (DD)

When an optionee sells or otherwise disposes of the shares of stock acquired through the exercise of an incentive stock option or through an employee stock purchase plan before the holding period for preferential tax treatment has lapsed.

In the case of Incentive Stock Options, the holding period is one year of the date of exercise and two years of the date of grant. At the time of disposition, the individual recognizes compensation income equal to the difference, if any, between the option price and the fair market value of the corporation's stock on the date of exercise. If the sale price is less than the fair market value of the stock on the date of exercise, the compensation income is limited to the total sales price less the total option price, less any fees.

In the case of purchases through an employee stock purchase plan, the holding period is one year from the purchase date and two years from the enrollment date. Compensation income in a disqualifying disposition is equal to the difference between the total fair market value on the purchase date and the total purchase price.

Distribution

Provide a repository of time and associated estimated and actual allocated labor costs to other systems

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

Distribution Network

A distribution network is a prioritized list of Inventory business units (IBUs). When a customer orders a product, the system uses this network to determine which warehouse the stock ships from.

Distribution Profile

A definition of ChartField distributions assigned for compensation costs. A distribution profile can be used to set up defaults for how the system should distribute costs associated with a position's salary, benefits, and earnings. PeopleSoft Budgeting-specific.

Distribution Rule

You use distribution rules to determine the order in which the system searches for matches against the distribution sets matrix when sales orders are entered.

Distribution Set

Distribution Sets assign account distribution information to combinations of defining elements used on sales orders.

Distribution Type

An identifier that defines one of the different transactions that move an item into or out of an inventory business unit. Distribution types are used to create debit and credit transactions to the general ledger via the Journal Generator.

Dividend

Distribution of earnings back to shareholders, prorated by the class of security and paid typically in the form of money or stock. The amount of a dividend is decided by the Board of Directors and is usually paid quarterly.

Document Management

The process through which a user has complete control of document version including the ability to view, query, and edit documents in a secure vault. Document management enables you to seamlessly perform online document queries and view documents directly, launching them from within PeopleSoft applications. You can associate pertinent documents with engineering change requests (ECR), engineering change orders (ECO), item revisions, bills of material, manufacturing and engineering routings, production component lists, and production operation lists.

Document Sequence Number

A value that the PeopleSoft system assigns to a document (such as an invoice, voucher, or journal) when you create a document for a business unit that you have enabled for document sequencing. The system determines the number by the values of the business unit, accounting date, and document type.

Document Sequencing

A flexible method that sequentially numbers the financial transactions (for example, bills, purchase orders, invoices, and payments) in your system for the purpose of statutory reporting and tracking of commercial transaction activity. Document sequencing requires that you classify all financial transactions into three transaction types—journal type, journal code, and document type—and that within each transaction type, all documents you enter are numbered sequentially. When you create a document (such as an invoice, voucher, or journal), the PeopleSoft system assigns a document sequence number to that document.

Document Type

The final level of three categories for defining a financial transaction (or document), necessary when using document sequencing. It represents the business purpose of a financial transaction, such as domestic customer invoice or customer credit memo. Document type is within one and only one journal code; journal code is within one and only one journal type. Document type is the only required category, because the values of the other two categories can be derived from document type.

Dollar Tolerance

In PeopleSoft Inventory, the acceptable cost difference between expected cycle count quantities and actual quantities counted. This value allows a margin of error for an item during cycle count reconciliation based on item cost.

Domestic Relations Order (DRO)

A preliminary version of a court order (usually stemming from a divorce settlement) ordering a division of a participant's pension benefits. The order is not in effect until it is determined to be "qualified" by virtue of meeting certain requirement. At that point it becomes a Qualified Domestic Relations Order, or QDRO.

Double Byte Characters

If you're 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].

Draft Worksheet

A work space used in PeopleSoft Receivables to track a draft through its processing life cycle.

Drill-Back Calculation

Assigns indirect dollars and Drill-Back calculations. Also, this picks-up all costs in the Calculations Detail (CALC_DETAIL_F00) that was assigned during direct calculations.

Drill Down

The ability to go down to the next level of detail in a set of data. For instance, if you're looking at an expense figure for a division, you can drill down to the expenses for each department in the division.

Drill Down

The ability to go down to the next level of detail in a set of data. For instance, if you're looking at an expense figure for a division, you can drill down to the expenses for each department in the division.

Driver Lookup Table

Tables associated with a driver that enable different rates and amounts unique to a budget center.

Drivers

In PeopleSoft Activity Based Management, drivers are a means of assigning dollar amounts from resources, activities, and cost objects to other resources, activities, and cost objects throughout the model in PeopleSoft Activity-Based Management. Drivers can also be assigned across business units. There are different types of driver categories (transactional, duration, and intensity), and different ways of specifying how those dollar amounts are calculated (amount, percentage, spread even, and direct), as well as different ways that dollar amounts are assigned (depending on assignment type and object type).

In PeopleSoft Business Planning, a driver can be defined as a set of values that are used as an input to another process. In this context, a driver should be interpreted as a projection of external factors and other indicators. The user can then define a relationship between the driver and a financial result. The driver values and the defined relationship then combine to produce a projection of the financial result. For example, a projection of the number of square feet used (driver) and the price per square foot paid in rent (driver) can combine to produce a projection of rent expense (financial result).

DRO

See Domestic Relations Order.

DSS (Decision Support System)

A DSS is a workstation-based analysis and reporting system, typically aimed at analysts and line managers. OLAP tools provide a powerful DSS.

Duration

In PeopleSoft Pension Administration, the utility that calculates the length of time between two dates.

Duration [Global Payroll]

An element type that calculates a period of time between two dates. For example, if you want to determine a payee's age, you can calculate the duration between his birth date and the calendar period end date.

Dynamic Group

A group in Time and Labor that enables you to establish criteria or attributes for a group of employees. All employees who fit this criteria at processing time belong to the group.

Dynamic Tree

A tree that takes its detail values—*Dynamic Details*—directly from a table in the database, rather than from a range of values entered by the user.

Dynamic Views

In PeopleSoft Demand Planning, a **Forecast View** that allows interaction with the forecast data using an alternate key structure. By using dynamic views, you streamline the working view and can complete the working-view design without having to anticipate all conceivable adjustments.

E**Earliest Change Date**

Determines both the range of dates and the amount of data that will be processed for each time reporter (see Batch Processing)

Early Punch

A punch that is more than the predefined number of hours/minutes before a scheduled punch where a time reporter is warned

Early Retirement Date (ERD)

A retirement date earlier than a plan-specified “normal” retirement date. Employees usually must meet age and/or service requirements to be eligible for early retirement, and early retirement benefits are often reduced to compensate for the longer duration of payments.

Early Retirement Factor

The reduction made to an employee's benefit if the employee elects for early retirement.

Early Warning

In commitment control, warning of possible future budget exceptions. You can specify that you are to receive a warning when commitments and expenditures reach a predetermined percentage of budget. For example, you can instruct the system to let you know when commitments and expenditures reach 50%, 80%, or some other percent of a budget.

Early/Late Adjustments

The PeopleSoft Pension Administration function that calculates early retirement factors or late retirement factors.

Earning Group

Part of a group of defaults assigned to job codes. Earnings group may include non-salaried items such as holidays and bonus pay dependent on individual company parameters.

Earnings

The amount owed to an employee based on salary, hours worked, or other calculation routines, plus other types of compensation and holiday, vacation, and bonus pay.

Earnings [Global Payroll]

An element type that defines the different types of compensation that are added to a person's pay. Examples include salary, commission, bonuses, and retirement pay.

Earnings Accrual Class

Categorizes a set of accruable earnings.

Earnings Code

Codes that represent the various types of earnings such as regular, overtime or leave.

Earnings Per Share (EPS)

The portion of a company's profit allocated to each outstanding share of common stock. Net income (reported or estimated) for a period of time is divided by the total number of shares outstanding during that period.

Earnings Type

An abbreviated and encrypted set of business instructions containing compensation instructions. Earnings Type may also contain Benefit Entitlement and Administration instructions, taxation instructions, Financial Accounting instructions, Organizational Administration instructions, work group and labor affiliation instructions, and other instructions.

Economic Loss

In Funds Transfer Pricing, this refers to the break fund economic loss, calculated by applying the theoretical value of the interest rate differential (IRD) against a cash flow stream, based on the amount of the prepayment or cancelled draw-down.

Economic Value Added

In the financial services industry, Economic Value Added is a financial metric that factors into the measurement of an activity's profitability the cost of economic capital assigned to that activity.

EDGAR (Electronic Data Gathering, Analysis, and Retrieval)

An electronic system implemented by the SEC that enables companies to file documents in conjunction with disclosure requirements mandated by the SEC.

EDI Agent

Used in EDI processing, the inbound EDI Agent loads trading partner data (flat files) into the PeopleSoft database using transaction, map, and trading partner definitions set up using EDI Manager. The outbound EDI Agent extracts information from the PeopleSoft database and generates data files that can then be processed for transmission to a trading partner.

EDI Manager

A suite of online pages used to define transaction sets, trading partner profiles, and translation maps for EDI transactions.

Edit Table

A table on 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.

EEO Company Code

In the United States companies are assigned this federal code for EEO and VETS100 reporting.

Effective Date

A method of dating information in your system. 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.

Effective Date

A method of dating information in your system. 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. The Effective Date usually defaults to your system's current date.

Effective Periods

In PeopleSoft Demand Planning, the number of periods of historical demand used in the Model Reset process. The value can be used to exclude older, possibly unrepresentative historical demand data from model **Optimization**.

Effective Sequence

A system-generated number assigned to distinguish between two job entries with the same effective date.

Effective Tax Rate

The ratio of income tax paid over gross income, showing the percentage of income actually paid in taxes.

Effectivity Date

The date on which a component can be added or substituted in production, typically specified on an item's bill of material.

EIS (Executive Information System)

An EIS is a workstation-based analysis and reporting system for executives. An EIS provides a higher-level view of the data than a DSS, and typically requires less knowledge about the underlying transactional systems. OLAP tools provide a powerful EIS.

Elapsed Schedule

A method of scheduling a time reporter's time that is based on TRC and duration. This method can be used for scheduling of elapsed time reporters (see Scheduling.)

Elapsed Time

Reporting non-clock time in increments of hours or partial hours (see Managing Time / Understanding Time Reporting).

Elapsed Time Service

A method of calculating a period of service that uses only the start and end dates of the period to determine the amount of service. Hours worked or other measures of the actual work performed during the period are not taken into account.

Electronic Certification System (ECS)

An automated Payment Voucher authorized by the Certifying Officer for use within the Treasury Department, Financial Management Service's financial system. PeopleSoft provides a method to record and generate data files for on- and off-cycle processed payments.

Element

In PeopleSoft Global Payroll, an element refers to both primary elements and supporting elements. Primary elements are comprised of earnings, deductions, absence entitlements, and absence take elements. Supporting elements are element components that are combined to create primary elements.

In PeopleSoft Enterprise Performance Management, elements are used to create a Profile. An Element can be one or more columns of data in an Enterprise Warehouse table, associated with a single dimension (for example, Customer, Product, Department, or Channel). An Element can also be KPI, Population, subscription data from a third party, preexisting Profiles, and data mining scores.

Element Group

Element Group identifies a group of elements to provide eligibility. You can then use this as a notational shortcut—instead of having to list each element, you can use the element group name. Element Group's expedite the process of manipulating earnings and deductions.

Element Name

Name assigned by the user for data fields, rules, formulas, and tables. For example, the names you give to new rules, elements, or objects.

Element Segment

When an element changes mid-period, requiring the affected element (and perhaps a subset of other elements) to be calculated multiple times on either side of the date on which the change takes place, element segmentation is used. Unlike period segmentation, the system segments only the elements you select, and creates separate result columns only for the specified elements. In element segmentation, there is only one gross-to-net result set.

Eligibility Group

Eligibility groups define the possible earnings, deduction, absence entitlement, and absence take elements that a payee might be eligible to receive. This enables you to group payees so as to assign eligibility for certain pay elements.

Eligibility Rule

PeopleSoft Benefits Administration uses eligibility rules during Benefits Administration processing to determine which benefit programs and options an employee is eligible for. Eligibility rules are closely associated with event rules: they determine what options an employee can *have*, while event rules determine which of those options an employee will actually be able to *choose*.

Elimination Set

See **Consolidations-Elimination Set**.

Email Template

Pre-defined parameters that establish automatic email generation during budget submittal, rejection, publishing, and/or targeting.

Employee

An individual employed by an organization and administered as an employee in the PeopleSoft Human Resources system.

Employee Accounts

The PeopleSoft Pension Administration function that tracks employee contributions to a pension plan.

Employee ID

A unique identification code for an individual associated with your organization.

Employee Paid Benefit

The portion of a pension benefit funded by the employee's own contributions to the pension plan. Also, the PeopleSoft Pension Administration function that determines this amount.

Employee Profile

This PeopleSoft Activity-Based Management feature enables time and labor information to be part of an analysis.

Employee Stock Purchase Plan

A type of statutory stock option plan through which employers grant options to their employees in order to provide them with additional forms of compensation.

Employee Survey

In PeopleSoft Workforce Analytics, a method for capturing information about the activities performed by a given employee as well as the amount of time they spend performing each activity to perform activity-based management.

Employee Survey Report

The Employee Survey Report is an annual regulatory report that the French government requires from employers with more than 200 employees. In French it is called "Le Bilan Social". The report is communicated to both labor unions and the government. It provides a snapshot view of the company over the past 3 years for about 200 indicators.

Employee Training Cost

Amount budgeted to pay for students' salaries while on training courses.

Employer Identification Number (EIN)

In the United States a company is typically defined as a business enterprise that has a unique federal Employer Identification Number (EIN) for payroll tax reporting purposes.

Employer's Liability Insurance Associations (Berufsgenossenschaften)

Social Insurance in Germany is maintained and administered by private organizations that act as employer's liability insurance associations. Employers pay out premiums to these associations, who administer and pay out funds to workers who are injured on the job.

Employment Cost Index (ECI) Adjustment

Annual increase to wages established/permitted by statute.

Employment Equity Computerized Reporting System (EECRS)

Canadian companies are required to report to the Federal Government on employment equity. PeopleSoft Human Resources contains the Canadian Employment Equity report (PER101CN), which creates a data interface file to the federal government's Employment Equity Computerized Reporting System (EECRS).

Employment Record Number (EMPL RCD#)

A field in PeopleSoft Human Resources Management Systems and PeopleSoft Workforce Analytics that indicates an employee has multiple job records in the system. A numeric value (0, 1, 2) is assigned to each job as a way to uniquely identify that job record.

Encumbrance

A claim against funds. It is a projection of future expenses based on the situation, as we know it today. Encumbering funds is not the same as spending them or even guaranteeing that you will spend them. It just means that if the situation as it exists today does not change, you will spend all of those funds by the end of the fiscal year.

Engineering Bill of Material (EBOM)

A listing of all the parts, raw materials, and subassemblies that form the basis of all item and product structures. EBOMs differ from MBOMs (Manufacturing Bills of Material) in that they are not visible within Production Planning or Production Management and are isolated from Manufacturing.

Engineering Change Order (ECO)

A revision to a blueprint or design, released by engineering to modify or correct a part and/or bill of material. PeopleSoft Engineering uses ECOs to manage and document required assembly and component changes.

Engineering Change Request (ECR)

A document that allows you to request manufacturing process improvements and report product defects directly to the engineering department. When workflow is enabled, ECRs can also be routed for review and approval, after which they change into ECOs.

Engineering Cost Version

The process of generating cost versions for new and modified configurations based on engineering bills of material (EBOM) and costing data.

Engineering Workbench

An engineering environment, separate from production, consisting of engineering bills of material (EBOM), engineering change requests, engineering change orders, EBOM cost roll-up capability, online BOM comparisons, and seamless integration to a document management vault.

Engineering Workbench

An engineering environment, separate from production, consisting of engineering bills of material (EBOM), engineering routings, engineering change requests, engineering change orders, EBOM cost roll-up capability, online BOM comparisons, and seamless integration to a document management vault.

Enterprise

In PeopleSoft Time and Labor, all of the business units of the installation site.

Enterprise Performance Management (EPM)

See PeopleSoft Enterprise Performance Management

Enterprise Portal

The PeopleSoft Enterprise Portal is a separate product offering purchased independently of any other PeopleSoft applications. It can be used with or without any PeopleSoft application. It can be used as a standalone corporate portal that does not access PeopleSoft data at all.

Enterprise Resource Planning (ERP)

The encompassing term for all the transaction-oriented database applications an organization deploys across its business enterprise. The term includes financial, manufacturing and supply chain, human resources, and payroll applications, among others.

Enterprise Warehouse (EW)

See PeopleSoft Enterprise Warehouse

Entry Authority

Authorization granted by employees to specific user IDs for entering expense data on their behalf.

Entry Currency

The currency used to enter budget data.

Entry Event

An automated process that generates multiple debits and credits resulting from single transactions, to produce standard supplemental accounting entries.

Entry Event Code

Designation of an Entry Event; an identifier or label.

Entry Event Generator

A mechanism that generates standard, supplemental accounting entries based on Entry Event codes.

Entry Event Process

An accounting transaction. Entry Event processes combine to form Entry Events. For example, requisition posting is a Purchasing process, and cash clearing is a Payables process. Each process can involve one or several Entry Event Steps.

Entry Event Step

Part of an accounting transaction. For example, the BUDG process includes these steps, among others: prepare allotment budgets, prepare organization budgets, and prepare revenue estimates. Entry Event steps combine to form Entry Event processes.

Entry On Duty Date (EOD)

Date that indicates when an employee started to work at his/her current agency.

Entry Type

Any activity that creates or updates an item.

EPM (Enterprise Performance Management)

See PeopleSoft Enterprise Performance Management

Equal Employment Opportunity Commission (EEOC)

In the United States the EEOC requires that most companies file one or more reports from a series named EEO-1 through EEO-9. These reports include counts by federal employment categories of male and female employees in certain ethnic groups.

Equitization

A process that enables parent companies to calculate the net income of subsidiaries on a monthly basis and adjust that amount to increase the investment amount and equity income amount before performing consolidations. For organizations with complicated parent/subsidiary business unit relationships, this automated process reduces the process time and reduces the possibility for errors.

Equity Increase

In PeopleSoft Workforce Analytics, Equity Increases are base pay increases granted to bring an employee's pay up to some internally specified standard for your organization.

Equivalent Standard Deviation

In PeopleSoft Demand Planning, a **Standard Deviation** developed during the Model Reset process that enables you to compare standard deviations from different models. The deviation is calculated by multiplying the **Model Equivalency Factors** defined on the **Control Group** by the standard deviation.

ERISA (Employee Retirement Income Security Act of 1974)

The U.S. Federal legislation enacted to prevent abuses of employee pension rights by employers.

Error Exception

A transaction that is stopped because the budget limits would be exceeded if it continued. For the transaction to proceed, action must be taken, such as canceling or reducing the transaction amount, increasing the budget amount, overriding the budget limits, or transferring available funds from another budget.

Error Ratio

The ratio of the statistical Standard Deviation to the base component that gives an indication of the accuracy of the forecast. The ratio is presented in the PeopleSoft Demand Planning Audit and Accuracy Review and is calculated during the **Forecast Calculation Process**.

Estimated Gross

Estimated labor cost associated with reported time (see Managing Time, Understanding time Reporting Codes).

Estimated Shipments

A group of shipment schedules used to manage sales order requested shipment data and actual shipment data. Used in conjunction with weight and volume pricing and freight charge calculations.

Ethnic Code

The Federal Office of Management and Budget (OMB) racial and ethnic census categories used for classifying individuals in U.S. Government reports.

ETL (Extract-Transform-Load)

See Extract-Transform-Load.

ETL maps

ETL maps provide rules for importing your source data to the data warehouse tables.

Evaluated Receipts Settlement (ERS)

A PeopleSoft Payables feature that matches receipts against purchase orders and generates vouchers without requiring an invoice.

Evaluations Periods

In PeopleSoft Demand Planning, indicates the number of future periods to use for the calculation of forecast errors. For example, if the number of evaluation periods is two, then the forecast error in April 2001 (after posting demand for April) would be determined by comparing the actual demand for April and the April forecast generated in February 2001. Evaluation periods are set separately for each view.

Event

Events are predefined points either in the Application Processor flow or in the program flow. As each point is encountered, the event fires on each component, triggering any PeopleCode program associated with that component and that event. Examples of events are FieldChange, SavePreChange, OnRouteSubscription, and so on.

Event

Occurrence or happening.

Event Class

An event or type of event that results in a change of benefits eligibility for an employee or dependent. Event classes are prominently used in COBRA and Benefits Administration processing.

Event Maintenance

The process that enables you to manage ongoing enrollments during a plan year. Changes involving maintenance include new hires and re-hires, terminations, family status changes, and changes to benefits eligibility.

Event Rule

Used by PeopleSoft Benefits Administration to determine how events are processed by the system. Event rules look at the benefit plan options an employee is eligible for and determine which options the employee can actually *choose*. Event rules are closely associated with eligibility rules but it is important to note that they are not the same. Event rules *should not* be used to determine eligibility.

Event Trigger

You use triggers to tell the system that when a change takes place to certain data (an event), it should perform an action automatically. When the event occurs, the system writes a line to a trigger table. Then when it's time for the action, the system reads the data from the trigger table and performs the appropriate action.

EW (Enterprise Warehouse)

See PeopleSoft Enterprise Warehouse.

See also PeopleSoft Enterprise Performance Management (EPM).

Expected Losses

In the financial services industry, the amount the institution predicts it will lose in portfolio value. Loan loss reserves are set aside to cover the expected losses.

Excepted Service

As defined by 5 USC 2103, the Excepted Service consists of those civil service positions that are not in the competitive service or Senior Executive Service.

Exception

User or system delivered, defined conditions applied to scheduled, reported or payable time that require audit or review (see Time Management)

Exception Rules

A rule (s) that is applied to scheduled, reported time, and payable time in order to determine conditions which require audit or review (see Time Administration).

Exception Severity

The degree of importance associated with an exception. For example, in exception which is a result of an employee clocking in late may have a Medium severity, while an exception which is a result of an employee not clocking in has a High severity (see Time Management).

Exception Time Reporting

A method of time reporting where only differences to the schedule are provided (see Time Reporting).

Excess Plan

A pension plan where the benefit formula provides an increased benefit for Final Average Earnings above a specified integration level. This compensates for the fact that Social Security benefits are based only on earnings up to a specified maximum.

Exchange Rate Variance

In PeopleSoft Cost Management, the change in currency exchange rate between the time the item is received into inventory and vouchered in Accounts Payable.

In PeopleSoft Payables, a matching feature that compares the exchange rate on the purchase order and the invoice and then copies any variance to PeopleSoft Inventory tables for analysis and accounting purposes.

Exclusive Pricing

Supersedes all pricing structures in effect for customers and products, except **Buying Agreement**, and enables you to drive pricing with a promotional structure. Exclusive pricing can be set up for a specific time frame and associated with particular orders.

Executive Schedule (EX)

Compensation and pay plan used by the Executive Branch of the federal government. Statutory pay limits are derived from several of the pay levels within this plan and imposed on the General Schedule and other existing pay plans throughout the Federal government.

Exercisable

The option shares that are available to the optionee to exercise.

Exercise

The transaction in which an individual purchases or “exercises” the right to purchase the option shares. The IRS refers to the purchase of company stock in an employee stock purchase plan as an exercise.

Exercise Date

The date on which an individual purchases underlying shares from and option grant or transacts a simultaneous purchase and sale of underlying option shares through a cashless exercise and collects option profit in cash or shares.

Exercise Price

The price per share required to exercise a stock option.

Exercise Proceeds

Cash, stock or other recognition received by a company as a result of option exercises, including cash or stock paid by individuals to exercise options and cash company tax savings from deducting non-statutory option profits at exercise.

Expense Location

Geographic area defined to enable the recording, tracking, and reporting of expense activity.

Expense Location Amount

Authorized spending defined for an expense type in a particular expense location and currency.

Expense Location Group

Collection of expense locations based on a common classification such as state, country, or continent.

Expense Type

Means of itemizing various kinds of business expenses. Examples are hotel, dinner, or ground transportation.

Expense Type Edit

User-defined requirement that mandates input of additional data—such as an airline ticket number or number of nights in a hotel—when an expense type is selected in an expense report.

Expense Type Group

Expense types that are classified together for reporting and tracking.

Expensed Item

Non-inventory item which may represent software, manuals, documentation, or any item for which no quantity on hand is maintained, but which can be specified on a bill of material (BOM). Expensed items can only exist as components on a BOM and cannot have a BOM, routing, or production area/item definition.

Expiration

The process by which the outstanding shares of an option cease to be exercisable, generally at the end of the option term. The length of the option term and the date of expiration are established in the Grant Agreement.

Expiration Date

In PeopleSoft Inventory, the date a lot exceeds its Shelf Life and is no longer acceptable for fulfillment or consumption. (Expiration Date = Creation Date + Shelf Life)

Expiration Date

The last day of an option term in which the option is canceled and no longer exercisable.

Expiration Grace Period

When you enter a stock action allows the exercise of the already vested shares as of the action date, the system will calculate the date these shares expire based on the grace period defined on the Stock Action Rules page for that stock action. The system will automatically cancel vested shares not exercised at the end of the expiration grace period.

Express Customer

A customer for whom the minimum necessary information is entered.

Expressions

Expressions enable you to create pseudo-columns made up of mathematical calculations based on actual fields on a table. Since expressions are resolved at run-time, duplicate information is not stored on the database.

Express Order

An order entry shortcut in PeopleSoft eStore and Mobile Order Management whereby the customer populates the shopping cart and goes directly to the order summary to checkout, bypassing any billing or shipment modification screens. Billing and shipping information defaults in as previously entered.

External Data

Data from external sources. For instance, in PeopleSoft Workforce Analytics, external data may include third party salary surveys and benchmark metric surveys.

External Scheme

In the United Kingdom an External Scheme is a vocational training, education and job placement program involving an employee, an employer and the government.

External System

Any system that is not directly compiled with the PeopleTools servers.

Extra Time

Any hours worked outside of an employee's normal (scheduled/shift) hours or days. Extra time may be scheduled in advance of when it is worked, and may be subject to special compensation rules. It may be treated differently than standard time for purposes of Benefit Entitlement and Administration.

Extraction

A reusable query that specifies what information should be retrieved from the Quality database.

Extract-Transform-Load (ETL)

The extraction and transport of data from one server to another remote server. In PeopleSoft budgeting ETL specifically refers to the process by which financial and human resource data is extracted from PeopleSoft Financials and HRMS and transferred to the PeopleSoft Enterprise Warehouse which PeopleSoft Budgeting uses to access and record data transactions. Within PeopleSoft Enterprise Warehouse, data migration typically refers to information moved from outside sources into the Operational Data Store tables.

Extrinsic Rewards

Tangible rewards that can be given to the individual. Typically categorized as financial and non-financial rewards. Financial rewards would include direct compensation, indirect compensation and deferred compensation. Non-financial rewards are provided to the individually and viewed as a benefit by the individual based on the culture of the organization such as the size or location of one's office.

In PeopleSoft Workforce Analytics, tangible rewards given to an individual. Typically categorized as financial and non-financial rewards. Financial rewards would include direct compensation, indirect compensation and deferred compensation. Non-financial rewards are provided to the individually and viewed as a benefit by the individual based on the culture of the organization such as the size or location of one's office.

F

Fact

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.

Fact Table

A fact table is where facts are stored in the PeopleSoft Enterprise Warehouse.

Fair Labor Standards Act (FLSA)

A federal regulation governing several time and labor issues. *FLSA Overtime* requires that all nonexempt employees be paid at a rate of time-and-one-half for all hours over 40 physically worked during a workweek. This requirement may be superseded by state or local laws when the lesser law is to the greater benefit of the employee, or by union contract. An *FSLA Workweek* is a permanently established, regular workweek for a group of employees.

Fair Market Value (FMV)

The price of a company stock based on the current market value as determined by supply and demand, or a valuation method. The stock market sets the fair market value for a public company. For a private company the fair market value is more subjective, but typically determined by financial factors or set by an outside valuation company.

Fair Market Value Tracking Methods

Methods used to track and report trading activity on various exchanges (i.e. NYSE, AMEX, NASDAQ, etc...).

Family Medical Leave Act (FMLA)

A federal regulation that protects health benefits and job restoration for employees who must take a leave from work to care for themselves or family members. FMLA regulations contain provisions regarding employer coverage, employee eligibility and entitlement, notice and certification, continuation of health benefits, and job restoration. PeopleSoft Benefits applications offer FMLA Plans that help employers and employees determine FMLA eligibility and schedule and track FMLA leave requests.

Federal Employee Group Life Insurance Program (FEGLI)

Generally, if the employee has Federal retirement coverage or is on a temporary appointment exceeding one year, he/she is eligible to participate in the FEGLI program. Once eligible, he/she is covered automatically for Basic Life Insurance and premiums will be deducted from gross salary unless coverage is waived within the first period of eligibility. The program offers Basic Insurance coverage and three types of optional coverage: Option A (Standard), Option B (Additional), and Option C (Family).

Federal Employee Pay Comparability Act (FEPCA)

This law provides a structure and methodology to determine and authorize locality-based pay adjustments to Federal employees in order to elevate their basic pay to be commensurate with private sector employees working in the same occupations in the same geographic localities. It also includes a feature to authorize agencies to make advance salary payments to attract candidates for open positions which have consistently been hard-to-fill in certain geographic areas.

Federal Employees' Compensation Act (FECA)

This law provides compensation and medical benefits to civilian employees of the United States for disability due to personal injury or disease sustained while in the performance of duty. A feature of this law provides for the continuation of pay (COP) without charge to leave for up to 45 calendar days due to disability and/or medical treatment following a traumatic injury. Employees file claims with the U.S. Department of Labor, Office of Worker's Compensation, which adjudicates the claims and compensates the employing agencies for the employee's pay and benefits during the claim period.

Federal Employees Health Benefits (FEHB)

Generally, the employee is entitled to coverage by the FEHB program if appointed to a position with Federal retirement coverage or has been on the rolls on a temporary appointment for more than one year. The Federal employer shares the cost of the premium (about 75%); actual premiums depend on the plan selected. If under a temporary appointment, the employee pays both the employer and employee shares. If the position is part-time, the employee pays the employee share and a portion of the employer's share.

Federal Employees Retirement System (FERS)

A retirement plan available to employees of the federal government. FERS covers all employees appointed to a position in the federal government after January 1, 1987. Coverage includes Social Security, a basic annuity plan, and a TSP.

Federal Employer Identification Number (EIN)

Used to identify the tax accounts of businesses. Businesses, which have employees or operate business as a partnership or corporation, must obtain an EIN.

Federal Insurance Compensation Act (FICA)

Employee and employer contributions to Social Security.

Federal Reserve Transit Number

A unique identifier for U.S.-based banks, allowing banks to transfer funds within the Federal Reserve system.

Feeder Line

A type of production line replenishment used in PeopleSoft Flow Production. If you are using feeder line replenishment, smaller production lines create subassemblies that feed directly to your production line.

FEGLI Living Benefits Act

Beginning 7/25/95, a Federal employee who is terminally ill may elect to receive a lump-sum payment equal to the full amount of basic life insurance only, or a limited portion designated in multiples of \$1000. An election to receive this benefit is irrevocable; the individual is considered terminally ill if his /her life expectancy is 9 months or less.

FICA (Federal Insurance Contributions Act)

FICA consists of both a Social Security (retirement) payroll tax and a Medicare (hospital insurance) tax. The tax is levied on employers, employees, and certain self-employed individuals.

Fictitious Calculations

Fictitious calculation rules perform temporary calculations. A fictitious calculation is a sub-calculation run during a normal calculation to determine a net that would have been computed if certain parameters were used. This result is then used for further processing in the normal calculation. A fictitious calculation is always started from inside a normal calculation, run for one payee, and run for a specified set of periods.

FIFO (First In First Out)

Method used by companies to record Disqualifying Disposition Income. If a company uses this method they record the optionees disposition of shares by attributing the shares to the earliest exercise, purchase or release dates for which shares remain available for sale.

Fill-In Employment

Employment held by persons during the time period after leaving their regular occupation in anticipation of, but before entering, military service.

Filter

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

Final Average Earnings (FAE)

The PeopleSoft Pension Administration function that averages earnings from a specified period of an employee's career. The result is used as a component of the pension benefit formula.

Final Forecast

The final forecast is the prorated version of the adjusted forecast, summarized to all levels of the product hierarchy. This is the best-guess version of the forecast that is used to make all decisions dependent on the forecast.

Final Table Merge Engine

Final Table Merge Engine is used by the PeopleSoft Enterprise Warehouse; it moves enriched data from one table into another. When you run a job in a jobstream, the immediate results are stored in temporary tables. At the end of the jobstream, the Merge Engine runs and merges the output temporary tables into the final tables, where processing can continue.

Financial Accounting

The accounting for a business entity's assets, liabilities, revenues, and expenses to determine its net worth and to produce financial statements. Within Generally Accepted Accounting Principles, a business has some latitude as to when and how to record its financial transactions, as long as it continues to meet its legal and regulatory requirements. A business' financial accounting requirements are not necessarily the same as its cost accounting requirements. The one should not be mistaken for the other (i.e. the extent to which a company's financial accounting system meets its cost accounting needs depends on how it has chosen to describe its chart of accounts and the level at which it has chosen to record financial transactions.

Financial Instrument

In the financial services industry, a specific product or service sold by a financial institution to its customers. In terms of the reporting hierarchy, a product falls under a ledger account, while an instrument falls under a product. A product may be treated like a generic description or category, while an instrument is a specific instance of a category.

Financial Performance Measures (FPM)

For the financial services industry, the Financial Performance Measures program performs calculations on financial instruments based on the rules defined in the Financial Calculation Rules module, and using input from the Cash Flow Generator, Stratification engine, Product Pricing, and Curve Generator. Its calculations include: measures of duration, option-adjusted spread and option-adjusted cost for PeopleSoft Funds Transfer Pricing, and Monte Carlo simulation for PeopleSoft Asset Liability Management.

Financial Product

In the financial services industry, a product or service sold by a financial institution to its customers. In terms of the reporting hierarchy, a product falls under a ledger account, while an instrument falls under a product. A product may be treated like a generic description or category, while an instrument is a specific instance of a category.

Financial Services Instrument

In the financial services industry, products created by financial institutions and sold to retail customers. Product prices and interest rates are set by the financial institutions and take into account its customers' behavioral models.

Financial Statement Simulation

A facility within Planning & Simulation which establishes rules for simulating future period, or pro-forma, financial statements. The user defines corporate financial policies, such as corporate tax rates, dividend distribution frequency, and force balancing rules, which are then applied to cash flows for a given future accounting period. The Financial Statement Simulator engine drives costs and revenues to accounts on PF_LEDGER_F00 via a scenario.

Financials Warehouse

See Warehouses.

First Year Amount

See 1st Year Amount.

Fixed Basis

The basis option enables you to create the data for the Basis online, as part of the rule. Fixed Basis is used with the Allocation Manager only. It is available with all methods except when Period-Based Allocation is being used. The Fixed Basis is a predetermined table that can be populated online.

Fixed Offering

The offering type is fixed when the end date of each offering is the same for all employees regardless of the employee's grant dates.

Fixed Percentage

A fixed percentage value. The source pool amount will be split based on this percentage to get the target amount. Used with the Allocation Manager.

Fixed Period Requirements

In PeopleSoft Enterprise Planning and Production Planning, a lot-sizing technique that sets the order quantity to the demand for a given length of time.

Fixed Picking Bin

A dedicated picking location for an inventory item. Fixed picking bins are replenished from bulk locations when the available quantity falls below the optimal quantity.

Fixed Plan

A stock purchase offering period where the ending offering date will be the same as the purchase date. Eligible employees will always purchase stock on the specific purchase dates and by the purchase rules you define.

Fixed Quantity

An **Inventory Policy** method that defines a fixed amount of an item to be ordered to meet replenishment needs. This method can be selected as an inventory policy for order quantity, safety stock, **Reorder Point**, and minimum and maximum parameters.

Fixed Source

The fixed source option enables you to create the data for the Source online, as part of the rule. Fixed Source is used with the Allocation Manager only. It is available with all methods except when Period-Based Allocation is being used. The Fixed Source is a predetermined table that can be populated online.

Flexible Credit

Any credit associated with a given benefits program, plan, or type of coverage. Credits based on an entire program can be applied toward the benefit costs however the employee chooses.

Flexible Hours

Hours during the workday, workweek or pay period during which a time reporter covered by a flexible work schedule may choose to vary his times of arrival and departure from the worksite (see Scheduling)

Flexible Spending Account (FSA)

An account to which an employee and (optionally) an employer pledge an annual amount for a plan year. The employee then submits claims for authorized expenses.

Flexible TimeSpan

A user-defined period into which costs can be collected. Flexible TimeSpans can be as long or as short as you like—covering multiple years or a single day. The main purpose of Flexible TimeSpans is to assist you in analyzing costs.

Flexible Work Schedule

A method of scheduling a time reporter's time that is based on a range of flex hours of start and stop times and core work hours. This method can be used for scheduling clock and elapsed time reporters (see Scheduling)

FLSA Status

A PeopleSoft Human Resources term that is used to indicate whether a job is exempt or nonexempt according to the Fair Labor Standards Act. All employees associated with a

particular job will receive that job's FLSA Status. FLSA Status is an eligibility determination factor for PeopleSoft Benefits Administration.

Forecast Attribution

A FSI (financial services industry) transformation process through which forecasted product originations are pooled and run through the cash flow engine for future periods.

Forecast Calculation Process

In PeopleSoft Demand Planning, the process by which a **Statistical Forecast** is generated for each item at each level of the view. When a **Forecast Item** is set to recalculate, the system tries several forecast calculation methods and picks the one with the least amount of error. This process also makes adjustments for promotions and filters for abnormal demand.

Forecast Definition

Forecast definitions are a set of forecasting rules that generally govern multiple forecasts distinguished by key properties such as products, customers, channels, and so forth.

Forecast Element

Each forecast within a single definition is called a Forecast Element.

Forecast Fulfillment

In PeopleSoft Demand Planning, a process used to manage forecasted demand over a period of time. The process makes it possible to divide the total forecast demand into portions so that certain portions can be met, even if the total forecast cannot be met entirely.

Forecast Item

In PeopleSoft Demand Planning, a logical item used as the basis to forecast demand. The components of a forecast item key are defined for each level in a forecast view.

Forecast Level

See Level.

Forecast Period

A period in time as defined by the calendar for which data is processed through the PeopleSoft Demand Planning model.

Forecast Start Period/Year

Determines the most recent period for which demand data is available for a forecast view. This period can also be described as the last actual demand period to have had an impact on the forecast.

Forecast View

See **View**.

Foreign Education

Education acquired outside of any state of the U.S., the District of Columbia, the Commonwealth of Puerto Rico, a Trust Territory of the Pacific Islands, or any territory or possession of the U.S.

Form 10-K

A form used for annual reports pursuant to Section 13 or 15(d) of the Securities Exchange Act of 1934 for which no other form is prescribed.

Form 10-Q

A form used for quarterly reports under Section 13 or 15(d) of the Securities Exchange Act of 1934, filed pursuant to Rules 13a-13 or Rule 15d-13. This report, which public companies are required to file quarterly with the SEC, provides unaudited financial information and other selected material.

Form 5500 Participant Count Extract

A PeopleSoft Pension Administration data extract containing data that a plan administrator needs in order to complete IRS Form 5500, used to report on the number of plan participants.

Form S-8

A form used to register securities offered by a reporting company under its employee benefits plans, including stock option plans. Also called the Registration Statement under the Securities Act of 1933.

Form W-2

A form used by employers to provide workers with a statement of wages, tips and other compensation from the previous year. This form, distributed employees by January 31 of each year, reflects state and federal taxes, social security, Medicare wages, and tips withheld.

Formula

Element which enables you to define your own formulas for use—gives further flexibility to define complex organizational needs.

Formula Plan

This compensation distribution plan type is based on a pay out rule, as the pay out rule is defined. The pay out rule can be based on a flat amount, a percentage, or a data element. Whereas a Target Plan distributes pay out based on a comparison of a performance measure against a target, in a Formula Plan the pay out is based just on the pay out rule.

French Professional Elections

French companies employing a certain number of employees must hold elections for selecting personnel representatives (Délégués du personnel), and members of the Work Council (Comité d'Enterprise).

Frozen Rate

A rate that is applied to allocate resources to activities in place of the actual, budgeted and capacity rates calculated by the Activity-Based Management (ABM) Engine.

FTE (Full Time Equivalency)

FTE is the percent of full time the employee should normally work in this job. Full time is defined by the Standard Hours specified in either the Salary Plan Table or the Default Standard Hours specified in the Installation Table.

FTP (Funds Transfer Pricing) Adjustments

Adjustments made to the PeopleSoft Funds Transfer Pricing (FTP) base rate, for such factors as geographic premiums, liquidity premiums, embedded options, or incentive programs.

FTP (Funds Transfer Pricing) Base Rate

In PeopleSoft Funds Transfer Pricing (FTP), this refers to the basic charge or credit that is applied to a ledger account, a product, or an off-balance sheet position.

FTP (Funds Transfer Pricing)

See PeopleSoft Funds Transfer Pricing.

Full-Time Equivalent

See FTE.

Function

A category of pension calculation. PeopleSoft Pension Administration divides a pension calculation into nineteen “core functions” such as Service, Final Average Earnings, and Benefit Formula.

Function Result

The calculation rules for any of PeopleSoft Pension Administration’s nineteen core functions. These rules match Definitions—the specific parameters for the function—to the Groups of employees that use that particular definition. Function Result also refers to the value produced by the rules.

Fund ID

In the financial services industry, Fund ID is a lookup code used to track investment funds associated with a financial instrument or account. Provided primarily by the financial analytic applications to track investment funds for insurance policies.

Funds Transfer Pricing (FTP)

See PeopleSoft Funds Transfer Pricing.

Fungible

This term describes a resource used for multiple activities.

Future Period

Any pay period which is not current and whose close date hasn't passed (see Time Reporting).

Future Periods

The number of periods of future forecasts maintained by the PeopleSoft Demand Planning system.

G**Gang Reporting**

See Crew Reporting.

General Deduction

Any non-benefit deduction. Examples include charitable deductions, union dues, parking, garnishments, and bonds. General Deductions are calculated from the General Deduction Table; Benefit Deductions draw on one of the benefits tables.

General Ledger Distribution

The process and guidelines by which accounting information is transferred from your PeopleSoft Receivables or Deduction Management system to a general ledger system.

General Schedule (GS)

Compensation and pay plan used by the Executive Branch of the federal government.

Generation Control

Generation control elements allow you to indicate to the system whether to process an element based upon criteria you define. There are six parameters that control this function and comprise the definition of the generation control element—HR Status, HR Action/Reason, Segment Status, Frequency, Formula, and Run Types.

Generic Conversion Factor

A conversion factor that applies universally between two units of measure. The factor is used in the conversions between levels of PeopleSoft Demand Planning **Forecast Items** and Inventory Planning **Policy Item**.

Generic Process Type

This term applies to 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," "SQR Report," and so on.

Geo RSZ Code

This code is for Belgian employers to track the geographical location for RSZ codes.

Geographic Location Code

In Canada this code is prescribed by the government and refers to the location a business is in.

Giveaway Adjustment Type

In PeopleSoft Order Management, the price break tables are set up to indicate what product the user receives as a free premium based on defined quantities or prices. The giveaway item does not have to be the same product that the customer is purchasing. For instance, you may set up a price break that indicates that a purchase of 100 widgets entitles the customer to one or more free T-shirts. The system automatically adds an order line for the free item. Giveaways cannot be applied to the total order.

Goals Matrix

In PeopleSoft Workforce Analytics, a matrix used to create calculation rules for group or employee performance goals. You can combine and standardize multiple performance goals into a single, weighted, goal score, against which actual performance is compared. A Goals Matrix can be used to in conjunction with a pay out distribution plan called a Target Plan.

Grace Period

A period that is a number of hours or minutes before or after a scheduled punch where a time reporter's punch is accepted. For Stock Administration, the period of time an optionee has to exercise an option after termination and before the option expires.

Grade

A range of pay in a graduated scale that includes positions of different occupational groups. The work performed should be equivalent as to the level of difficulty and responsibility and the level of qualification requirements of the work. The levels are established and designated within a specific pay plan by law or regulation.

Graduate Education

Successfully completed education in a graduate program for which a bachelor's or higher degree is normally required for admission. To be creditable, such education must show evidence of progress through a set curriculum, i.e., it is part of a program leading to a master's or higher degree, and not education consisting of undergraduate and/or continuing education courses that do not lead to an advanced degree.

Grandfathered Benefit

A benefit that an employee was entitled to prior to a change in the plan and that defines the employee's new minimum level of benefits. The change might be caused by a plan merger, new legislation, or a plan amendment.

Grant

A contractual right giving an individual the option to purchase a specified number of shares of stock through an Equity Compensation Plan. Also known as an option.

Grant Agreement

The legal document issued by a company defining the number of shares granted, grant price, vesting schedule and other terms and conditions of the stock option or stock award.

Grant Date

The date the individual begins participating in a stock purchase offering. The date on which an option or other award is granted. The date the company enters into the grant agreement. The underlying stock's fair market value on this date generally derives the option price.

Grant Price

The price per share at which the stock option was granted. This is the price per share the individual must pay when exercising the option.

Gross Salary

The sum of an employee's salary and earnings defined as part of gross salary. The gross salary is used to calculate budget amounts for benefit plans defined as a percentage of an employee's salary.

Gross-up

The process used to calculate taxes and resultant gross pay from a check for an exact net amount.

Group

In PeopleSoft Billing, a specific term for a posting entity composed of one or more transactions (items, deposits, payments, transfers, matches, or write-offs).

Group

Any set of records associated under a single name or variable in order to run various calculations in PeopleSoft Business Processes. In Time and Labor, for example, employees are placed in groups for time reporting purposes, while in Administer Variable Compensation, groups identify which employees are eligible for what forms of compensation. In PeopleSoft Pension Administration, you'll use Custom Statements to define criteria for grouping employees, then by associating calculation rules (Definitions) with specific Groups, you can vary rules for different classes of employees.

Group Asset

A financial asset with no cost information. It is used to depreciate the sum of the costs of its associated group member assets.

Group Asset Depreciation

The depreciation of a group asset calculated using an average service life set by a local regulatory agency and a calculated group depreciation rate.

Group Coverage (Or Generic) Qualification Standards

Standards prescribed for groups of occupational series that have a common pattern of education, experience, and/or other requirements.

Group Member Asset

A financial asset with cost information. Cost information for all group members of a group asset is summed up to the group asset level, where depreciation is calculated.

Group Security

The ability to grant or deny access to groups. You can set up group security by Group ID or by user ID.

Group Security [Time and Labor]

The ability to grant access to employee time, by providing security through Time and Labor's groups functionality. For example, you might want your employees to only access their own records, or allow your supervisors who handle all of the time input for have access to specific groups. You can restrict the user from accessing everyone, or allow the user to be able to access only their own records, or only a specific group. This feature also provides the ability for employees to report their own time.

Group Type

An indicator of the activity that created the billing group: billing, maintenance, payment, transfers, or unposted.

H

Handicap Code

A code that identifies a type of physical or mental impairment that substantially limits one or more of an employee's major life activities.

Hazard/Disposal Code

An inventory item group sharing a disposal routine.

Headcount

The number of people represented by a given Employee Survey record in the PeopleSoft Enterprise Performance Management product line.

Health and Safety Executive (HSE)

Health and Safety reporting for your UK operations is sent to the local office of the HSE per the requirements of the RIDDOR (Reporting of Injuries, Diseases, and Dangerous Occurrences Regulations).

Health Benefits Code

An alpha/numeric code that identifies each Health Benefit plan.

Health Benefits Effective Date

Date the health benefit plan goes into effect or the effective date of cancellation.

Hierarchy

Hierarchy refers to the relationship between the levels in a dimension.

Highly Compensated Employee (HCE)

An IRS employee category applied to employees who are considered “highly compensated” according to a federally set standard. This distinction is used for the purposes of nondiscrimination tests, to determine that Section 401 and Section 129 plans do not discriminate in favor of highly compensated employees.

HIPAA

The Health Insurance Portability and Accountability Act of 1996. PeopleSoft Benefits applications enable you to comply with this act, which requires that employers provide Certificates of Group Health Plan Coverage to employees who have their health coverage terminated. This certificate lists group health coverage an employee had for the twelve month period prior to the date coverage ended as a result of termination of coverage. The HIPAA certificate will be used by subsequent health coverage carriers to evaluate pre-existing condition clauses, if applicable.

Historical Periods

In PeopleSoft Demand Planning, a component that indicates the maximum number of periods of historical demand maintained for a **Forecast Item** within a **Forecast View**. Historical periods must be a minimum of two years in order to support the development of seasonal models based on an item's demand history.

Historical Rules

An element used to set up rules that retrieve data from prior periods. Historical rules can be used in formulas and fictitious calculations.

Historical Usage Calculation Method

In PeopleSoft Inventory Planning, a method that defines the set safety stock or minimum inventory level. The usage is based on the review of historical demand over the number of effective periods. The historical demand quantity is determined by one of four methods; maximum possible usage, Lead Time, estimated daily or period use, and static values calculations.

Hold Grade/Step

Grade/step the employee was in prior to receiving a temporary promotion.

Hold Last Equivalent Increase (LEI)

Date held by an employee for this event prior to receiving a temporary promotion. Necessary in order to establish the WGI due date if returning to original grade/step.

Hold Position Description

The new position description number that is the result of a reclassification action prior to the NOA being processed.

Hold Purchase

A flag that tells the system to keep this participant in the purchase process. The hold flag is maintained at the contribution page.

Hold Within Grade Increase (WGI) Due Date

WGI due date prior to an employee receiving a temporary promotion.

Holding Period

Typically refers to the holding period required for ISO's and Qualified Section 423 Purchase Plans, to receive preferential tax treatment on a disposition of shares. See Disqualifying Disposition.

Hours Counting Service

A service calculation that uses actual or generated hours to determine the service credited to a pension plan participant.

Hours Equivalence Service

A service calculation that uses hours to determine service, but that uses a set number of hours per day, week, or other period worked rather than counting actual hours.

HR Action/Reason Category

A group of related job actions—for example, hire and rehire—treated similarly for pension purposes in PeopleSoft Pension Administration.

HRMS Warehouse

See Warehouses.

I**Ignore Plan**

Complex event processing feature of PeopleSoft Benefits Administration that enables the user to designate plan types linked to a particular Event Rules/Event Classification combination as being unaffected by Benefits Administration processing.

Ignore Violations

The ability to report over capacity violations but not to score or repair them during the optimization process.

Imputed Income

Theoretical income that a company pays on behalf of an employee but the individual does not actually receive. This “theoretical income” must be added to the employee’s gross wages. In general, imputed income refers to the value of excess Group Term Life or Dependent Life coverage.

In Punch

Indicates start of a shift.

In the Money Option

When the fair market value of the stock is greater than the grant price of an option.

INAIL code

In Italy, the INAIL code is used to classify jobs according to the level of risk associated with the job and the related risk insurance required by the employer. INAIL codes are defined by the employer.

Incentive Pay Plans

In PeopleSoft Workforce Analytics, pay plans that are formula-driven based on the expected results defined at the beginning of a performance cycle. Incentive plans are designed for the individual worker, or for group levels such as teams, business units, divisions, or company-wide. Incentive plans are used for a variety of reasons; including cost control, alignment of employee and shareholder interests, and increased focus on specific performance indicators.

Incentive Plans

Pay plans that are formula-driven based on the expected results defined at the beginning of a performance cycle. Incentive plans can be designed for the individual worker or at group levels such as teams, business units, divisions or company wide.

Incentive Stock Option (ISO)

For an option to be considered an Incentive Stock Option, it must have the following characteristics:

- The option must be granted pursuant to a plan which includes the aggregate number of shares which may be issued under options and the employees (or class of employees) eligible to receive options, and which is approved by the stockholders of the granting corporation within 12 months before or after the date such plan is adopted;
- The option must be granted within 10 years from the date such plan is adopted, or the date such plan is approved by stockholders, whichever is earlier;
- The option is not exercisable after the expiration of 10 years from the date such option is granted;
- The option price is not less than the fair market value of the stock at the time such option is granted;
- The option is not transferable by such individual otherwise than by will or the laws of descent and distribution, and is exercisable, during his lifetime, only by him, and;
- The optionee, at the time the option is granted, does not own stock possessing more than 10% of the total combined voting power of all classes of stock of the employer corporation or of its parent or subsidiary corporation.

Incomplete Punch

A punch that cannot be processed (i.e. missing employee ID, invalid date or time).

Incremental Budgeting

A budgeting option during budget development that uses prior year actual or budget values as a basis and then applies a percentage that increments the base. PeopleSoft Budgeting-specific.

Incumbent

An employee currently assigned to a position.

Indirect Compensation

Typically involves non-cash types of compensation awarded to the individual in exchange for their contribution to the organization. Common types of indirect pay include health and welfare benefits (for example, medical, dental, vision, long-term disability, short-term disability, unemployment insurance), payment for time not worked (for example, holiday, vacation, sick), and employee services and perquisites (for example, club memberships, parking, holiday gifts).

Indirect Cost

A cost that is assigned by management to an activity or a cost object. An example is the cost of office space assigned to an activity.

Individual Occupational Requirements

Requirements, e.g., experience or education, for particular occupational series of positions within a series and are used in conjunction with a group coverage (generic) standard.

Individual Retirement Record (IRR)

Used by the Office of Personnel Management (OPM) as the basic record for determining the retirement benefits payable to separated federal employees and their survivors. Employees covered by the CSRS retirement plan require SF-2806. Employees covered by the FERS retirement plan require SF-3100. In addition, the SF-2806-1 and SF-3101 are used for corrections to the IRR. See also Correction to IRR.

Inherit Control Group Policies

In PeopleSoft Inventory Planning, a feature that controls whether the policy for an item is set explicitly or defaults from the associated **Policy Control Group**. A series of check boxes enable you to define which policies to inherit.

Initial COBRA Events

The event which makes an individual eligible for COBRA coverage. Typical initial COBRA events include loss of benefits eligibility due to termination, reduction in hours, retirement, and military leave, as well as divorce, death of employee, and Medicare entitlement. See COBRA and Secondary COBRA Events.

INSEE (National Institute for Statistical and Economical Studies) Codes

INSEE is an official statistics and economics organization in France. INSEE codes for your French company's organizations are used in regulatory reporting.

INSEE PCS (Classification par Catégorie Socio-Professionnelle) Code

Each PeopleSoft Human Resources French Jobcode is linked to a four-digit INSEE PCS, or social/professional classification code.

In-Service Date

In PeopleSoft Asset Management, the date upon which an asset is placed in service. In-service date is used in conjunction with an asset's prorate convention to determine Begin Depr Date.

Inservice Placement

Includes a noncompetitive action in which a position is filled with a current or former competitive service employee through promotion, reassignment, change to lower grade, transfer, reinstatement, reemployment, or restorations. Inservice placement also includes noncompetitive conversion of appointees whose Federal excepted positions are brought into the competitive service under Title 5 CFR 316.702, and Department of Defense/Nonappropriated Fund (DOD/NAF) and Coast Guard NAF employees whose positions are brought into the competitive service.

Insider

An officer, director or principal shareholder of a publicly owned company and members of his or her immediate family. This category may also include other employees of the company and people who obtain nonpublic information about the company.

Insider Trading

When a person trades a security while in possession of material non-public information in violation of a duty to withhold the information or refrain from trading. The securities law broadly prohibits fraudulent activities of any kind in connection with the offer, purchase, or sale of securities.

Instance

A row of data on the Positive Input table. Instances of positive input can be entered manually, or can be system generated. They can also be received from other applications, such as PeopleSoft Time and Labor.

Integration Level

The salary level in a defined benefit excess plan at which a higher benefit rate becomes applicable. For example, the following formula uses a \$10,000 integration level: 1% of Final Average Earnings up to \$10,000 plus 1.75% of Final Average Earnings over \$10,000.

Integration Template

A high-level template that defines the integration between PeopleSoft Projects and your other financial applications. Each integration template you create defines a specific set of business units from your other financial applications. Each project is then assigned an integration template containing this preset integration information. You can use Integration Templates to set up joint ventures, and new transactions added to that project will reflect the business units defined in the integration template.

Intensity

The cost for each unit of the activity driver.

Interest

Some companies pay interest on the monies that are being withheld from employees' paychecks. The interest plus the employees' stock purchase contributions are used to purchase stock at the end of the purchase period.

Interest Rate Modeling

An FSI feature that allows you to model interactively interest rate scenarios for Asset Liability Management, and to run rate scenarios and analysis in real time.

Interest Rate Sensitivity Model

In the financial services industry, this support module describes in granular terms how a group of customers holding a specific type of instrument with a particular interest rate will respond to changes in interest rates in the market.

Interface Loader

An SQR delivered with PeopleSoft Asset Management that is used to transfer load lines into the PeopleSoft Asset Management loader tables.

Internal Data

Data from PeopleSoft ERP systems, or other legacy ERP systems used by your organization.

Interpolation

To calculate a value of a function, or series, between two known values.

Interunit Account

The account for each business unit to which other business units in the same corporation refer when they need to distribute amounts across business units. These accounts are used to keep the individual ledgers in balance when a single transaction affects multiple business units.

Inter-Unit Drivers

Drivers that provide a means of establishing relationships between the cost objects of one organization with the supporting activities of the organizations that share business units and models.

Interunit Transaction

A transaction that involves moving amounts from an account in one PeopleSoft General Ledger business unit to an account in another General Ledger business unit.

InterUnit Transfer

A transfer that occurs between different business units.

IntraUnit Transfer

A transfer that occurs within one business unit.

Intrinsic Rewards

A reward that is generated by the worker internally such as job satisfaction, as opposed to Extrinsic Rewards which are tangible rewards.

Inventory Adjustment

A process that enables you to change the quantity of an item in the inventory system to match the actual physical quantity found in the **Storage Location**.

Inventory Business Unit

Usually a warehouse. You establish a separate inventory business unit (IBU) for any one of the following reasons: 1) You want on hand visibility to a specific location of your business that manages inventory. 2) You want to define replenishment rules for a specific location of your business that manages inventory. 3) You maintain standard and average costs in a specific location of your business that manages inventory.

Inventory Cost Element

A cost that can be associated with inventory items and inventory transactions. Examples include freight, overhead, and transportation. Each cost element has a unique cost code.

Inventory Item

A tangible commodity that is stored in an Inventory business unit (Ship From warehouse).

Inventory Location

See **Storage Location**.

Inventory Policy

In PeopleSoft Inventory Planning, a set of rules that controls how inventory policy values are calculated for items. Inventory policy is defined at the **Policy Control Group** and stockkeeping-unit levels. The elements that make up inventory policy are order quantity, safety stock, **Reorder Point**, and minimum and maximum policies.

Inventory Transaction

An event that moves inventory into, within, or out of the inventory business unit. Examples include material transfers, inventory adjustments, and standard issues.

Inventory Transaction Group

An identifier that categorizes transactions by type for costing purposes. For example, you can group all types of interunit transfers together.

Invoice Format Identifier

An identifier for the formatting options that determine the sorting and summarization levels of invoice information.

IRC 423 (Internal Revenue Code 423)

The section of the IRC that defines a Qualified Employee Stock Purchase Plan.

IRR Fiscal Data Accumulation

This report accumulates all retirement deductions for employees, as well as any LWOP and any basic pay that was received when an employee was not covered by the CSRS or FERS retirement plans.

IRR Remarks

Special remarks that are documented on an employee's IRR. IRR Remarks can be set up ahead of time and can be system-entered text or employee-specific.

IRR Status

IRRs can be in pending or final status. Those in pending status can be updated and corrected. A final status indicates that the IRR has been processed and can't be updated or corrected except through a Correction IRR or a Supplemental IRR.

IRR Worksheet

A preliminary IRR form that enables an agency to print a pending IRR for a separated employee, review it and make corrections, if necessary. Agencies can also use the IRR Worksheet to view a current IRR for an active employee.

ISO IRS \$100K Limit

The limit the IRS places on the exercisable value of Incentive Stock Options (ISOs) of \$100K per calendar year based upon the fair market value at the time of grant (Section 422 of the Internal Revenue code).

ISO to NQ Grace Period

The period of time after which an Incentive Stock Option is treated as a Non-Qualified Stock Option for tax purposes upon the termination of employment according to Internal Revenue Code Sections 421 and 422. Depending on the termination reason the option is treated:

- If the termination reason is for any reason other than death or disability, and an exercise occurs more than three months from the termination date, the system withholds taxes as if the option is a non-qualified stock option.
- If the termination reason is disability, the system withholds taxes if an exercise occurs more than twelve months from the termination date.
- If the termination reason is death, the system always treats the option as an ISO.

Issue

See **Material Issue**.

Issuer

A legal entity that has the power to issue and distribute a security.

Item

See **Inventory Item Planning Item** or **Receivables Item**.

Item Content Provider

Third-party software consisting of web-based catalogs of item and price information. These systems benefit the design and purchasing of new products by accelerating item location, maximizing design reuse, and reducing acquisition costs. PeopleSoft Purchasing, Engineering, and Inventory integrate to Item Content Providers, and the information is used by many other PeopleSoft applications.

Item Rounding Rules

A set of rules determining how fractional values are rounded so that calculations result in whole numbers. Rounding rules are used in conjunction with **Quantity Precision Rules**.

Item Simulation

In PeopleSoft Demand Planning, a process that enables you to interact with the forecast in a manageable manner and perform "what-if" analysis by comparing the effects of different forecast models.

Item Type

An identifier that defines inventory items at a very high level, and may include sets of Item Families. For example, the families Computer Items and Office Furniture might be categorized by types like Outside Manufacturing, Finished Goods, and Work In Progress.

Item-Specific Conversion Factor

A conversion between the same two units of measure when the measurements have a different value for an item. For example, a conversion between packaging unit and stocking unit.

Iterative Processing

Refers to a concept on only re-calculating those payees who have had changes and need to be recalculated (if you choose to run your payroll multiple times before actually finalizing it). This concept saves you a lot of time as you only have to recalculate those payees who have had a data change or who you indicate you would like to be recalculated.

J**Java Server Handlers (JSH)**

The JSH manages network connectivity, making service requests from the Jolt Repository, and translating Tuxedo buffer data into the Jolt buffer.

Java Station Listeners (JSL)

The JSL handles the work of the client connection, tracking client messages, and session handoff.

Job Code

An ID for a job as defined on the Job Code table.

Job Code Components

The pay components assigned to a job code by associating rate codes with job codes on the Default Compensation page or the Non-Base Compensation page of the Job Code table.

Job Code Cost

Evaluation of salaries for specific job codes.

Job Compensation Rate

The compensation rate of the corresponding job row.

Job Events

Actions relevant to an employee's employment—such as a hire, transfer, or termination—that can affect benefit program or plan eligibility. Used by PeopleSoft Benefits Administration. See Event Class.

Job Order Cost Accounting

A cost accounting method that attempts to develop a discrete cost for each job performed or product produced. Only the material, labor, and overhead required to complete the job are attributed to the job cost.

Joint and Survivor Payment Option

A form of pension payment in which benefits are paid for the life of the participant and a beneficiary. Should the beneficiary outlive the participant, the benefit continues (often in a reduced amount) for the life of the beneficiary.

Joint Staffing Report

In the United Kingdom governmental agencies are required submit the Joint Staffing Report. Although it is mainly designed for government sector organizations, commercial organizations may also use this SQR to provide a summary of their staffing by department, job code, gender and full/part time employment status.

Jolt

A BEA/Tuxedo companion product that runs on an application server domain and is used to listen for Web Client Java requests and transfer them to Tuxedo.

Journal Code

The second highest level of three categories for defining a financial transaction (or document), necessary when using document sequencing. Examples of journal code are domestic sales and export sales. This category is preceded by journal type and followed by document type.

Journal Generator Template

A table containing defaults to be used in journal generation. PeopleSoft Asset Management and Billing require one journal generator template for each transaction type.

Journal Line

A record storing a double-sided, balanced entry for a given journal. A single journal usually includes multiple lines. The sum of the monetary amounts for the journal lines in one journal totals zero (debits = credits).

Journal Template

A list of the characteristics of the general ledger journal entries that will be created from your PeopleSoft Receivables system.

Journal Type

The highest level of three categories for defining a financial transaction (or document), necessary when using document sequencing. Examples of journal types are sales journal and purchase journal. This category is followed by journal code, then document type within the journal code.

Journal Voucher

A PeopleSoft Payables voucher that enables you to make accounting entry modifications while keeping your PeopleSoft General Ledger and Payables systems in sync. Like the adjustment voucher, the journal voucher is linked to an existing voucher.

K**Kanban ID**

A unique identifier used to track Kanban cards and replenishment requests when using PeopleSoft Flow Production.

Keep Ledgers in Sync

An option in PeopleSoft General Ledger that defines how a transaction should be posted—to all ledgers in a ledger group as opposed to only a single specified ledger.

Key

See **ChartKey**.

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.

Key Performance Indicator (KPI)

KPI is used by the PeopleSoft Performance Management analytical applications. KPIs are high-level measurements of how well an organization is doing in achieving critical success factors. A KPI defines the data value or calculation from the Data Warehouse tables upon which an assessment is determined.

KPI (Key Performance Indicator)

See Key Performance Indicator.

Knowledge, Skills, And Abilities (KSA)

Also known as Competencies, these are attributes required to perform a job and are generally demonstrated through qualifying experience, education, or training. *Knowledge* is a body of information applied directly to the performance of a function. *Skill* is an observable

competence to perform a learned psychomotor act. *Ability* is competence to perform an observable behavior or a behavior that results in an observable product.

L

Labor Costs

Actual expenditures associated with *salary* portion of time reporter expense.

Labor Dilution

A process that occurs after the Labor Distribution process in PeopleSoft Time and Labor. The labor dilution process takes the costs that the payroll system has calculated for payable time, determines an average or rate per hour, and applies the average amount evenly across all reported hours for the day.

Labor Distribution

The process of distributing payroll expense to the corresponding payable time entries generated in PeopleSoft Time and Labor.

Labor Distribution Amount

An actual labor cost associated with reported time.

Last Equivalent Increase (LEI)

Reflects the effective date of the last step received in grade or the last promotion, whichever is most current (does not include QSI). Used as the basis to establish an employee's WGI due date.

Last Physical Counting Event

The last date the inventory item was counted. This information is stored with each inventory item.

Last Purchase Date

The item's most recent purchase date in the inventory business unit.

Last Putaway Date

The item's most recent putaway date in the inventory business unit.

Last Putaway Document Number

The item's most recent putaway document identification number in the inventory business unit.

Last Receiving Date

The item's most recent receipt date in the inventory business unit.

Last Shipping Date

The item's most recent ship date in the inventory business unit.

Last Shipping Document Number

The item's most recent shipping document identification number in the inventory business unit.

Law Enforcement Officers (LEOs)

Positions within the Federal government involving law enforcement. Under FEPCA, many of these positions are entitled to additional special pays.

Lead-Time Estimated Usage

An inventory planning method for calculating historical usage of an item. The historical demand is prorated on a daily basis and then multiplied by the number of days lead time for each effective historical period. The maximum period value is then used as the safety stock or minimum stock level. This method should be used for items that have a steady demand pattern throughout each period.

Lead-Time Period Usage

An inventory planning method for calculating historical usage of an item. The purchase lead time is rounded up to a specified number of periods. The historical demand is calculated as the maximum usage during these periods and the safety stock or minimum-stock level is set to this value.

Leave

Time entitled to an employee as a benefit, such as, Sick, Vacation, STD, and LTD. This process is managed by HRMS (see Time Reporting).

Leave Accrual Processing

Processing of leave accruals is used to maintain employee leave balances. All leave benefit plans accrue leave by length of service or number of hours worked. Leave accrual processing is used to determine the employee's leave accrual award and resulting leave balance.

Leave Accruals

Hours that employees earn to use at another time, such as annual leave and sick leave.

Leave Plan

A method for earning and managing leave time.

Leave Without Pay (LWOP) Total (Cumulative)

An employee's cumulative number of hours of leave without pay (LWOP).

Ledger Group

In PeopleSoft General Ledger, a group of ledgers consisting of one primary ledger and secondary ledgers.

Ledger Mapping

Ledger mapping is a process that enables you to relate expense data from your 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 the PeopleSoft Enterprise Warehouse (EW), Ledger Mapping enables you to map general ledger accounts to the EW Ledger table.

Ledger Template

A table containing records and fields common to all ledgers that ensures that all ledgers specified in a ledger group share the same physical layout.

Ledger Type

The unique combination of a single ledger, scenario, and fiscal year. Multiple ledger types make up a ledger type set.

Ledger Type Set

A collection of ledger types, the members of which will represent the members of your ledger type dimension.

Legend ID

A way of recording information that is displayed upon the Issuance Instruction Report. Can be used to record a notice that should appear on the back of a stock certificate indicating that the shares represented are "Restricted Securities." Can also be used to indicate how shares should be processed, as in the case of Swaps, Trades, Repurchases and SAR Exercises.

Level

A section of a tree that organizes groups of nodes.

Defines a set of **Forecast Item** with a common key structure. Each level is related in a hierarchical definition with other levels in the view. A level definition contains descriptive and control data that relates to the operation of the forecast at each level within the view.

Level

The section of a tree that organizes groups of nodes.

Level Income Payment Option

An annuity form of pension payment in which payments are increased in early years (prior to eligibility for Social Security benefits) and decreased in later years when Social Security benefits are also received. The goal is to provide a relatively constant total retirement income both before and after Social Security eligibility.

Life Profile

In PeopleSoft Demand Planning, a feature that enables you to establish product forecasts based on predefined patterns in an item's life cycle.

Lifecycle (of Reported Time)

A representation of time through the various stages of Time and Labor; includes processing of current, future, and previous period time from scheduling and time capture through Time Administration and distribution.

LIFO (Last In First Out)

Method used by companies to record Disqualifying Disposition Income. If a company uses this method they record the optionees disposition of shares by attributing the shares to the most recent exercise, purchase or release dates for which shares remain available for sale.

Line-Item Budgets

The budget amounts associated with ChartField distributions that make up an organization's budget. Line-item budgets include personnel costs as well as operating and maintenance costs. They also include revenue estimates. PeopleSoft Budgeting-specific.

Line Schedule Editor (LSE)

PeopleSoft Production Planning utility or tool that displays production tasks for multiple products on multiple resources across multiple periods of time.

Literal Mapping

In PeopleSoft Demand Planning, a mapping option for formatting data that is common to all records being imported. This enables you to set an available field value for all the loaded rows.

Load

The feature that initiates a process to automatically load information into a PeopleSoft application—for example, populating the PeopleSoft Benefits database with plan-level election information.

Load Activation

Load Activation enables you to specify exactly which part of your Data Mart to build, including security. You set up load activation on the Load Activation page.

Load Planning

The PeopleSoft Inventory feature that picks, packs, and ships orders by Load ID. Load Planning is also used to estimate shipping weight, volume, and charges.

Loader Table

Any table in PeopleSoft Asset Management used to store load lines before they are loaded into the system as open transactions. The loader tables comprise INTFC_FIN, INTFC_PHY_A, and INTFC_PHY_B.

Loan Exercise

A form of cash exercise, typically requiring a loan agreement and a promissory note.

Local Code

In PeopleSoft Demand Planning, a type of validation used for a user-defined field code. If a user-defined field is marked to require local table validation, **User-Field Code** are used to determine the list of valid values for the field.

Local Functionality

Local functionality is the set of information in PeopleSoft HRMS that is available for a specific country. You can access this information when you click on the appropriate country flag push button in the global window, or when you access it by a local country menu.

Location Accounting

An accounting method that captures and records material movement within the warehouse, providing accounting visibility based on where the inventory resides. You can designate certain **Storage Area** as raw material, WIP, or finished goods by assigning the corresponding account ChartField (account, department, product, and project ID) to the storage area. All inventory locations in a storage area use the storage area account.

Location Code

Locations enable you to indicate the different types of addresses a company has—for example, one to receive bills, another for shipping, a third for postal deliveries, and a separate street address. Each of these addresses has a different location number. Every customer role must have a primary location, which will be used throughout the system on all panels that display a customer address. The primary location—indicated by a *1*—is the address you use most often when contacting the customer, and may be different from the customer's main address.

Location Summary

A Picking Plan option that sorts the picking plan according to the highest-level sort options defined and prints the order lines and the total item quantity to pick from each **Storage Location**. Because the layout of the printed report reflects the actual positions of stock to be picked, personnel can follow a serpentine path through the warehouse, fulfilling all orders on the picking plan without revisiting locations.

Lock for Confirm

A flag on the Pay Line record that enables users to access the database 7 days a week, 24 hours a day, without affecting or interrupting payroll processing. Issues a warning message "A payroll is currently in process for this employee. This data will not be processed until the next payroll."

Log file

One way that you can monitor the build process is to review the log files that the build process automatically generates. Keep in mind that the log file is entirely separate from the script file; do not confuse the two. How much information that the log file contains is up to you. You can set up your logging so that all status (both good and bad) appears in the log, or you can specify that just the errors or warnings appear in the log. This section describes the options you can specify in regards to the Build log file.

Long-Term Variable Compensation

In PeopleSoft Workforce Analytics, a component of direct compensation that consists of long-term payments to an employee in the form of stock programs, and deferred compensation.

Lookup Codes

In the financial services industry, these are user-defined codes that enable the system to define and categorize incoming Instrument table information. They also provide a means for you to report on specific data, such as treasury position, balance type, and ledger account.

Lot Status

The status assigned to a lot. In PeopleSoft Inventory, a lot's status can be Hold, Open, Rejected, or Restricted.

Lump Sum

A tax method that determines withholding based on the Canadian Lump-Sum tax table.

Lump Sum Payment Option

A form of pension payment in which some or all of a participant's benefit is paid as a single sum.

Lump Sum Reporting

A Time and Labor process that enables you to report time in a lump sum of hours or units for a single Time Reporting Code, and quantities of time. The system uses a batch process to gather the information you enter, perform edits, and update the daily time tables. The system uses the default assignments you establish for workgroups, taskgroups, shifts and so on.

M***Maintenance Worksheet***

A work space for creating write-offs, matches, or adjustments to clean up posted items.

Manage Base Pay Structure

See Base Pay Structure

Manage Compensation Planning

A PeopleSoft Workforce Rewards module that facilitates modeling and analysis of compensation costs across organization units, specific job classifications, or groups. You can focus on the impact of changes to workforce size, or on changes to fixed and variable compensation elements, and determine their effects on current and future payroll costs.

Manage Market Compensation

A PeopleSoft Workforce Rewards module you use to match your company's jobs to similar jobs found in published market compensation surveys. You then calculate a target market rate based on a weighted average from multiple surveys. This market rate is then used to assess your company's gap to market and to perform cost impact analysis.

Manage Retention Planning

A PeopleSoft Workforce Rewards module that enables organizations to analyze the factors that lead to employee turnover, and how retention of key employees affects business performance and goals.

Manual Checks

Any checks calculated and prepared outside of the PeopleSoft Payroll system that you must enter into the system manually.

Manual Count

A PeopleSoft Inventory procedure in which you enter the actual count data and then create the counting event with its header, item records, and count quantities.

Manual Events

Events that are inserted by the user manually through the BAS Activity table. Events are actions that occur, which potentially change employee benefit coverage eligibility—see Event Class for more information. Used by PeopleSoft Benefits Administration.

Manufacturing Cost Element

A particular category of an item's cost. For example, when you produce a subassembly that has a cost of \$100, the cost can be broken down further into material costs, labor costs, and overhead costs.

Manufacturing Execution Systems (MES)

Third-party system that enables detailed planning and execution of production activities from production order release to completing finished goods. PeopleSoft Manufacturing integrates to MES.

Manufacturing Task

Any job that can be performed within your manufacturing facility. A manufacturing task is associated with the work center in which the task is completed.

Map File

A file that defines the relationship between fields in a third-party system and PeopleSoft Demand Planning tables.

Mapper Type

This defines whether you are mapping actual or budgeted general ledger line items to resource ID within PeopleSoft Enterprise Performance Management.

Marginal Tax Rate

The tax rate that applies to the next dollar of income generated.

Market Compensation

A compensation review process in which you match your company's jobs to similar jobs found in published market compensation surveys, for the purpose of establishing new target market rates. Also referred to as Market Based Pricing or Market Analysis.

Market Capitalization

The value of a corporation as determined by the fair market value of its issued and outstanding common stock. It is calculated by multiplying the number of outstanding shares by the current fair market value of a share. Analysts look at market capitalization in relation to book, or accounting, value for an indication of how investor's value a company's future prospects.

Market Rate

Compensation rates, usually for regular base compensation or total cash compensation, found in published salary surveys. You use the Market Compensation module in PeopleSoft Workforce Rewards to age and weight this data, to create market rates you can compare against your organization's current pay rates.

Mark-to-Market (MTM) Model

In the financial services industry, the reevaluation of a portfolio's position at current market levels.

Market Variance

A comparison of the difference between an individual's, or group's, actual compensation, and available market compensation data for a comparable population in industry. Market compensation data is usually tied to job codes, and comparisons are usually made between similar jobs. Although the variance to market can be evaluated for any of the compensation components in the Compensation tree hierarchy (such as Total, Direct, or Base), market compensation data is most typically available for, and used in evaluating Base Pay (Base Salary). The main point of reviewing the market variance is to evaluate how well your workforce is paid in comparison to both prevailing compensation in industry, and your own organization's compensation strategy.

Mass Adjustment

A process of applying an amount or percentage change to one or many line item budgets at once. PeopleSoft Budgeting-specific.

Mass Cancellation of Requisitions and Purchase Orders

A utility that allows you to select and cancel groups of requisitions and purchase orders. You can use this utility during the year as well as at year-end in preparation for closing. The utility enables you to specify ChartField criteria for selecting documents for cancellation. For example, you can select all requisitions or purchase orders for a particular fund and organization, which have a remaining balance. Then you may select a subset of those records to approve for cancellation.

Mass Change

A user-configurable entity that defines the movement of data between the tables that store your business information. Mass Changes enable you to define the criteria by which you move or replace data in your tables. Based on the configuration of your system, Mass Change dynamically builds data access and gives you complete control over your system processing.

Mass Change Template

The foundation for defining mass changes. Mass change templates enable you to control which fields will be available for the operator to specify when defining a mass change, and whether those fields will be used as selection criteria or defaults.

Mass Change Type

The building blocks used in defining mass change templates. Mass change types specify which records the resulting mass change will select from the database, alter, and subsequently write back to the database. They also set up system field defaults that run behind the scenes to ensure that this mass change is processed correctly.

Mass Validate Metadata Utility

A PeopleSoft Enterprise Warehouse utility that enables you to validate, but not compile, Metadata objects. Mass Validate certifies all “as of dates” created for Filters, Constraints and DataSets for the specified run date. This utility helps ensure that your Metadata is valid at run time and increases your chance of a successful engine run.

Match

A process in PeopleSoft Workforce Planning, by which the system compares the roles, competencies, and accomplishments in the current competency inventory, with the requirements of a given competency strategy.

Matched Punches

A period between two consecutive punches during which some activity happens measured intervals.

Match-Funding

In the financial services industry, Match Funding refers to funding an asset with a like (term to maturity) liability. This helps an organization apply the appropriate funds transfer price. Although the actual asset might be funded with shorter-term liabilities, it does provide a better measure of financial performance for that asset, such as Risk Adjusted Return on Capital.

Material Costing

An inventory accounting method that assigns a cost to items in inventory. These costs can be assigned equally across all items or tracked individually for each item.

Material Issue

An event that triggers stock fulfillment requests for items in inventory.

Material News

Company news that could be expected to affect the value of a company's securities or influence investors' decisions. Material news includes information regarding corporate events of an unusual and non-recurring nature, news of tender offers, unusually good or bad earnings reports, and a stock split or stock dividend.

Material Release

A PeopleSoft Manufacturing process that—after material has been picked—decrements on hand inventory balances for the inventory storage areas and increments inventory to the WIP

locations defined by the routing or production area. The process also changes the production ID's or production schedule's status from Released to In Process.

MAX Method

See Maximum Method Policy.

Maximum Compensation Hours

The greatest number of hours to be paid for a specified TRC (see Time Reporting).

Maximum Lead-Time Usage

In PeopleSoft Inventory Planning, a policy control value that sets the safety stock level to the maximum quantity required during the lead time. This method is normally used when the demand for an item is low or intermittent but sufficient stock must always be available.

Maximum Method Policy

In PeopleSoft Inventory Planning, a policy that controls the way in which the system determines a reasonable high limit for the maximum inventory level of an item. The system provides warning messages when the inventory level exceeds the maximum level.

Maximum Taxable Wage Base

An annual earnings threshold used for Social Security purposes. Pension plans sometimes provide different levels of pension benefits for earnings above and below the Maximum Taxable Wage Base.

Measure

A measure represents the amounts brought into a cube—the numerical data.

In data warehousing, a Measure is a field type used interchangeably with fact. Measures are types of amounts. Any numeric field you want to apply a Data Manager rule against should be a measure.

Measure ID

In the Define Market Compensation module of PeopleSoft Workforce Rewards, a Measure ID is the identification code for a measure. For market compensation surveys, the Measure ID describes the percentile for each type of pay, as well as the regression statistic type. In PeopleSoft Workforce Analytics, for Benchmark Surveys, the Measure ID describes the type of benchmark.

Measure Value

In PeopleSoft Workforce Rewards, Measure Value is the calculated market rate value from market compensation surveys for a given percentile of a market rate, and for a given scenario and job code. This is the annual rate you compare against the compensation paid for similar jobs in your company. The Measure value can also be the regression statistic value used for

Regressing Market rates. In PeopleSoft Workforce Analytics, for Benchmark Surveys, the Measure Value is the delivered Benchmark Ratio.

Member

A member is the OLAP equivalent of a node or detail value on a PeopleSoft tree. A member is a single item within a dimension, such as a single product name, department ID, or part number. Member names must be unique, even across dimensions. Cube Manager uses the term Dimension Field Mapping to identify members, dimension parents, and label mappings.

Merchant

In PeopleSoft eStore and Mobile Order Management, a level of online (web or wireless device) display and order processing controls set by the seller. See also Merchant Variant.

Merchant Variant

Associated with a user ID, a subordinate level of merchant controls that enable customers to access different variations of the same PeopleSoft eStore website. In PeopleSoft Mobile Order Management, variations are primarily used to differentiate order processing options and fulfillment methods or locations.

Merit Matrix

In PeopleSoft Workforce Analytics, a matrix used to define the amount rules for base compensation increases for your workers. A Merit Matrix provides the salary increase parameters for each review rating in a rating scale. The salary increase parameters are expressed in terms of a percentage. The percentage increase amounts are usually structured to express the company's pay strategy relative to employee performance, and the employee's degree of range penetration in their salary range.

Merit Matrix Increase

In PeopleSoft Workforce Analytics, an increase to an employee's base pay awarded based upon a Merit Matrix.

Message definition

The object definition specified in Application Designer which contains message information for PeopleSoft's Application Messaging system.

Metadata

Information about data. Metadata is the information a database or application stores to describe your business data. At its simplest, metadata defines the structure of a data field—its data type and size, for example. Metadata can also describe more complex data relationships, such as the rollup structure for a chart of accounts. Reporting and analysis tools should be able to use this metadata to let users access data just as they would from within the application, without having to understand how it is stored.

For Enterprise Performance Management, metadata is used to describe the data stored in the PeopleSoft Enterprise Warehouse. There are different types of metadata, for example, TableMaps, DataMaps, and constraints. You typically define these when you set up the warehouse; however, Metadata (particularly constraints and DataMaps) is used to develop business rules that manage aspects of the dimensional models. Metadata enables technical users to define relationships between warehouse tables and enables business users to easily identify the data that interests them without having to know the database structure.

Meta-SQL

Meta-SQL: Meta-SQL constructs expand into a platform-specific SQL substrings. They are used in functions that pass SQL strings, such as in SQL objects, the SQLExec function, Application Engine programs, and so on.

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.

Method

A method can only be executed from an object, using dot notation. You have to instantiate the object first, before you can use the method.

For Enterprise Relationship Management, a method is the algorithm or formula that defines how the budget amount for a line-item budget is calculated or how it is derived if a calculation is not necessary. Types of methods include amount per FTE, itemization, annual percent growth rate based on a historical figure, and number of units multiplied by cost per unit. PeopleSoft Budget Planning-specific.

Method Amount

The amount for a line item budget resulting from the application of a method. It represents the budget amount after the method is applied to a line-item budget but before any adjustments or allocations are applied. PeopleSoft Budgeting-specific.

Method Base

The defined value to which a method is applied, if applicable. Not all methods require a base. PeopleSoft Budgeting-specific.

Method Driver

The factor used in a method's algorithm. For the method, amount per FTE, FTE is considered the driver. PeopleSoft Budgeting-specific.

Method of Payment

In PeopleSoft Grants, designates whether a payment is to be through a cost invoice or a Letter of Credit.

Method Parameter

A defined and derived value within a method, which drives an expense or revenue calculation. For the method, Amount per FTE, the number of FTEs is considered the driver parameter. PeopleSoft Budgeting-specific.

Metric

A metric is a calculation of facts. A metric is usually a number, but can be anything you want to measure.

Metric Object Security

Metric Object Security determines whether an individual can see a metric object in a Data Mart.

Midpoint (Pay Range Midpoint)

In PeopleSoft Workforce Analytics, the middle value in a pay range, halfway between the minimum and the maximum, calculated as $(\text{Minimum} + \text{Maximum})/2$.

Midpoint Progression

In PeopleSoft Workforce Analytics, the percentage difference from one grade midpoint to the next higher-grade midpoint, calculated as $(\text{Midpoint2} - \text{Midpoint1})/\text{Midpoint1}$.

MIN Method

See Minimum Method Policy.

Minimum Benefit

See Grandfathered Benefit.

Minimum Compensation Hours

The lowest number of hours to be paid for a specified TRC (see Time Reporting).

Minimum Method Policy

In PeopleSoft Inventory Planning, a policy that controls the way in which the system determines a reasonable low limit for the minimum inventory level of an item. The system provides warning messages when the inventory level drops below the minimum level.

Missed Punch

A punch that is not entered at the scheduled time (see Time Reporting).

mkvdk

Verity's command-line tool used to index a collection, insert new documents, perform simple maintenance tasks like purge and delete a collection, and control indexing behavior/performance.

MLS

Multilingual support.

Modal transfer

Modal transfers allow you to transfer an operator from one component to another component (the modal component) modally; that is, requiring the operator to OK or Cancel the modal component before returning to the originating component.

Modal transfers give you some control over the order in which the operator fills in pages. They are useful for finite tasks related to the main transaction. They are particularly useful in cases where data in the originating component can be derived from data entered by the operator into the modal component.

Model Equivalency Factors

In PeopleSoft Demand Planning, factors that adjust model errors to allow a fair comparison. During the Model Reset process, the errors associated with each of the models are multiplied by their associated factors. The factored errors are then compared to select the model with minimum errors.

Models

In the PeopleSoft Enterprise Warehouse, Models enable replication of an organization's business processes for analysis of cost flow through customers, departments, and channels.

Model Recalculation

In PeopleSoft Business Planning, users may checkout slices of the entire model for their appropriate role. This requires the entire model to be periodically recalculated to incorporate the users changes for dependencies in other areas of the model.

Morphing

Morphing is a technique of automatically transforming the look and feel of an interface based on the needs of an active object. The Application Designer toolbar and menus dynamically transform based upon the type of object definition that is active.

Mortality Table

A table showing rates of death by age. Mortality tables are part of a pension plan's actuarial assumptions.

Moving Average

In PeopleSoft Demand Planning, a model that averages a selected number of the most recent demand periods and creates a forecast of demand for the next and subsequent periods.

Multibook

A functionality supporting the requirement of a company to carry one set of books in their local currency (functional currency) and another set of books in the currency of their parent company (reporting currency). In PeopleSoft General Ledger, multibook functionality is multiple ledgers having multiple-base currencies defined for a business unit, and the option to post a single transaction to all base currencies (all ledgers) or to only one of those base currencies (ledgers). Also commonly known as dual-book.

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.

Multidimensional Analysis

A type of analysis that enables you to look at data from many different dimensions, or attributes. You identify the dimensions of the data, then combine the dimensions in various ways. For example, you might identify five dimensions of your sales data: sales, region, channel, product line, and time. Once you've identified the dimensions, you can "slice and dice" the data based on combinations of these dimensions, such as sales in the Western region for the last quarter.

Multidimensional Database (MDDb)

A database that stores data for multidimensional analysis in a proprietary multidimensional format. Users access MDDBs exclusively for reporting and analysis, never transaction processing, so they are optimized for retrieval speed.

For Enterprise Performance Management, a Multidimensional Database stores data for multidimensional analysis in a proprietary multidimensional format. These databases are used exclusively for reporting and analysis, and never transaction processing, so they are optimized for retrieval speed.

Multiple Jobs

Multiple jobs allow you to hire an employee into more than one concurrent job and have them processed through Payroll, Benefits, and Pension. In order to enable this feature, the Multiple Jobs check box must be selected in the PeopleTools Options page.

Multiple-table dynamic tree

The user drills down through a hierarchy of parent and child records.

Multivariate Forecasting Techniques

In Enterprise Planning and Simulation, this is a forecasting method that uses both the recorded history for the target value and the history and forecasts for other variables (causal factors) to infer, not only a forecast for the target value, but also a functional relationship between the causal factors and the target value.

N**National Association of Securities Dealers, Inc. (NASD)**

Self-regulatory organization of the securities industry responsible for the regulation of The NASDAQ Stock Market and the over-the-counter markets. The NASD operates under the authority granted it by the 1938 Maloney Act Amendment to the Securities Exchange Act of 1934.

National ID Number

Different countries track some form of National ID for payroll, identification or benefits purposes. For example, German workers are assigned a Social Insurance Number, UK workers have a National Insurance Code, and US laborers have a Social Security Number. Each of these different types of National IDs has unique formatting requirements associated with them as well.

Nature Of Action (NOA) Code

Indicates the type of personnel action being processed.

Nature Of Action Description

Describes the NOA code.

Nature Of Action Effective Date

The date the personnel action is effective.

Negative Amortization

Occurs when a loan payment does not cover the interest due on the loan payment, resulting in an increase of the principal amount.

Net-To-Zero Adjustment

A prior period adjustment where no compensation affecting fields on the pre-existing (original) record are changed by the adjustment.

New Hire Report

In the United States the Personal Responsibility and Work Opportunity Act of 1996 (the so-called Deadbeat Dads law) requires employers to report new hires to specified agencies within a pre-determined number of days from the hire date.

Next Level Item

In PeopleSoft Demand Planning, the **Forecast Item** at the next level that contains the current item as a child. This is the key of the group item at the next level up and is always within the same view.

Next Year

PeopleSoft Benefits term referring to the next open enrollment processing year.

NIC (Numéro Interne de Classement) Code

In France NIC numbers identify the entities inside the same enterprise, and represent an Internal Filing Number.

No Control

A target control that allows the user to submit a budget even if it is not within the planning target and the tolerance levels. The system tracks the budget against the defined planning targets but does not generate any warnings or validations. Users can still compare their planning targets against their budget amounts on the Planning Targets page in Line Item Budgeting.

Node

An individual item on a tree. Nodes summarize detail values or other nodes, and may or may not roll up into other nodes or levels.

Node

A node is a name that you can use to refer to some source of HTML content. In more practical terms, a node is a URI string that defines the database and server to be used when the portal servlet attempts to retrieve content, proxy addresses, and assemble pages.

Non-Base Pay

A pay component not included in the job comp rate calculation. It is used by payroll only in the paysheet calculation. For example, non-base pay can be set up for additional work, holiday pay, bonuses, and so on.

Non-Benchmark Jobs

See Benchmark Jobs.

Noncompetitive Action

An appointment or placement in a position in the competitive service that is not made by selection from an open competitive examination, and that is usually based on current or prior Federal service. A noncompetitive action includes:

- All of the types of actions described under inservice placement, above
- Appointments of non-Federal employees whose public or private enterprise positions brought into the competitive service under Title 5 CFR 316.701; and
- Appointments and conversions to career and career-conditional employment made under special authorities covered in 5 CFR 315, Subpart F.

Nondiscrimination Tests (NDT Tests)

Tests used to help employers ensure that their organization's 401(k), 401(m), and Section 129 dependent care reimbursement plans do not discriminate in favor of highly compensated employees. See Highly Compensated Employees.

Non-Employee

Those workforce resources hired to perform a specific job and/or hired for a specific period of time. Although non-employee time will be entered into Time and Labor for the purposes of managing their Task time, non-employee earnings will not be updated to Payroll and they will not be paid through the Payroll system.

Non-HR Employee [Time and Labor]

An individual employed by the corporation who is administered outside of the PeopleSoft Human Resources system.

Non-Job Event

Actions which result in changes to an employee's personal or demographic information that also affect benefit program and plan eligibility—such as an a state or postal code change, a family status change like a divorce, or a birthdate change. Used by PeopleSoft Benefits Administration. See Event Class.

Non-Productive Time

Any employee scheduled work time spent on tasks (or non-tasks) other than those which the employee was hired to perform. This could include time spent in training, time spent in meetings, travel time, and time spent reporting time.

Non-Qualified Dependent

Dependents such as domestic partners, their children, and other people who do not meet the definition of qualified dependents as presented in IRS Section 152. PeopleSoft Benefits applications enable the creation of benefit programs that offer health and life coverage to non-qualified dependents.

Nonqualified Plan

A plan that doesn't conform to ERISA rules. Employers cannot take a tax deduction for contributions to a nonqualified plan; instead, plan benefits are generally paid directly from the employer's assets.

Nonqualified Stock Option (NQ)

Any option that does not satisfy the conditions of a statutory stock option under the Internal Revenue Code and therefore does not qualify for preferential tax treatment. Generally, companies can design nonqualified options in almost any way they like. Features are:

- The grant price may be less than fair market value (with some exceptions under state law).
- Grants are not limited to employee of the company or subsidiary.
- No taxable income is recognized at the time of grant.
- Options can be granted to anyone (Employees, Consultants and Board of Directors).
- Difference between the fair market value on the date of exercise and the grant price is treated as compensation income.
- In the U.S., withholding tax obligation arises at the time of exercise.
- Company receives a tax deduction equal to the compensation income recognized.

Nontaxable Benefits

Any employer contributions that are not subject to Federal Withholding Tax, such as an employer's portion of a 401(k) plan.

Normal Form of Payment

The payment form associated with the amount calculated by the benefit formula. Pension Administration uses it as a basis for converting to optional forms of payment.

Normal Hours

The hours an employee is normally expected to be at work for any given workweek.

Normal Line Of Promotion (Career Ladder)

The pattern of upward movement from one grade to another for a position or group of positions in an organization.

Normal Retirement Date (NRD)

The date on which an employee is eligible to retire and begin receiving pension benefits. Eligibility for normal retirement is typically based on age only.

Normalized database

A normalized table adheres to certain standards designed to improve the productivity of the database user. Normalization makes the database much more flexible, allowing data to be combined in many different ways.

The standards for a normalized database are called forms, such as first normal form, second normal form, and so on.

Normalized Loss

In the financial services industry, Normalized Loss is the expected loss on a loan and is netted out of the profit and loss statement for management accounting or profitability measurement purposes. Similar to the bank's loan loss reserve, it enables the institution to analyze and account for expected losses on a more detailed level, by financial product.

Northern Ireland Report

In the United Kingdom the Fair Employment (Northern Ireland) Act of 1989 requires private sector employers with more than 10 employees to submit the Northern Ireland report to the Fair Employment Commission annually. The report indicates the religious composition (referred to as Community Background—Catholic, Protestant, Other) of the workforce, job applicants and appointees.

Not To Exceed (NTE) Date

Types are as follows:

- Appointment NTE Date: Indicates the length of time a person may serve in a position.
- Classification Temporary NTE Date: Established temporary date that is used for a temporary classification of a unique position.
- Hospitalization coverage.
- LWOP NTE Date: NTE date is the last day the employee is in leave without pay status. The employee is scheduled to return to duty the next workday.
- Position NTE Date: Indicates the length of time a position is available for use.
- Promotion NTE Date: Specific NTE Date: Specific time for an increase in grade on a temporary basis.
- Suspension NTE Date: Specific time an employee is to be on suspension. No salary is paid for the period.

nPlosion

A PS/ nVision feature that enables you to expand rows and columns in your spreadsheet to underlying details, as in drilldown.

Numeric constant

Numeric constants are any decimal number used in PeopleCode.

O**Object-Based Modeling**

Object-Based modeling technology enables you to create parent and child models. In the PeopleSoft Enterprise Warehouse, you set up such models using the Scenario Manager.

Object reference

An object reference is one that uses the current object. For example, in the case of a component, pages within the component are related objects. The menus that use the component are its object references.

Occupant Of Position/Vice

Indicates new position or former occupant of a position.

Occupational Series Code

Designates a grouping of positions similar in work and qualification requirements. They are designated by a title and four digit number (e.g., the Accounting Series, GS-0510).

Off Date

A specific date that is defined as an off day (see Scheduling).

Off Day

A 24-hour period rounded by daybreaker with no associated shifts (see Scheduling).

Off Day Type

A classification of off days (i.e. holiday, plant shutdown) (see Scheduling).

Off-Cycle Processing

The process of calculating and creating a paycheck for one or more employees aside from the normally-scheduled (*on-cycle*) payroll run for their pay group. You typically use off-cycle payroll processing for employees who are being terminated, new hires who weren't entered into the system in time for the last on-cycle payroll run, and employees who received an incorrect paycheck during a normal on-cycle payroll.

Offer Period

This is the period of time in which an employee's ESPP share price is determined.

Officer

An insider who sits on the Board of Directors and who is also an employee of the corporation. Examples include CEO, CIO, CTO, CFO, COO, Corporate Secretary, and Treasurer.

Official Forwarding Address

An employee's mailing address following separation.

Official Languages Act (OLA)

Canadian federal institutions are required to report on the official languages used in their departments, in accordance with the Official Languages Act (OLA).

Official Personnel Folder (OPF)

The repository of a Federal employee's official documents related to Personnel history.

Official Personnel Folder (OPF) Address

Indicates the address where the Official Personnel Folder is maintained.

Off-Invoice Discount (OI)

A per unit discount deducted from the customer invoice and given by a manufacturer for promotional activities. Off-invoice discounts can originate from a National Allowance or Customer Promotion, and are passed to PeopleSoft Order Management so the discounts are applied correctly during order entry.

Offset Plan

A pension plan where the benefit formula includes an offset of a portion of the participant's Social Security benefits.

OLAP

Online Analytical Processing. OLAP is the multidimensional analysis of application data, performed interactively. The acronym contrasts with OLTP (Online Transaction Processing), which is what most production business application systems do.

OLTP (Online Transaction Processing)

OLTP refers to the applications that perform the business transactions that keep your company running, such as processing invoices or enrolling employees in benefits programs.

Ontario Employment Equity Commission (OEEC)

The OEEC requires employers in Ontario to complete workforce surveys.

Open Enrollment

The scheduled annual re-enrollment of plan participants into appropriate benefit programs and, within those programs, benefit options.

Open Price

The price at which a security starts a trading day.

Open Season

A time period during which Federal employees are open to re-enroll in a specific benefit plan and option. Open Seasons can be scheduled at varying times throughout the year and multiple Open Seasons can occur concurrently with each other. For FEHB processing, it is generally the time period from mid-November through mid-December. For Thrift Savings Plan (TSP) processing, these are semi-annual and are generally held from May 15 - July 31 and November 15 - January 31. Open seasons for FEGLI are infrequent and special notification from the OPM would be issued to all Federal employees should they occur.

Open Transaction

A transaction that has not yet been processed in PeopleSoft Asset Management.

Operation

In PeopleSoft Manufacturing, a job or task performed in a specified amount of time, done in one work center, and using one or more resources.

Operational Data Store (ODS)

A staging area in PeopleSoft Enterprise Warehouse for source application data and pre-processed data for tables optimized for reporting.

OPF Code

Indicates where the OPF is maintained.

OPM

Office of Personnel Management.

Optimization

In PeopleSoft Demand Planning, the process of evaluating and improving forecast model parameters.

Optimize

The process of creating a new PeopleSoft Planning schedule by repairing the violated constraints in a schedule automatically. The Optimizer can be prioritized for meeting due dates, for minimizing overtime costs, and so on.

Option

A contractual right that gives the individual the option to purchase a specified number of shares of stock through an Equity Compensation Plan. Also known as a grant. Regulatory agencies also refer to an option as the right to purchase stock in an employee stock purchase plan. These options are considered granted on the offering begin date.

Option Adjusted Cost (OAC)

In the financial services industry, the difference in the average expected return between an instrument without embedded options that are otherwise identical to the fully loaded instrument and the instrument fully loaded with embedded options.

Option Adjusted Spread (OAS)

In the financial services industry, the average return expected for an instrument, over the short-term risk-free rate, for all projected interest rate paths generated using Monte Carlo simulation.

Option Types

Types of stock options. PeopleSoft Stock Administration supports the following stock option types. Incentive Stock Options (ISO), Nonqualified Stock Options (NQ), Tandem Incentive Stock Options/Stock Appreciation Right (ISO/SAR), Tandem Nonqualified Stock Option/Stock Appreciation Right (NQ/SAR), Restricted Stock Award (RSA).

Optional Forms of Payment

Any alternative forms of payment available to a participant retiring under a pension plan. These can include: annuity options paid over the participant's (and possibly a beneficiary's) lifetime; certain term options paid over a specified number of years; and lump sum options paid out in a single payment.

Options Outstanding

The total number of option shares held by optionees. It is the number of Grants less the number of Exercises, Cancellations, and Expirations.

Order Group

Order groups link order terms that default into sales orders and quotes when you select an order group code.

Order Line Number

The line associated with an order identification number. The order line identifies an item and the requested quantity.

Order Quantity Policy

In PeopleSoft Inventory Planning, a policy that determines how replenishment order quantities are calculated for an item. For example, you can use a static number, provide upper and lower limits, or use an economic order quantity calculated by the system.

Ordinary Income Tax

An individual's tax on earnings from wages, tips, and all other sources except capital gains. Includes option profits upon exercise of non-statutory options.

Origin ID

A code that identifies the location of a payment deposit in PeopleSoft Billing. Origin ID also distinguishes the method of the payment's entry: online, external, or lock box interface. Billing origin ID identifies the remit to origin for billing. This function is mainly used for specifying where the customer should send payment.

Original Option

A stock option that is eligible for repricing. This option has a grant price greater than the current FMV.

OSHA 200 Occupational Injury and Illness Recordkeeping Log

In the United States this record-keeping logbook meets reporting requirements for reporting occupational injuries and illnesses to the Occupational Safety and Health Administration (OSHA). It lists the case numbers and details of each injury and illness that occurred during a calendar year.

Out Punch

Indicates the end of a shift.

Out-of-the-Money

A term used to describe an employee stock option when the current market price is below the option grant price. When an option is out-of-the-money, it costs more to exercise than the underlying stock is worth. Such options are also described as being "underwater."

Output Result Tables

Refer to the database tables that are populated with information at the end of each pay calculation.

Output VAT

VAT collected on sales or outputs.

Outside Scope of VAT

A transaction determined as not subject to VAT. No VAT code is associated with this type of transaction. The transaction is still logged in the VAT transaction table, but no tax is applied.

Outside The Register Appointment

An appointment in the competitive service made under an agency's applicant supply system because either there is not a sufficient number of eligibles on the appropriate register or no competitor inventory exists. Agencies are also authorized to make temporary limited appointments outside the register at grades GS-12 and below.

Outstanding Option

A stock option that still has unexercised (vested or unvested) uncanceled or unexpired shares. Options with a "pending" status are not included. Only options with a status of 'active' or 'suspended' are considered outstanding.

Overlapping Promotions

Multiple customer promotions related to the same customer, and the same product, at the same time.

Override Rate

Cost per hour or unit reported with time used to replace the time reporter's default rate. (see Time Reporting)

Override text

Text not derived from field descriptions.

P***Package level***

The top level of organization is the package level. The package is the entire transaction set file, addressed to your company much as a mail package would be.

Page

A page defined in Application Designer as part of a PeopleSoft Internet Architecture application.

Page Assembly

Page assembly is one of the functions of the portal servlet. Page assembly involves intercepting the user's content request, retrieving the content, and properly formatting it using a pre-defined portal template. To complete the page assembly process, the portal servlet merges content from any HTML documents that it retrieves along with the defined template

HTML. The assembled page is then sent back to the user's web browser as a single HTML document.

Page buffer

Consists of rows of buffer fields that hold data for the various records associated with page controls, including the primary records, related display records, derived/work records, and translate table records.

Pagelet

A page designed to appear on a customized homepage. A pagelet is smaller than the typical page dimensions in many PeopleSoft applications. It can be based on either a page designed in Application Designer or on an iScript.

Paired Punches

Two punches for the same employee in chronological order that exists for the purpose of determining the duration between the punches.

Par Value

The nominal or face value of a security. It establishes a price floor below which shares may not be issued. With common stock, the company issuing the stock sets par value. Par value has no relation to fair market value. Some companies issue no par value stock.

Parallel Processing

In the PeopleSoft Enterprise Warehouse, parallel processing is a system function that "locks in" the information you use for processing. This enables the system to run identical or similar processes at the same time without impacting your results. Running concurrent processes greatly reduces the amount of time it takes to run within the system.

Parent Budget

In commitment control, you can build a hierarchy between different budgets, such as summary and detail budgets. Specifying a relationship of parent and child between a summary and a detail budget for purposes of budget inquiries enables you to retrieve information about either budget through the other.

Parent/Child Models

Object based modeling technology enables you to create parent and child models. In the PeopleSoft Enterprise Warehouse, you set up such models using the Scenario Manager.

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.

Parent Task

A higher-level Planning task in a schedule's hierarchy that drills down into subtasks. Its start time is the start time of its earliest subtask, and its end time is the end time of its latest subtask. Planning tasks are distinct from Manufacturing tasks.

Partial Pay

The pay processed whenever a job record has an effective date in the middle of a pay period. Typically, this happens whenever you hire, terminate, transfer, or change the rate of pay for an employee mid-period.

Participants

Individuals who elect to participate in the stock purchase plan.

Participation

The PeopleSoft Pension Administration function that determines whether an eligible employee has met the plan's rules for joining the plan. Generally, these rules are based on age and service criteria.

Participation ID

In the financial services industry, this is a lookup code used by the financial analytic applications to identify the participants (syndicators) involved in, or responsible for, a financial instrument or group of instruments.

Passive Control

A target control allowing the user to submit a budget even if it is not within the planning target and tolerance rules. The system responds by sending an email to the user of the next budget center level indicating that the budget exceeded planning target tolerance levels.

Passive Events

Events that are initiated by a change that has taken place over time, rather than by a direct data entry action. Events are actions that potentially change benefit coverage eligibility. Examples of passive events include an employee's reaching the age of retirement. See Event Class for more information.

Pattern Reporting

A Time and Labor process that enables you to report a start and stop date, a pattern of one or several time reporting codes, associated hours, amounts, or units and task information once for an employee. The system transforms the information into instances of daily time for each scheduled employee work day based on the employee's schedule.

Pay

Types of "pay" are as follows:

- **Basic Pay:** generally, the total amount of pay received during any one calendar year at the rate fixed by law or administrative action for the position held by the employee or judicial official prior to any deductions and not including any special payments or premium pay.
- **Gross Pay:** total compensation earned by an employee, annuitant, or survivor of a judicial official prior to any deductions. Includes basic pay plus locality pay; availability pay (if any) for LEOs; special payments (if any); an annuity (if any); plus awards (if any).
- **Premium Pay:** pay provided to an employee as a regular addition to basic pay (e.g., administratively uncontrollable overtime (AUO), availability pay, overtime, night differential, holiday pay, etc.).

Pay Basis

A code indicating the principal condition in terms of time, procedures or criteria, that serves as a basis for computing an employee's pay.

Pay Calculation

Formula that calculates an employee's gross to net.

Pay Calendar

Payroll processing cycle for a given pay group.

Pay Components.

Rows in the compensation record. They build the compensation packages in the compensation record.

Pay Confirmation

Process in which the system updates all to-date totals on the database for earnings, deductions, and taxes for pay groups assigned to a given Pay Run ID.

Pay Entity

A pay entity is the organization responsible for making payments to payees. You can also use a pay entity to define the type of currency to be used when processing calculations. The pay entity is a legal definition of an organization from a payroll perspective. In many cases, an organization and a pay entity are the same. However, PeopleSoft Global Payroll does not define a relationship between an organization and a pay entity.

Pay Frequency

Defines how often employees in a pay group are paid—weekly, biweekly, monthly, and so on.

Pay Group

A set of employees grouped together for payroll processing. It's a way of "bundling" payees for more efficient processing. A pay group is made of payees that the system processes at the same time during a pay run.

Pay Period

The established time segments for which employees in a pay group are paid. Pay Periods are defined by their beginning and ending dates.

Pay Plan

A code that denotes the pay schedule under which an employee is paid , e.g., JS, UG, UJ, etc.

Pay Slip

Either an actual check or an advice notice of a direct deposit. You build these to match your organization's needs. A pay slip is the details of a payment you've made.

Pay Structure

In PeopleSoft Workforce Analytics, Pay Structure consists of a series of pay ranges or grades, each with a minimum and maximum. You develop pay structures to support and reinforce your company's pay strategy (for example, to target the market 50th percentile).

Payable Date

The date that a corporate distribution, such as a dividend, is payable to the record holders of a corporation's securities.

Payable Time

Time that is ready to be collected by the payroll system (see Time Reporting).

PayCycle

A set of rules that define to PeopleSoft Payables the criteria by which it should select scheduled payments for payment creation.

Payee

Any payroll recipient. A payee can be an employee or a non-employee of an organization.

Payee Process Stat Record

A record created for each payee during the payroll process. The system creates one Process Stat record per payee for each calendar.

Payee Section

Type of section that can be added to a process list. A payee section defines a set of elements that is to be resolved for a particular payee.

Payline

Record containing standard payroll information for an employee, such as the amount of regular pay, number of regular hours, additional pay (if any), and tax information and job data.

Payment Interface

An Application Engine process that loads payment information from the Banks Statement tables and the Staging tables to the Application tables and performs various checking and default operations.

Payment Predictor

PeopleSoft Receivables' automatic cash application feature that pairs open items with unapplied payments based on predefined algorithms.

Payment Schedule

A schedule of payment dates for leased assets.

Payment Selection

A process by which PeopleSoft Payables selects scheduled payments that are eligible to be paid in a pay cycle.

Payment Worksheet

The work space in which open items are paired with unapplied payments.

Payroll Certifying Officer

The individual with the delegated authority for approving all items relating to payroll for those employees under his/her authority.

Payroll Process Tables.

Records holding data necessary to process a payroll, such as employee, company, and tax information.

Paysheets

Repository for the raw data necessary to calculate pay for employees, including earnings, hours, deductions, taxes, and accounting data.

PBGC Rates and PBGC Grading

The interest rates published monthly by the Pension Benefit Guaranty Corporation. There is an "immediate" rate that applies once benefits commence as well as a series of "graded" rates—calculated based on the immediate rate—that are used during the time between benefit determination and a deferred benefit commencement.

Pegged Chain

A method the PeopleSoft Enterprise Planning and Production Planning solvers use to determine feasible plans. The method ties tasks together in order to explicitly record which supplies are being used to satisfy which demands. Through this process, the Planning engine first determines which independent demand has the highest priority. Then, it determines the lateness preference ranking for dependent demand tasks.

Penalty

A user-assigned value for constraints that can be violated, determining how the schedule's score will be calculated. Setting the penalty configures the constraint to your priority. Use the Control Page to assign a higher penalty to violations that are more critical to your schedule or a lower penalty to constraints that you can deal with externally.

Pending Exception

Any known exception to an employee's scheduled workday. Pending Exceptions are future dated (future is defined to be for a date under report beyond the last date of the employee's current pay period).

Pending Item

Information in PeopleSoft Receivables that has been entered in or created by the system, but hasn't yet been posted. During the Receivable Update process, the system uses the pending items to update customer balances—either by creating new items or by adding item activity lines to existing items.

Pending Time

Time that has been reported or is assumed to have been reported (based on employee work schedule and calendar date) that has not been used by the business entity. Pending Time may be for past, current, and future pay periods. It is the label for those time transactions that are waiting to be used by the business (for example, approved and unapproved time not yet updated to Paysheets).

Pension Status

An employee's standing with regard to a particular pension plan. For example, employees can be active participants, terminated deferred vested, or in pay status.

PeopleCode

PeopleSoft's proprietary language; it is executed by the PeopleSoft Application Processor. PeopleCode generates results based upon specific actions, based upon existing data or the

actions of a user. Business Interlink Objects are executed by calling the `execute()` method from PeopleCode. This makes external services available to all PeopleSoft applications wherever PeopleCode can be executed.

PeopleCode Event

An action that an end-user takes upon an object, usually a Record Field, that is referenced within a PeopleSoft page.

PeopleSoft Activity-Based Management (ABM)

A PeopleSoft Analytic Application that aligns organizational costs with operational activities, enabling a coordinated approach to expense and PeopleSoft Activity-Based Management. PeopleSoft Activity-Based Management identifies and assigns operational activities to products, customers, or services.

PeopleSoft Analytic Applications

These are applications within Enterprise Performance Management (EPM) that help you enrich the data in the PeopleSoft Enterprise Warehouse and perform forward looking simulations and scenarios. These applications include: PeopleSoft Activity-Based Management (ABM), PeopleSoft Asset Liability Management (ALM), PeopleSoft Balanced Scorecard (BSC), PeopleSoft Funds Transfer Pricing (FTP), PeopleSoft Risk Weighted Capital (RWC), and PeopleSoft Workforce Rewards. PeopleSoft Funds Transfer Pricing and PeopleSoft Risk Weighted Capital are applications that target the financial services industry (FSI).

PeopleSoft Asset Liability Management (ALM)

PeopleSoft Asset Liability Management provides financial service institutions with the analytical tools to define, measure, monitor and manage interest rate risk, liquidity risk, options risk, and to some extent exchange rate risk. The primary audience for PeopleSoft Asset Liability Management is the financial institution's Asset/Liability Committee (ALCO).

PeopleSoft Balanced Scorecard (BSC)

PeopleSoft Balanced Scorecard converts an organization's vision and strategy into a comprehensive set of performance and action measures that provide the basis for a strategic management system.

PeopleSoft Budgeting

A budgeting application that is a combination of Education and Government (E&G) Budget Planning and Budgets (commercial). This application resides on the EPM database, primarily using the ODS layer of the PeopleSoft Enterprise Warehouse for its data.

PeopleSoft Business Analysis Modeler (BAM)

A multi-dimensional modeling tool used to support several analytic applications.

PeopleSoft Business Planning

A planning application that enables financial executives to model various alternatives and set corporate financial targets to achieve their strategic goals. PeopleSoft Business Planning integrates with PeopleSoft Analytic Forecasting, PeopleSoft Activity Based Management, PeopleSoft Workforce Analytics, and PeopleSoft Budgeting applications.

PeopleSoft Customer Behavior Modeling

A PeopleSoft application that enables you to: a) create a customer profile by extracting customer data from the Enterprise Warehouse; b) create segments and samples from the profile to efficiently target marketing campaigns and further analyze customer behavior; c) append to it additional data from external sources such as demographic, credit or psychographic information; d) use a data mining tool to create a predictive model; e) score the customers in your profile using the predictive model. You can then publish the results to another transactional application.

PeopleSoft Customer Scorecard

A product that provides a pre-defined set of customer-oriented key performance indicators (KPIs), to help you build a scorecard specific to your organization. This scorecard facilitates the measurement and communication of customer satisfaction, customer activity, and objectives across your organization.

See also PeopleSoft Balanced Scorecard

PeopleSoft Enterprise Performance Management (EPM)

Enterprise Performance Management is a comprehensive, integrated analytic business solution designed to increase the value of organizations by enabling people to make better decisions. The PeopleSoft Enterprise Performance Management product line consists of the PeopleSoft Enterprise Warehouse and optional analytic applications and Data Mart products.

PeopleSoft Enterprise Warehouse (EW)

PeopleSoft's data warehousing solution. The PeopleSoft Enterprise Warehouse provides the tools necessary to query, analyze, and present information to provide the optimal environment for business intelligence. It is the central repository for data that will be used with the analytic applications in the Enterprise Performance Management product line, and can also serve as a standalone data warehouse. The PeopleSoft Enterprise Warehouse consists of dimension, fact, reference, and error tables, reporting and ETL tools (Informatica PowerMart), as well as the Operational Data Store (ODS). The tables in the PeopleSoft Enterprise Warehouse are maintained separately from your transaction-based systems to allow for comprehensive analysis of data originating from any Online Transaction Processing (OLTP) or legacy system.

PeopleSoft Funds Transfer Pricing (FTP)

A PeopleSoft Analytic Application that enables an institution to accurately measure and tune profitability. PeopleSoft Funds Transfer Pricing is an interest rate that represents the value of an asset or liability to the institution. PeopleSoft Funds Transfer Pricing is based on market rates, adjusted for risk and cost variables, specific to the institution. By assigning PeopleSoft Funds Transfer Pricing to each item on the balance sheet, the institution can remove the

effects of interest rate volatility from business units, so that profitability measurements are based on factors within their control, that is, credit quality, pricing and product strategy.

PeopleSoft Operations Data Stores (PODS)

See Data Warehouse Tables

PeopleSoft Marketing Insight

A tool that helps you analyze your marketing campaigns and activities. It helps you determine the effectiveness of marketing events based on factors such as number of leads generated, profiles of respondents, campaign return on investment, and campaign forecasted costs to complete.

PeopleSoft Risk Weighted Capital (RWC)

A PeopleSoft Analytic Application that enables the financial services industry to accurately measure capital that has accounted for risk. RWC allocates capital to various levels within a financial institution according to risk, providing the opportunity to measure performance based on how well each business unit, product, customer, or transaction generates income given its perceived level of risk as quantified by the allocation of capital.

PeopleSoft Sales Activity Insight

A tool that helps you analyze key components of the sales process, such as pipeline status, discount analysis, and sales process.

PeopleSoft Support Insight

A tool that helps you determine the effectiveness of your customer service organization. It helps you answer questions such as: Are we effectively handling customer issues? Has contact center performance changed from last year to this year? Which product quality issues are most prevalent?

PeopleSoft Workforce Analytics (WFA)

PeopleSoft's complete workforce analysis solution, which includes the PeopleSoft Workforce Rewards analytical application and the PeopleSoft Workforce Analysis Insight. The complete solution set includes the PeopleSoft Enterprise Warehouse and PeopleSoft Balanced Scorecard products. It helps to manage strategic employee compensation, goals, and competencies, as well as retention.

PeopleSoft Workforce Rewards

PeopleSoft Workforce Rewards is an analytical application you use to align your workforce compensation and retention initiatives with the strategic objectives of your organization. Modules include Market Compensation, Base Pay Structure, Compensation Planning, Workforce Simulation, and Retention Management. You integrate data from multiple internal and external sources, enrich it using rules you define based on any data in your PeopleSoft Enterprise Warehouse, and simulate multiple scenarios of future workforce compensation and

retention activity. You then analyze and evaluate your scenarios, and make actionable decisions you can communicate back to your PeopleSoft ERP systems for execution.

PeopleSoft Portal

The portal bundled with every PeopleSoft 8 application. It provides a simple navigation system, based on existing menu definitions that have been imported into the portal registry. Navigation to content outside of PeopleSoft applications is not provided.

Percent Cycles Without a Shortage

In PeopleSoft Inventory Planning, a method used with safety stock policies. The value is derived from the percentage of replenishment cycles that will complete without a stockout.

Percent Demand Fill

In PeopleSoft Inventory Planning, a method that can be used with safety stock policies. This method defines the percentage of the total quantity ordered that must be filled without a backorder.

Percentage Tolerance

The acceptable percentage difference between expected cycle count quantities and actual quantities counted in PeopleSoft Inventory. This value provides a margin of error for an item during cycle count reconciliation count quantities.

Performance Appraisal Code

Indicates the level of performance of an employee.

Performance Appraisal Due Date

Date established based on the WGI or LEI for the yearly appraisal of an employee.

Period Closing Offset

In Time and Labor, the closing date beyond which this pay period is not considered current any longer, if the period's closing date is different from its end date. You can enter a positive or negative number of days.

Period Method

In PeopleSoft Inventory Planning, a method used to determine how a single static policy value is to be calculated from time-phased results with static policies.

Period of Interest

The maximum period of time containing the data needed to run all the rules in a rule program (see Batch Processing)

Period Segmentation

When an element (like compensation rate) changes mid-period, requiring all other elements in the process list to be calculated multiple times on either side of the date on which the change took place, period segmentation is used. The system calculates each element more than once, using the components that were effective during the different time slices. The system keeps the results of these calculations separate with the object of creating two gross-to-net result sets.

Periodic Processing

In PeopleSoft Pension Administration, any of several batch processes that a plan administrator must run on a regular basis—for example, consolidation of payroll data.

Personal List

A user-created list of products defined in PeopleSoft eStore, used to quickly populate the shopping cart when creating a new order in either PeopleSoft eStore or Mobile Order Management.

Personal Register (Registre Unique du Personnel)

In France, companies are required to be able to produce, at any given time, a Personal Register. For a given establishment, this report lists current employees and employees who left up to 5 years ago.

Personnel Action

Personnel actions are changes to employee data or status resulting from such activities as promotions, transfers, terminations, salary increases, and leaves of absence.

Personnel Representatives (Délégués du personnel)

In France it is mandatory for companies with more than 11 employees to elect personnel representatives who will represent all of the employees before management.

Perspective

In PeopleSoft Balanced Scorecard, a category for organizing critical success factors and key performance indicators. Usually there are four: financial, customer, internal process, learning and growth.

PF Ledger

The PF Ledger (PF_LEDGER_F00) is an important fact table within the PeopleSoft Enterprise Warehouse. The primary function of the PF Ledger table is to support PeopleSoft Enterprise Performance Management reporting. The data that gets posted to the PF Ledger must be accurate and clean.

PF Business Unit

PF Business Units differ from other PeopleSoft Business Units in that they represent functional or strategic areas of an organization, rather than separate legal entities.

Phase Type

A label for the different phases you want to define for a project. Costs can then be calculated by project phases. Examples include planning, startup, construction, and cleanup.

Physical Accounting

The PeopleSoft Inventory feature that updates tables based on count result input, regardless of how the count was created or the data collected.

Physical Inventory Process

In PeopleSoft Asset Management, the process by which you extract asset data from the Asset Management database to load into your bar code scanning device. You then scan the assets and load the data gathered during the actual physical inventory into Asset Management, enabling you to generate physical inventory results for review. You perform matching and generate transactions to reconcile the data in Asset Management with the results of your physical inventory.

PIA

PeopleSoft Internet Architecture. This is the fundamental architecture on which PeopleSoft 8 applications are constructed, consisting of an RDBMS, an application server, a web server, and a browser.

Piece Work

Method of compensating time reporters based on units completed rather than hours worked

PIN

Technical term for an element. In PeopleSoft Global Payroll, PIN is often referred to in the online object names and within the batch code. PIN stands for Pay Item Name.

Placeholder

A temporary location designator in an engineering bill of material for a component item that has yet to exist. These temporary placeholders have to change into approved items before transferring engineering bills of material (EBOM) to manufacturing bills of material (MBOM).

Plan Administrator

The person selected by the employer to perform the administration of a plan under PeopleSoft Pension Administration.

Plan Eligibility

The PeopleSoft Pension Administration function that uses job data to determine whether an employee may participate in a pension plan. An employee can be eligible based on job data but not be participating because of an unmet service or age requirement.

Plan Type

A unique ranges of codes used during payroll calculation to determine deduction processing rules. See also Benefit Plan Type.

Plan Year

The annual period that a pension plan uses to measure service, earnings, and benefits. Generally, the pension plan year will match the fiscal year of the plan sponsor.

Planning Item

A non-inventory item designated as being used for planning purposes only. It can represent a family or group of inventory items. It can have a planning bill of material or planning routing, and can exist as a component on a planning bill of material. A planning item cannot be specified on a production or engineering BOM or routing, nor used as a component in production. Quantity on hand will never be maintained.

Planning Level

The level on a dimension's tree used for planning. Typically, a customer will choose not to plan at the lowest level of available detail, such as the individual product level. Instead, the individual products are mapped to their corresponding product group and the planning is done at the product group level.

Planning Targets

The amount the budget must equal, such as a budget spending limit or cap for expenses where users can not exceed the amount. Planning Targets are presented at a summary level. This term is interchangeable with Spending Limits.

Planning Target Tolerance

The percentage and/or amount a user can be over or under the planned budget target.

Planning Target Control

Values set at the user role level, including no control, active control, and passive control. For more information see no control, active control, and passive control.

Planning Task

Any activity in PeopleSoft Planning that creates a schedule.

PODS (PeopleSoft Operations Data Stores)

See Data Warehouse Tables

POI. Personnel Office Identifier.

Also known as Submitting Office Number (SON). These are codes assigned by the OPM to the office(s) delegated authority within an agency to process personnel actions on Federal employees.

Pointers

A pointer is an "address" of a driver quantity, or value, within the Enterprise Performance Management product line. Pointers are used as a means of defining where driver quantities exist in tables that reside in the PeopleSoft Enterprise Warehouse. Pointers enable you to extract values from any location in the warehouse and then use these values as driver quantities. There are three different kinds of pointers: explicit, implicit, and multidimensional.

Policy Control Group

In PeopleSoft Inventory Planning, a feature for setting up order quantity, safety stock, reorder quantity, and maximum and minimum policies. The **Control Group** is assigned to a set of **Planning Item**. The policies of the associated planning items can be set explicitly or defaulted from the policies on the policy control group.

Policy Generation

In PeopleSoft Inventory Planning, a set of run options used to control the functions and behavior of the Policy Generation program.

Policy Item

An item record which is related to a location and for which **Inventory Policy** is held. A policy set, **Planning Item ID**, and Location ID uniquely identify a policy item. The combination of an item and a location is called a stockkeeping unit.

Policy Set

Defines a set of the items for which **Inventory Policy** is to be calculated. Each policy set is assigned a unique ID and includes information that defines, for example, the associated **Forecast View**, time periods, and planning horizon.

Policy Simulation

In PeopleSoft Inventory Planning, a feature that simulates the effects of various stocking scenarios, compares current policy with simulated policy, and determines the best inventory investment strategy.

Population

A Population is defined on top of DataMaps created using Enterprise Warehouse metadata. The Population builder allows you to easily format a SQL statement, using filters, to select

rows from one or more tables joined together in a DataMap. You can view the results of a Population directly from the browser.

Pop Up Payment Option

A variation on a joint and survivor payment option under which the benefit payable to the participant is increased if the beneficiary should die prior to the participant.

Portal

A portal is a web site that helps you navigate to other web-based applications and content. Users often consider a portal their “entry point”—the place they typically visit first after launching their web browser.

Portal Registry

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.

Portal Registry API

The Registry API is provided for accessing each portal registry from PeopleCode, COM, Java, or C programs. Providing the same kind of registry management capability as the online administration pages, it can be used by external systems to update the registry to reflect changes in the content reference URL, taxonomy, and effective dates. The Registry API is fully described in the PeopleCode documentation.

Portal Servlet

A Java servlet that runs on a web server. The portal servlet intercepts user requests for content, retrieves content, and builds a single HTML document to be displayed in the user’s browser.

Portal Solutions

Portal Solutions are separate product offerings from PeopleSoft that consist of pre-built, packaged solutions focused at different audiences (customers, suppliers, and employees). Because they are both pre-built, supported application products, Portal Solutions can be deployed swiftly and easily, saving significant resources when compared to other custom-built solutions.

Position

The officially assigned duties and responsibilities that make up the work performed by an employee. Positions are linked to Job Codes, which can be considered the electronic version of the Position Description. There can be a many-to-one relationship between the Position and Job Code.

Position Budgeting

The budget amounts (salary, benefits, and earnings) associated with positions within an organization. Position budgeting can be calculated based on position information loaded from a human resource system. Position budgets are used to generate line-item budgets for personnel costs. PeopleSoft Budgeting-specific.

Position Change

A move by an employee to another position during the employee's continuous service under the same appointment within the same agency.

Position Date Created

Date the position was created for use in the agency.

Position Description (PD)

In accordance with OPM guidelines, an official description, authorized and approved by an agency official, describing duties and responsibilities to be performed. Position classification standards are used to describe the work, classify the work components by occupational series, and factors (e.g. supervisory control, scope, complexity, competencies required) are used to determine the grade level (i.e., salary range) for the position.

Position Description Number

A number assigned to identify various types of Position Descriptions.

Position Description Required

Identifies those positions for which a position description must be maintained.

Position Number

A number that identifies an authorized Position.

Positive Input

Data such as hours worked or a bonus amount entered for elements that change each pay period. Positive input can be entered manually, generated by the system, or received from other applications.

Positive Task Reporting

A method of time reporting in which all required task elements must be provided (see Time Reporting)

Positive Time Reporting

A method of time reporting in which all elements of time must be provided (see Time Reporting)

Post Differential Percent

Additional compensation that may be paid to certain employees who work in Guam or the Northern Mariana Islands.

Post Differential, Non-Foreign

A differential payable to an employee at a location in a non-foreign area if conditions of environment differ substantially from conditions of environment in the contiguous United States and warrant its payment as a recruitment incentive.

Post-56 Military Deposit

The OPM provides guidelines to Federal agencies on how to calculate and process these voluntary employee deductions from pay toward the employee's current retirement fund for those periods of eligible military service.

Posting

In PeopleSoft the process by which accounting entries are created or updated based on user transaction input and accounting entry templates. In PeopleSoft Receivables, posting is also known as Receivable Update.

These posted accounting entries in the feeder systems, such as accounts receivable or accounts payable, must be further processed by the Journal Generator to create journal entries before posting to the General Ledger occurs.

Post-Tax Deductions

See After-Tax Deductions

Pre-encumbrance

An encumbrance that occurs before an employee/employer relationship exists. You encumber funds for an employee you have on staff; you pre-encumber funds for an employee that you anticipate hiring. For example, you would pre-encumber funds for a new position that has just been approved but not filled.

Pre-encumbrance Ledger

Stores pre-encumbrance amounts. Updated by posting pre-encumbrances, such as purchase- or hiring-requisition source transactions (including journal entries).

Premium

Any additional compensation for extra hours worked, often expressed in terms of factor-above-normal-per-hour pay, such as time and a half (where one-half is premium pay), double time or triple time. Also, any additional pay provided to a time reporter based on compensation rules (see Time Administration in your *PeopleSoft Time and Labor* PeopleBook).

Prenote

A prenotification or waiting period requested by banks before processing payroll direct deposits.

Pre-Retirement Survivor Annuity (PRSA)

A benefit paid to a beneficiary if a pension plan participant dies before commencing benefits. Qualified plans must offer a pre-retirement survivor annuity, although the employee can be required to pay for the coverage with a reduction in the benefit.

Pre-Tax Deductions

See Before-Tax Deductions

Previous Day's Close

The previous trading day's last reported trade.

Price Break

Defined in PeopleSoft Order Management, and linked with price sets, a price break defines the actual adjustments that are made to the list price. They are valid only within a time frame you establish.

Price Rule

Used in conjunction with price sets, rules are essentially a decision tree that establishes the search order the system uses in reviewing sets for a match on the variables they reference for price adjustments.

Price Set

Linked with price breaks, a price set specifies the parameters for your price adjustment. It establishes selection criteria, determines whether the break is based on quantity or price, and defines how the adjustment is applied.

Price Source

A service provider or publication that reports the trading activity for a stock traded on stock exchanges. Examples include Wall Street Journal and Bloomberg.

Primary event code

Primary event codes, also called purpose codes, specify the status of the transaction: whether it's a new transaction, a cancellation, a duplicate, a status request, and so on. Every transaction has a primary event code assigned to it.

Primary Insurance Amount (PIA)

The benefit amount calculated under the Social Security benefit formula.

Primary scroll record

Primary scroll records are the principal SQL table or view associated with a page scroll level. A primary scroll record uniquely identifies a scroll level in the context of its page: each scroll level can have only one primary scroll record; and the same primary scroll record cannot occur on more than one scroll at the same level of the page.

Prior Period

In Time and Labor, any payroll period before the current one.

Prior Period Adjustment

A change or correction to previously reported time or task information, or an insertion of time or task information. Often requires the original report to be offset (reversed) and the correct information to be recorded. (see Time Reporting)

Priority Rank

The numeric value assigned to inventory **Demand Priority Rules**. The lower the number, the higher priority of the rule and the orders matching that rule.

Private

A tracking method used by a privately held company to track their daily prices. The Board of Directors typically establishes a price for a period of time. Stock of a privately held company is not traded on an exchange.

Private Views

User-defined views available only to the user who created them. For more information, *see* Budget Views.

Process

See Batch Processes.

Process Definition

Process Definitions are created in the Process Scheduler Manager pages to define each specific run request. A Process Definition is comprised of a variety of variables including panels associated with a request, Process Groups, output parameters, run locations, and many more.

Process Group

Used to associate specific Process Definitions with a Class Profile in Security Administrator. This allows administrators to restrict an operator's ability to initiate requests.

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

Multiple Process Definitions can be logically linked into a job request to process each request serially or in parallel, and optionally initiate subsequent processes based on the return code from each prior request.

Process List

The set of instructions the system uses during a payroll process to determine which elements to resolve. A process list is comprised of sections that identify the sets of elements to be resolved. You build process lists and attach them to calendars.

Process List Manager

The program used during batch processing that reads the Process List and calls the PIN Manager to resolve elements on the list.

Process Request

A single "run request," such as an SQR, a COBOL program, or a Crystal report that you run through Process Scheduler.

Process Run Control

A PeopleTools variable used to retain Process Scheduler-defined values needed at runtime for all requests referencing a run control ID. This is not to be confused 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.

Process Scheduler

A PeopleTool that performs tasks behind the scenes of your application. It can run several kinds of processes, such as COBOL, SQR, and Application Engine programs. You can schedule processes to run on a regular schedule or at your request. Processes can run on your workstation or on a server.

Process Scheduler Server Agent

The server-based program (PTPUPRCS) that manages the selection, validation, and initiation of all queued requests for each defined server within your batch environment (Process Scheduler).

Process Type

A global process definition under which related process definitions are grouped. This allows for easy maintenance of Process Definitions that share common parameters.

Processing group

In order to partition application processing between the client and the application server, it is necessary to define units that, as a whole, run in one location or the other. We call these units processing groups.

Processing groups can encompass one or more PeopleCode events. Some processing groups can run either on the client or on the application server, such as Component Build and Component Save.

Product

A commodity primarily defined in PeopleSoft Order Management. It may be: 1) The Order Management view of an inventory item that has attributes the same as or different from those of its inventory counterpart. 2) A commodity that is not a stocked inventory item such as a product kit or a service. 3) A tangible commodity that is drop shipped from another vendor and is never stocked in inventory.

Product Alternate

Alternative products that can replace the product ordered when it's out of stock or a problem with a particular product temporarily prevents shipment.

Product Catalog

A configurable list of available products that may be of interest to a specific customer. You can create two types of catalogs—inclusive catalogs that contain all the products you want made available to a customer and exclusive catalogs that contain the products you want to withhold from your customer. By attaching product catalogs to a Sold To customer, you define the products they can buy.

Product Definitions

This support module describes how other support modules process the instruments that belong to a particular product. For example, different products have different cash flow characteristics and may be stratified differently, or may react differently to changes in interest rates. This module enables you to specify each of these actions for each instrument.

Product Kit

A commodity that consists of a fixed set of components that are sold together. It appears as a single line on an order, but is represented by multiple lines on shipping documents. Product kits may comprise inventory items, non-stockable products, or a combination of both.

Product Kit Component

A commodity that is part of a product kit. It may be an inventory item or a non-stockable product such as a service.

Product Pricing Model

In the financial services industry, this defines models that describe indices upon which future rates are paid or charged for an individual product.

Product Tree

A user-defined graphical representation of a company's product structure. A product tree defines how products are promoted and determines what users have authority to promote those products.

Production Maintenance Spreadsheets

A set of spreadsheets generated by an nVision process, containing production ID and production schedule data extracted from PeopleSoft Production Management. You can add and maintain production quantity data using these spreadsheets and then import the data back to Production Management.

Production Option

Effective-dated combinations of BOM codes and routing codes. You can create multiple effective-dated BOM code/routing code combinations (or production options) for an item. These combinations can be extracted to PeopleSoft Production Planning. They enable the specification of multiple production variations for an item and provide control of seasonal variations by effective date.

Production Option Cost

A cost based on a specific BOM/routing combination (also known as a production option). The Production Option Cost utility rolls up production options costs based on specific BOM/routing combinations, enabling you to cost individual production options and later to have the DataLink pass this cost to the Planning engine. If this utility isn't run, the Planning engine will use the standard item cost based solely on the primary BOM and routing instead.

Productive Time

Employee scheduled time spent performing any task for which a position was created; work performed on behalf of a business entity that is required for that entity to fulfill its business purpose. Employees doing the work they or someone else was hired to do.

Productive Unit

In Italy employers organize employees into productive units based on agreements between the unions and the employer.

Profile

A data set that you aggregate from the Enterprise Warehouse, according to the filters you specify, the Key Performance Indicators you select, and the 3rd party demographic data you include.

Profile Factor

In PeopleSoft Demand Planning, the weight index assigned to each **Forecast Period** to take into account seasonal fluctuations in demand. The factor or index typically measures the percentage of difference between the base demand and the expected actual demand in the period.

Profiles

Group of employees defined according to a list of job codes and departments. You can use these profiles to ascertain training demands within your organization based on set criteria.

Profit Manager

The Profit Manager is a set of integrated tools that enable true profitability reporting. Profit Manager features are tightly integrated with the PeopleSoft Analytic Applications and provide you with ways to ensure data integrity, edit data, and post data to the Performance Ledger table.

Project

The highest level of hierarchical organization within PeopleSoft Projects. Projects provide the structure to which activities and resources are added. Each node on a Projects tree represents a project. Projects can contain other projects as well as activities and resources. This provides a hierarchical relationship between projects and facilitates cost roll-ups.

In Enterprise Performance Management you use a Project to create or modify a Profile. A Project contains pointers to data elements that you include in a Profile.

Project

In PeopleSoft Time and Labor, a specific endeavor undertaken to achieve a specific goal. Typically, projects are approved and undertaken with level of cost, schedule, and performance already agreed upon. A project is composed of a set of tasks, each of which requires staffing, provisioning, and/or scheduling. Project progress is often measured in terms of task completion.

Project ID

The name or number by which a project is to be identified in all tables and pages.

Project Type

A user-definable grouping of projects. Project types are optional.

Projected Run Date

In PeopleSoft Demand Planning, a calculation made that projects a life volume for a period based on a calculated run rate or performance ratio.

Projection.

An estimated pension benefit calculated as of a future date or any estimated data used as the basis for such a calculation.

Promotion

- For positions under the same type job classification system and pay schedule, a promotion changes the employee to a higher grade level or makes permanent a Promotion NTE;
- When the old and new positions are under different job classification systems and pay schedules, a promotion changes the employee to a position with a higher rate of basic pay or makes permanent a Promotion NTE.

Promotion Pattern

In PeopleSoft Demand Planning, an **Event** function that enables you to apply weights to promotions across a range of **Forecast Period**.

Prompting Profile

A task profile usually used by account managers as a way of creating task profiles for employees who report task time differently by customer. For example, you might have an account manager who has fifty customers; when the account manager comes in each day to report time, the system will display all the customers, and indicate which customers it will use as a default if she doesn't manually report time.

Prorated

In Enterprise Planning and Simulation, prorated is when the computed forecast and the summarized forecast are two different versions of the statistical forecast. In addition, the forecast at the product family level can be allocated down to the individual products. Usually this allocation is done in proportion to the calculated product forecasts at that level. This version of the (statistical) forecast is called the allocated or prorated statistical forecast.

Pro-Rate Purchase

A purchase in which the number of shares to be purchased is prorated according to a specified factor. This may occur when the total number of shares to be purchased is greater than the number of shares allocated to the stock plan from the treasury.

Prorated Forecast

In PeopleSoft Demand Planning, a forecast developed by factoring the group forecast down one level at a time to make the sum of the item forecast equal to the aggregate forecast. The prorated forecast tends to be more accurate than the **Statistical Forecast**.

Proration Rule

Element that defines how you want to prorate an item. You use proration rules in numerous places—for instance you could prorate an earning, deduction, or many of the elements that make up an earning or deduction.

Proration Threshold Ratios

In PeopleSoft Demand Planning, the upper and lower ratios used as thresholds for **Reasonableness** checks when a forecast is developed using proration.

Provider

An entity, such as an insurance company, that provides one or more of the benefits your company offers. For example, Metropolitan Life Insurance Company is a provider to companies that use a Metropolitan life plan.

Proxy Person

A highly compensated executive. Corporations must include information regarding the most highly compensated executive officers in their proxy reporting.

Proxy Statement

The document that must accompany a solicitation of proxy appointment under SEC regulations. The purpose of a proxy statement is to provide shareholders with the appropriate information to make an intelligent decision.

PSADMIN

A PeopleSoft utility providing a menu interface to create, configure and administer application server domains and the Process Scheduler Server Agent (Batch Server).

PSADMIN

A PeopleSoft utility providing a menu interface to create, configure and administer application server domains and the Process Scheduler Server Agent (Batch Server).

PSAPPSRV

PSAPPSRV is the main server process running within a domain. PSSAPPSRV performs the functional requests, such as building and loading components. It also manages the memory and disk-caching for PeopleTools objects on the application server. Each PSAPPSRV process maintains its own memory and disk cache.

It provides authentication services for incoming users. For instance, it checks the PeopleSoft OPRID against the directory server or PSOPRDEFN table.

PSQCKSRV

Essentially, PSQCKSRV is a copy of the PSAPPSRV. It performs quick, read-only SQL requests. It is an optional Server Process designed to improve performance by handling items in the PSAPPSRV transaction request queue.

PSQRYSRV

Like the PSQCKSRV server process, PSQRYSRV is designed to alleviate the workload of PSAPPSRV. PSQRYSRV is designed to specifically handle all user-generated queries submitted by PeopleSoft Query (PSQED.EXE). This server process is designed to improve overall application server performance whether or not you have PSQCKSRV configured. It is specifically, and exclusively designed to process PeopleSoft Query transactions, which can be very SQL intensive.

PSSAMSRV

It processes conversational SQL transactions primarily for Application Designer.

Public Company

A company that has held an initial public offering and whose shares are traded on a stock exchange or in the over-the-counter market. Public companies are subject to periodic filing and other obligations under the federal securities laws.

Public Views

Coordinator-defined views, available to anyone using the application. For more information, *see* Budget Views.

Publish/Subscribe

Publish/Subscribe type messaging is performed with PeopleTools Application Messaging technology. You can send data from one PeopleSoft system to another in an asynchronous mode—meaning the two systems don't have to be sending and receiving at the same time. This is possible because the message transfer is accomplished through a Web server with an "http: gateway."

Pull List

Similar to a pick list, a pull list contains multiple replenishment requests, including the location, quantity, and item quantity required in a specific sorting sequence. You use pull lists in PeopleSoft Flow Production with Inventory replenishment.

Pull Ticket

A document containing the details of a single request replenishment request, including Kanban ID, item, quantity, source, and To locations. You use pull tickets in PeopleSoft Flow Production with Inventory replenishment.

Punch

Precise instances of date and time recorded for a user and measured in seconds, minutes, hours, day, month and year and time zone (see Time Reporting)

Punch Duration

Length of time between two punches in increments of hours or partial hours (see Time Reporting)

Punch Matching

Area of the application which converts paired punches to punch duration by processing rounding rules and assigning the tasks to the appropriate logical day based on rules established by the user

Punch Restriction

The facility to constrain a time reporter's ability to create a punch that deviates from the schedule (see Time Reporting)

Punch Type

A user defined classification of punches, i.e. In, Out, Start, Stop (see Time Reporting)

Purchase

The issuance or purchase of shares through a stock purchase program. The purchase is made using current contributions from a participant and any carry-forward remaining for the participant from previous purchases.

Purchase Price

The discounted price paid for the shares at the end of a purchase period.

Purchase Price Variance

A PeopleSoft Payables matching feature that compares purchase order and inventory prices for any variance in the prices.

Purge Rules

The rules that define criteria to clear data you no longer need from previous open enrollment processing cycles in PeopleSoft Benefits Administration.

Pyramiding

A computer calculation enabling an individual owner of one share of stock to use the stock-swap technique to exercise a stock option of any size without using cash. Not many corporations permit pyramiding.

Q

QDRO

See Qualified Domestic Relations Order.

QDRO Alternate Payee

A former spouse who is entitled to a portion of a participant's pension benefits as a result of a court order.

QJSA (Qualified Joint and Survivor Annuity)

A post-retirement death benefit for a spouse. Plans subject to this requirement must provide an annuity for the life of the participant with a survivor annuity for the life of the participant's spouse.

QMCSO (Qualified Medical Child Support Order)

A QMCSO is a court order that requires a group health care plan to provide benefits to the child of a participant as part of a child support arrangement on the behalf of that participant. Base Benefits enables the tracking of QMCSOs for dependents.

Qualified Domestic Relations Order (QDRO)

A court order ordering a division of a participant's pension benefits. This is normally the result of a divorce and gives a portion of the pension benefits to the former spouse.

Qualified Plan

A pension plan for which the employer can take tax deductions for contributions to the plan. Investment income of the plan trust fund is not taxable to the employer. Tax law places restrictions on the plan rules.

Qualifying Dispositions

A transaction whereby a participant sells shares acquired through a stock purchase plan two years after the grant date and one year after the purchase date.

Quality Function

A level of configuration that enables you to define the fields and attendant information that provides a base level for inspection plan and integration. Quality functions enable you to map process-specific field information into PeopleSoft Quality for identification, tracking, and analysis.

Quality Ranking Factors

Knowledge, skills, and abilities that could be expected to enhance significantly performance in a position, but are not essential for satisfactory performance. Applicants who possess such

KSAs may be ranked above those who do not, but no one may be rated ineligible solely for failure to possess such KSAs.

Quality Server

A PeopleTools-based analysis and graphing program.

Quality Step Increase (QSI)

A step increase awarded to an employee for sustained high quality performance.

Quantity Allocation Method

In PeopleSoft Inventory, the method used to determine how available quantity will be allocated to prioritized demand lines when using the online reservations page.

Quantity Precision Rules

A set of rules specifying whether item quantities for a given unit of measure are expressed as whole numbers or as decimals. Quantity precision is set at the inventory **SetID** and item-UOM levels.

Query

A set of data members that are selected from a Class catalog (provided by the Business Interlink Plug-in) as well as a generic form of Criteria. The criteria are composed of <left-hand-side> <Relational Operator> <right-hand-side> statements that can be concatenated using a set of logical operators. All operators and class catalogs are dynamically provided through the Business Interlink Plug-in.

R

Race And National Origin Code

A code that identifies the employee's basic racial and national origin category.

Range of Dates Reporting

A Time and Labor process that enables you to report a start and stop date, a time reporting code and task information for a single employee. The system transforms the information into instances of daily time based on the employee's schedule or default work schedule, replacing the scheduled time with the entered Time Reporting code and the number of scheduled hours on a day-to-day basis.

Range Penetration

In PeopleSoft Workforce Analytics, Range Penetration is the degree to which an employee's actual pay has progressed through their salary grade, and is expressed as a percentage. The calculation is:

Range penetration = (Employee Base Pay – Range Minimum)/(Range Maximum – Range Minimum).

Range Width

In PeopleSoft Workforce Analytics, the difference between the maximum and the minimum values of the pay range calculated using the following formula (and expressed as a percentage):

(Maximum – Minimum)/Minimum.

Rapid Time Entry

The process that enables you to enter daily time for single employees without the system editing your field entries. The system populates temporary tables, which are used by a batch process that reads, edits and moves the data into the appropriate time and labor tables. You cannot prompt for valid values in any of the fields, and the online system does not edit any of the data you enter against other tables.

Rate Code

Alphanumeric reference to the cost per hour or unit of time reported to a specific TRC.

Rate Code [Global Payroll]

IDs for pay components. Rate codes define rates of pay and are set up in the Comp Rate Code table. Rate codes are then used to represent pay components in pages and when you configure compensation packages in the compensation record.

Rate Code Group

A rate code group is a number of pay components (represented by rate codes) bundled into a subset of a compensation package. The rate code group is used to calculate percentage-based components that do not apply the percentage to all pay components in the compensation package. Rate code groups are constructed on the Rate Code Groups page.

Rate Code Type

Defines how the monetary value of the rate code is calculated. The compensation rate code type is defined on the Comp Rate Code table. Valid values are Flat Amount, Hourly Rate + Flat Amount, Hourly Rate, Percent, and Points.

Rate Combinations

The combination of rate types and conversion rates with account types that is linked to your budgeting model. Typical rate types are current, commercial, floating, average, and historical. Effective dates define different rates for different budget periods. There are several conversion rates for any pair of currencies including not only the current rate, but others rates such as average, historical, asking, and so on. These different types of rates are appropriate for different types of accounts.

Rates

The arrays of values used to calculate the cost of a plan to an employee. Rates can be age-graded, service-related, or general, depending upon the benefit plan type. Any number of benefit program and benefit plan combinations can use each set of rates.

In Enterprise Performance Management, a rate is determined by the user and specifies the dollar amounts to be calculated for each model. This is a financial services industry term.

Rating Model

The scale used by your company to measure competency proficiency. The default rating model is the PSCM (PeopleSoft Competency Management) Rating Model that PeopleSoft delivers with your PeopleSoft Human Resources System.

Raw Punches

See Actual Punch; typically this is distinguished from a rounded punch (see Time Reporting)

Reason Code

Reason Codes provide explanations for occurrences such as returned stock and changes to order headers, lines, or schedules.

Reason Code

A code describing employee time such as comments for sick time or travel time.

Reasonableness

In PeopleSoft Demand Planning, a technique that checks the trend and projected annual growth to make sure that a forecast is realistic. If a forecast falls outside either boundary, the system automatically adjusts it and sends a warning message.

Reassignment

Change of an employee from one position to another without promotion or change to lower grade.

Recalculate Forecast

In PeopleSoft Demand Planning, a forecasting feature that uses the existing model and its associated parameters to create a new forecast.

Recalculate VAT at Payment

Allows the VAT amount to be adjusted at the time of payment if an early payment discount is taken. This calculation option is only valid when VAT is calculated at Gross.

Receipt Cost Method

Determines how you cost receipts. Receipt cost methods include Actual, Non-Cost, and Standard.

Receipt Line

A line associated with a Receipt ID that identifies an item and quantity. If the respective tracking is activated, the lot, lot suffix, and serial number are also identified.

Receivable Update

See **Posting**

Receivables Item

An individual receivable. An item can be an invoice, credit memo, or debit memo. Items and payments combined comprise a customer's balance.

Reconciliation

Within PeopleSoft Enterprise Performance Management, reconciliation differs slightly when it is performed within the PeopleSoft Enterprise Warehouse and when it is performed within the PeopleSoft Analytic Applications.

In PeopleSoft Funds Transfer Pricing (FTP) and PeopleSoft Risk Weighted Capital (RWC), reconciliation identifies differences between Performance Ledger balances and the instrument or position balances, which are risk weighted according to the basis rules you have assigned. The first step in reconciling basis rule balances is to reconcile the individual balances for accounts, instruments, and positions. Reconciling the total balances is the second step. This means that you reconcile the difference between Account/Instrument balances, and the difference between Account/Position balances.

In the PeopleSoft Enterprise Warehouse, reconciliation is a period-end process that posts journal entries to the Performance Ledger for the discrepancies found when you reconciled the individual balances. Typically, you'll run the PF Reconciliation engine after a period to compare "to and from amounts" between tables such as REVENUE_F00 and PF_LEDGER, or the GL_LEDGER and the PF-LEDGER.

Record Date

The date a stockholder must officially own shares in order to vote at the meeting or to derive an adjustment resulting from a stock split or a stock dividend. The Board of Directors sets the Record Date.

Record Definition

A logical grouping of data elements.

Record field

Once a field is inserted into a record definition it becomes known as a Record Field within the record.

Record Group

A set of logically and functionally related control tables and views. Record groups exist for two basic reasons:

- To save you time—with Record Groups, TableSet sharing can be accomplished quickly and easily, eliminating an enormous amount of redundant data entry
- To act as a safety net—Record Groups ensure that that TableSet sharing is applied consistently across all related tables and views in your system.

Record Input VAT 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 being tracked on a transaction, this flag is always set to Yes. This flag is not used in Order Management, Billing, or Receivables, where it is assumed that you are always recording only output VAT, or in Employee Expenses, where it is assumed that you are always recording only input VAT.

Record Output VAT Flag

For certain transactions within PeopleSoft Purchasing, Payables, and General Ledger, it may be necessary to record both input VAT and output VAT on the same transaction. Generally, this would be a situation where the purchaser was required to self-assess VAT. Within these situations, this flag indicates that you are recording output VAT on the transaction. This flag, in conjunction with the Record Input 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. This flag is not used in Order Management, Billing, or Receivables, where it is assumed that you are always recording only output VAT, or in Employee Expenses, where it is assumed that you are always recording only input VAT.

Record Owner

The "Stockholder of Record" of the stock. This may be different from the "Beneficial Owner" of the stock.

Record Suites

Record suites are temporary tables that enable the system to track how many processes are running. These temporary tables leave the fact tables accessible for processing other jobs simultaneously without impacting your processing.

Reduction In Force (RIF)

Method used to reduce the number of government workers in an agency.

Reemployed Annuitant

An employee who has retired from Federal employment and is receiving an annuity. His/her salary is reduced by the amount of the annuity.

Reference Designators

A user-defined alphanumeric identifier that determines where a component is placed in an assembly.

Reference Transaction

In People Soft 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.

Referential Integrity

Issues that occur when an update to an instance of one object invalidates one or more instances in a related object. In other words, when you make a change to one area of the application, referential integrity makes sure the changes do not adversely affect another area of the application.

Refresh Time

The process that retrieves the appropriate current version of objects related to employee time (such as task profiles or work schedules) and associates them with that time.

Region Codes

Regions may or may not be physical entities, they may simply be another way to geographically categorize an area. When a region does represent a physical entity, the region code has the same characteristics as a business, that is, an address and a language spoken.

Register of Separations and Transfers (ROST)

The ROST is a regulatory compliance document used by federal agencies to summarize the information in an employee's Individual Retirement Record (IRR). The ROST is a one-page cover sheet that accompanies a batch of IRRs being submitted to the Office of Personnel Management (OPM) at the time of an employee's separation from a federal agency. Employees covered by the CSRS retirement plan require SF-2807. Employees covered by the FERS retirement plan require SF-3103.

Registration

The name or names that appear on the stock certificate to indicate who owns the stock.

Registration Statement

The document that must be filed to permit registration of an issue of securities under the Securities Act of 1933. A major component of the registration statement is the prospectus that is to be supplied to prospective purchasers of securities.

Regression Analysis

A statistical technique that determines the relationship between two or more variables. Regression predicts the value of one variable (the dependent variable) based on one or more independent variables.

Regular Base Compensation

In PeopleSoft Workforce Analytics, the annualized, quoted, compensation rate for a job. Consists of fixed compensation, does not include variable compensation.

Regular Time

An employee's normal (scheduled/shift) work hours.

Regular Time

In PeopleSoft Workforce Analytics, an employee's normal (scheduled/shift) work hours.

Regulation T

Federal Reserve Board regulations governing the extension of credit by brokers or dealers, including their participation in same-day sale transactions and sell to cover exercise.

Regulatory Region

The Regulatory Region functionality in PeopleSoft HRMS is designed for use in performing regulatory and regional edits. You'll use Regulatory Region to drive PeopleCode edits, perform set processing, and control what codes and values the operator sees. A Regulatory Region can be any country (or province or state) where there are specific laws and regulations addressed by functionality in PeopleSoft HRMS.

In Enterprise Performance Management, a Regulatory Region is a region with a common regulatory framework; such as a country (CAN for Canada), or a smaller state or provincial entity (CANBC for British Columbia).

Related Education

Education above the high school level that has equipped the applicant with the KSAs to perform successfully the duties of the position being filled. Education may relate to the duties of a specific position or to the occupation, but must be appropriate for the position being filled.

Release

An industry standard term associated with the lifting of a company's Repurchase Option from a portion or all shares from a Restricted Stock Award (RSA). RSA's are subject to release schedules, similar to vesting schedules.

Relevant Constraint

A constraint PeopleSoft Planning considers when it calculates a score and when it optimizes the schedule. See also **Scorecard** and **Optimize**.

Reloads

Some stock option plans provide for the grant of a "reload" stock option in connection with stock option exercises, typically by means of stock swaps. A reload option feature provides that upon a stock exercise, the employee will receive an automatic grant of a new stock option at the then-current fair market for the shares that they exercised or for the shares that they used to swap.

Remark Codes

Codes that cause the printing of pre-set text passages on a notice of action form. Some passages are general purpose and others are specific to the personnel action being processed.

Remit From Customer

A customer who is responsible for payments billed to other customers. During cash application, it's useful to look at open items for the Remit From group.

Remittance Worksheet

A work space in PeopleSoft Receivables used to select drafts for remittance to the bank.

Reorder Point

The identifier that automatically locates a replenishment need for an inventory item. When the physical quantity in a location falls below the reorder point, a replenishment request can be created.

Reorder Point Policy

In PeopleSoft Inventory Planning, a policy that determines when a replenishment order is launched for an item. The policy has several methods that include days supply, lead time demand, and **Fixed Quantity**.

Replacement Option

The "new" "replacement" stock option that will replace the original stock option. This option will have a grant price lower than the original stock option.

Replenish

A process that indicates when items need to be resupplied from external sources. In PeopleSoft Inventory, the process can occur on an ad hoc basis or at predefined reorder points.

Replenishment Request

In PeopleSoft Flow Production, an online request for material made when the material is needed. You can generate replenishment requests manually or automatically using backflushing. You can communicate that request using pull lists, pull tickets, or Workflow notifications.

Report Scope

A feature that creates multiple instances of an nVision report using a single report request. Each instance contains data specific to an individual ChartField, such as a business unit or department, or for a group of ChartFields, such as all sales departments. Using Scope, each report instance can share the same layout while containing data unique to these particular ChartFields.

Reported Time

Clock time or elapsed time provided to the system by the user (see Time Reporting)

Reporting Person

An insider that is regularly considered by the SEC to have material information and policy-making authority for the corporation. These individuals are subject to the reporting requirements promulgated by Section 16 of the Securities Exchange Act of 1934. Reporting Persons typically include Directors, Officers, and shareholders with 10% holding interest in the equity of the registrant's securities.

Repricing

An agreement between the corporation and the optionee that allows the optionee to cancel an outstanding high-priced, usually "Out-of-the-Money" stock options for lower-priced options.

Repricing Election

Eligible optionees can choose (elect) to accept the corporation's repricing offer or choose to decline the offer.

Repurchase

The reacquisition of shares of stock from an individual by a corporation. This usually occurs when an individual fails to meet the vesting requirements on a RSA or option that is exercised before it vested. The corporation might pay the original cost of the shares to the individual or the fair market value of the shares at the time of repurchase.

Repurchase Option

An irrevocable, exclusive option to repurchase up to the number of shares that constitute Unreleased Shares at the original purchase price per share. The Company shall exercise said option. The repurchase of outstanding shares is regulated under the laws of all states (except Massachusetts). Under some laws, as under the Model Business Corporation Act, the repurchase is prohibited unless the corporation remains solvent, in both the equitable and bankruptcy senses of insolvency and after taking any liquidation preferences of other outstanding stock into account.

Repurchase Right

A company's contractual right to buy back from an employee any stock resulting from the exercise of the option. The buy back can be at fair market value, book value, or the original purchase price.

Reservation Method

The method used to reserve soft reservation items — either batch COBOL reservations or on-line allocation and reservation.

Reserved

A flag indicating that the inventory item is reserved for stock fulfillment in the inventory business unit.

Reserved Orders

Orders that have been reserved against on-hand available quantity at the business unit-item level. Reserved orders are found in the DEMAND_INV table.

Reset

In PeopleSoft Demand Planning, a function of the **Forecast Calculation Process** that determines which forecast model will produce the best forecast, meaning the model with the lowest ratio of error.

Resolution

An activity that closes or partially closes a deduction, such as matching it to a deduction authorization, writing it off, or sending it back to PeopleSoft Receivables.

Resolution Entry Type

Code that identifies how to process activities for items in PeopleSoft Deduction Management and how to create accounting entries.

Resolution Method

A set of rules that defines how to automatically match or write-off deductions in PeopleSoft Deduction Management.

Resolution Worksheet

The workspace in which deduction items are paired with offset items and resolved or written off in PeopleSoft Deduction Management.

Resource

In PeopleSoft Manufacturing, any crews, machines, and tools that can optionally be used at work centers to complete tasks. In PeopleSoft Performance Measurement, any homogeneous grouping of general ledger line items.

Resources

Resources are the economic elements that are required to perform the activities associated with your business. Resources are consumed in the performance of these activities, and thus denote operating costs. In PeopleSoft Activity-Based Management, resources are typically regarded as the groupings of one or more general ledger accounts. In a service business, resources might include salaries, office rentals, and costs of capital such as information systems, depreciation, real estate taxes, and other associated costs.

Resource Amount

The monetary amount of a single, specific resource transaction. The Resource Amount maps to the Posted Total Amount when posted to the general ledger.

Resource Category

A field for defining individual resource types more specifically. For example, if you have a resource type of labor but want to break it down further for tracking purposes, you might define resource categories of architect hours, carpenter hours, plumber hours, and electrician hours. Resource categories are optional.

Resource Driver

In Activity-Based Management, a Resource Driver defines the quantity of resources used by an activity.

Resource Group

A category of resource types. You can define relationships between the resource types within a resource group to facilitate analysis of project costs. For example, if you had resource types for standard labor and overtime labor, you could group them together in a resource group to calculate total labor.

Resource Planning

In PeopleSoft Activity-Based Management, Resource Planning focuses on resources allocations that create expected results like driver rates and cost object costs.

Resource Quantity

A field on each resource transaction that identifies nonmonetary quantity. For example, on a resource line for 12 ball valves the quantity would be 12.

Resource Source

A field on each resource transaction that identifies the system in which the cost originated. For example, PeopleSoft Payables would be the resource source for a resource transaction created from a voucher in that system.

Resource Subcategory

A field for defining individual resource types and categories more specifically. For example, if you have a resource type of labor, and resource categories of architect hours, carpenter hours, and plumber hours, you might want resource subcategories of regular hours and overtime hours. Resource subcategories are optional.

Resource Supplied

An attribute that enables you define a resource as committed or flexible. A committed resource is one that will not likely change in the short term. A flexible resource is more likely to change within the short term.

Resource Transaction

An individual cost line within PeopleSoft Projects. It is through resource transactions that individual costs and types of costs are tracked. Each resource transaction contains a cost and a quantity and as many identifiers of that cost as necessary. Resource transactions are created when you receive information from other systems, run allocations with project resources as the target, or perform internal transactions such as billing, project closure, or adjustments.

Resource Type

The resource transaction field in PeopleSoft Projects that identifies the resource associated with a given cost. Resource types may be very general or very specific depending on your needs; they are used in conjunction with resource categories, resource subcategories, and resource groups.

Resource Use

Resource Use defines the behavior of a resource within PeopleSoft Enterprise Performance Management. An intermediate resource is a grouping of general ledger line items that may be allocated to another intermediate resource or to a primary resource.

Restricted Punch

A punch which is not accepted because it occurs outside of the predefined number of hours and minutes before or after a scheduled (Understanding Time Collecting Device)

Restricted Securities

Securities issued privately by the company, without the benefit of a registration statement. Restricted securities are subject to a holding period before they can be sold under Rule 144.

Restricted Stock Awards (RSA)

An award of shares of stock to an individual, typically granted at the par value or for no consideration. The shares are awarded on the basis of some future performance goal, either the passage of time (vesting) or the attainment of a specific goal. When the goal is achieved, the vesting occurs. The individual, typically, has all other shareholder rights over these shares such as, voting and dividend rights. The shares are issued in the name of the individual at the time of the award and are held in escrow until vesting occurs. If an employee terminates prior to the vesting of the shares then the company normally repurchases the unvested shares.

Retained Grade Effective Date

Date employee became eligible or began receiving a retained grade and pay.

Retained Grade Expiration Date

Expiration date of an employee's retained grade and pay.

Retest Date

In PeopleSoft Inventory, the date a lot should be inspected to determine whether it is still acceptable for fulfillment or consumption. (Retest Date = Creation Date + Retest Lead Time)

Retirement

Types of retirement are:

- Mandatory Retirement.
- Disability Retirement.
- Voluntary Retirement.
- Special Option Retirement.
- ILIA (In Lieu of Involuntary Action) Retirement.

Retirement Coverage Code

A code used to denote an employee's retirement coverage. The major ones include the following:

- Civil Service (CSRS)
- Federal Employees Retirement System (FERS) and FICA
- Foreign Service (FS)

- CSRS Offset
- CSRS - Special (for LEOs)
- FERS and FICA - Special (for LEOs)
- Social Security System
- None

Retroactive Benefits/Deductions

Deductions taken or benefits granted due to a recalculation of previous benefits and deductions. Late or modified union contracts, late paperwork, and delays in benefit enrollment processing may all result in a need for benefit/deduction recalculation.

Return Type Code

A designator on returned material authorizations (RMAs) that indicates what actions the return initiates. This may include replacement of the product or the creation of a credit memo in PeopleSoft Billing.

Reverse Split

A reduction in the number of outstanding shares of a corporation's stock, with a corresponding increase in the stock's value.

Reversionary Annuity

A form of pension payment where the retiree foregoes all benefit during his or her lifetime so that the entire benefit is paid as an annuity to a beneficiary after the retiree's death. If the beneficiary predeceases the retiree, the benefit is forfeited.

RIDDOR (Reporting of Injuries, Diseases, and Dangerous Occurrences Regulations)

Health and safety regulations in the United Kingdom requiring employers to report certain types of health and safety incidents to the Health and Safety Executive (HSE).

Rider

A special court-ordered or regulatory provision that may be applied to an enrollment to expand or limit any dependent or beneficiary coverage.

Risk Function

In Risk Weighted Capital, this is a user-defined formula that the system uses to derive risk weights.

Risk RuleSet

Used to assign a number of rules to a basis, for processing by the PeopleSoft Risk Weighted Capital Application. Used to group together a number of rules that frequently apply to the same type of balance.

Risk Type

In Risk Weighted Capital, this defines the types of risk associated with your business or activity. For example catastrophic, credit, legal, operational, regulatory, foreign exchange, market, interest rate.

Risk Weight

In Risk Weighted Capital, the risk weight is assigned by risk type, and is used to calculate capital allocation or normalized loss for the account or activity.

Risk Weighted Capital (RWC)

See PeopleSoft Risk Weighted Capital

RIZIV Code

This code is for Belgian employers to track the Federal Institute for Illness and Disability Insurance category.

ROE (Record of Employment) Reason Codes

ROE codes are defined by the Canadian government for employers to record employment actions such as Return to School or Pregnancy Leave.

ROLAP (Relational Online Analytical Processing)

ROLAP refers to the analytical processing and analysis of a relational Data Mart cube. ROLAP, is a form of OLAP that leverages the power and flexibility of relational databases.

Role

A role consists of a designated set of tasks, competencies and accomplishments required for a job code or a position.

Role user

A PeopleSoft Workflow user. A person's Role User ID serves much the same purpose as their Operator ID does in other parts of the system. It allows the system to uniquely identify the user and to determine what data the user has access to.

PeopleSoft Workflow uses Role User IDs rather than Operator IDs because it needs different kinds of user information than the rest of the system does. Specifically, it needs to know how to route work items to the user---an email address, for example---and what role the user plays in the workflow. Plus, you can include role users in your workflow who aren't PeopleSoft application users and who don't have Operator IDs.

Roles

Roles describe how people fit into the 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 Forward

In commitment control, rolling budget balances forward from the budget ledger you are closing (the source budget ledger) into the new (target) budget ledger.

In PeopleSoft Enterprise Planning and Production Planning, a utility that moves tasks from the past to a valid point in the future using time periods rather than fixed dates.

Roll Up

The act of totaling sums based on information tree hierarchies. You can roll up data for any group of details that you have defined as dependent with the Tree Manager.

Rolling Plan

An ESPP offering period where the purchase date is measured from the offer start date. If at the purchase date, the current stock price is lower than the last stock purchase price, you may elect to reset your employees to the new lower purchase price. The offering period is now based of the new purchase date.

Rollup

In PeopleSoft Demand Planning, the process of adding up the demand and forecast **Array** from one level to the next from child to parent. Information such as caption, description, and unit of measure can also be rolled up. The process is also referred to as summarization.

Rounded Punch

A punch that has a company's rounding requirements applied to it (see Time Administration)

Rounding Rule

Defines a rounding rule. You use rounding rules in numerous places—for instance you could round an earning, deduction, or many of the elements that make up an earning or deduction.

Routing (Manufacturing and Engineering)

A set of information detailing the method to manufacture a particular item. It consists of sequentially numbered operations that reference the task to be performed, the work center in which the task is to be performed, the resources to be used, and the time required to complete the task. Engineering Routings differ from Manufacturing Routings in that they are not visible within Production Planning or Production Management and are isolated from Manufacturing.

Routing Option

In PeopleSoft Planning, a valid method for replenishing supply for an item. There are two types of routing options: build options and purchase options. An item may have more than one routing option.

Routing Transit Number (RTN)

A number that identifies the financial institution to which an electronic payment should be sent for deposit.

Routings

Routings connect the activities in the workflow. They are the system's means of moving information from one place to another, from one step to the next. Routings specify where the information goes and what form it takes—email message, electronic form, or worklist entry.

RSZ (Rijksdienst Sociale Zekerheid) Category Codes

These government defined Social Insurance category codes are used to maintain social security records for your Belgian employees. RSZ Categories are associated with a Contract Type, Statute and Substitute for Claey's Formula calculations.

Rule

Representation of a company's compensation, task allocation, or exception requirements (see Creating Rule)

Rule 10b-5

A SEC rule that prohibits trading by insiders on material non-public information. This is also the rule under which a company may be sued for false or misleading disclosure.

Rule 144

A SEC rule that applies to public re-sales of restricted securities as well as all sales by affiliates. The requirements include (1) current public information about the issuer, (2) a one-year holding period for "Restricted Securities," (3) unsolicited brokers' transactions, (4) an amount limitation. the greater of 1% of the outstanding stock or the average weekly trading volume may be sold during any three-month period, and (5) a Form 144 filing.

Rule Actions

Functions that can be used in the creation and application of a rule (see Time Administration)

Rule Elements

Customer defined pieces of information which are passed to Time Administration in order to apply and evaluate rules (see Time Administration)

Rule Period

A Time & Labor period used in the evaluation and application of a rule (see Time Administration).

Rule Program

Specifies the set of rules the Time Administration process will execute and the order in which it will execute the rules.

Rule Results

Net effect of the application of a rule; for instance, the creation of time, initiation of workflow, modification of reported tasks (see Time Administration)

Rule Templates

Templates used to quickly create a variety of rules for the Time Administration program to execute when processing reported and/or scheduled time. Some examples are. compensation rules for overtime and holidays, notification rules for irregular attendance, and rules for just about any other time-reporting situation that requires special processing.

Rules

Rules are your company's business practices captured in software. Rules determine what activities are required to process your business data.

Rule Set

Rule Sets enable you to apply basis rules to your PeopleSoft Analytic Application in the sequence that you prefer. This is particularly helpful if there are multiple basis rules for the same account node, product node, or position source code. Rule Sets can control the execution sequence of your rule combinations, filter combinations, or both. The first occurrence on the node will be applied and any other occurrence will be ignored. Rule Sets are also used with the Data Manager, and with the Currency Conversion engine.

Rules/Time Administration

A physical implementation or execution of a company's compensation, exception and task rules (see Time Administration)

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 some type of program that manipulates data in some way.

Run Control ID

A unique ID to associate each operator with their own run control table entries. Process Scheduler.

Run ID

Code that uniquely identifies a Run Control for batch processes.

RWC (Risk Weighted Capital)

See PeopleSoft Risk Weighted Capital

S**Safety Stock Policy**

In PeopleSoft Inventory Planning, a policy that determines how safety stock quantities are calculated for an item. The policy has several methods that include days supply and percentage of demand fill.

Salary

Rate of compensation received by an employee.

Salary Group

Part of a group of defaults assigned to job codes. A salary group may include items such as steps and grades dependent on individual company parameters.

Salary Plan

A plan of salary defaults, grades, and step components

Salary Step Components

Pay components assigned to a salary step by entering the corresponding rate codes on the Salary Step Components page.

Salvage Value

An estimate of the amount of money one might receive upon selling an asset once that asset reaches its useful life. Salvage value is used in several depreciation calculations, including Straight Line.

Sales Order Rebate and Penalties

Rebates or penalties that are calculated against sales orders independent of **Buying Agreement**.

Sales Person

A required field used in PeopleSoft Receivables, Billing, Order Management, and Deduction Management when working with items. Each item must be assigned to a credit analyst. If no

credit analyst is assigned to an item, the credit analyst assigned to the customer is used as the default.

Sales UOM

The only units of measure that can be referenced on sales orders and quotes. You establish them on the Product Attributes by UOM page.

Same-Day Sale

An exercise and sale occurring on the same day. The exercise of the option and sale of the underlying shares take place simultaneously. The broker uses the proceeds of the sale to pay the company the exercise price and any tax withholding and the optionee the net cash (less any brokerage commission/fees).

Sample Method

One method of entering characteristic readings for a quality control plan. Using this method, for one given control plan you inspect all the characteristics for the first sample, then all the characteristics for the next sample and so on.

Scale

On a Goals Matrix (In PeopleSoft Workforce Analytics), a scale that defines the lowest, middle, and highest levels of performance needed to achieve associated minimum, midpoint, and maximum levels of compensation pay out. These are referred to as the Threshold, Target, and Stretch levels, respectively. The scale can be used to standardize multiple performance goals to a common scale of measurement.

Scenario

A scenario is a particular outcome you are analyzing when you run in Scenario Manager. Scenarios enable you to study various changes in organization models you created. For each parent and child model you want to study, you create a scenario ID that you use with all run controls.

There are two types of scenarios defined in the Scenario Manager: Historical and Forecast. In the case of an Historical Scenario all future looking fields will be display only and the Scenario Manager component serves as a wrapper to run any analytic engines. In the case of a Forecast Scenario, the Scenario Manager refers to all the business rules, forecasts, and economic assumptions that make up the scenario.

In PeopleSoft Budgeting a scenario is a ChartField used in PeopleSoft Budgeting to identify different budget iterations that use different assumptions.

Schedule

Specific task, date, and time to be worked by a Time Reporter (see Scheduling)

Schedule 13D or 13G

Disclosure forms required to be filed with the SEC and the company by a shareholder (or shareholders) that own(s) more than 5% of a public company. Schedule 13G is a short-form version of the 13D and may generally (but not always) be used only by institutional investors.

Schedule Group

A category of employees or employee groups associated for purposes of time scheduling.

Schedule Line Number

The line associated with an Order ID. The schedule line identifies an item and scheduled ship quantity that may be different from the requested quantity due to item availability.

Schedule Number

A number identifying the salary table form that an employee's pay is computed. Also has a second meaning related to the Payment Voucher processing for the ECS.

Schedule Reconciliation

In PeopleSoft Payables, the process of reconciling scheduled payments by Payment Schedule ID. Schedule Reconciliation helps U.S. federal agencies meet their requirement to schedule or group together payment orders for submission to the Treasury Disbursing Office.

Schedule Template

An ordered pattern of workday(s) and/or off day(s) used in scheduling (see Scheduling)

Schedule Type

In PeopleSoft Payables, an indicator of the nature of items purchased with a Payment Schedule.

Scheduled Punch.

A time reporter's expected punch (see Scheduling)

Scheduling

A function of PeopleSoft Time and Labor and PeopleSoft Global Payroll that enables you to create work schedules and assign them to employees.

Scorecard

A weighted sum of constraint violations in a schedule that evaluates the schedule's validity (that is, acceptability). The score is calculated by adding the value for each relevant violated constraint. See also **Penalty**, **Weight** and **Relevant Constraint**.

In PeopleSoft Balanced Scorecard, views of a strategy tree's components and Key Performance Indicators with red, yellow, or green scores that show its assessments.

Scrap

Any material outside of specifications and possessing characteristics that make rework impractical.

SearchIndex

A set of objects that give the programmer the ability to create, delete, insert, and update a search index and the items within it. Search index items contain a set of statistics about the document that has been indexed (keywords, number of occurrences, proximity to other words, and so on) as well as a key that can be used to point to the document (a URL, database key, or file path).

SearchQuery

A set of objects that allow the programmer to pass a query string and operators to the search engine and receive a set of matching results with keys to the source documents from the search index in return.

Seasonal Index

In PeopleSoft Demand Planning, measures the amount by which a forecasting period is influenced by seasonal effects. The index typically measures the percentage of difference between the base demand in the period and the expected actual demand in the period. An index of 100 indicates an average period in a seasonal cycle.

Seasonal Profile

In PeopleSoft Demand Planning, identifies the weight index assigned to a forecast time period to take in account seasonal fluctuations in the demand.

Seasonality Group

In PeopleSoft Demand Planning, a group of **Forecast Items** with a repetitive pattern of demand from year to year where some periods are higher than others. Typically a group of items is designated as a contributor to the seasonality group. Contributors are chosen because they are representative of the group, are stable, and have at least two years of demand history. The seasonality group profile is more stable than individual profiles of the contributors because the Aggregation process smoothes out random errors.

In Enterprise Planning and Simulation, a Seasonality group is a group of items with similar seasonal patterns. To determine if a forecast element is seasonal or nonseasonal, by averaging their history values over a year and determining where they were above and below average. Seasonality groups capture means seasonal behavior among related products.

Secondary COBRA Events

COBRA qualifying events that extend the amount of time a participant is eligible for COBRA coverage. For an event to qualify as a secondary COBRA event, it must fulfill the following qualifications: The participant must already be enrolled in COBRA coverage as a result of an initial COBRA event, the initial COBRA event must be one that is associated with a change to the employee's job status (such as a reduction in hours, termination, or retirement), and the

secondary event must be one of the COBRA event classifications that involves loss of coverage for the dependent (such as divorce, marriage of dependent, or death of employee). See COBRA and Initial COBRA Events.

Secondary event code

Secondary event codes, also called transaction codes, specify the type of transaction in detail. For example, a transaction's secondary event code could say that the transaction is a catalog order, a rush order, or a request for a sample. Not all transaction types include secondary event codes.

Section

A set of logically related elements that are to be resolved during the payroll process. You define your payroll process by creating sections and adding them to process lists. You can create four different types of sections: standard, generate positive input, sub-process, and payee.

Section 16(a)

Provision of the Securities Exchange Act of 1934 that requires company insiders to file changes in beneficial ownership of the company's equity securities and periodic reports disclosing their holdings.

Section 16(b)

Provision of the Securities Exchange Act of 1934 that requires that any profit realized by a company insider from the purchase and sale, or sale and purchase, of the company's equity securities within a period of less than six months must be returned to the company. It is also known as the "short-swing profit" rule.

Section 423

The Internal Revenue Code section that regulates Employee Stock Purchase Plans.

Section 83(b) Election

A tax filing within 30 days of grant that allows employees granted restricted stock to pay taxes on the exercise date, rather than the date when restrictions lapse. If an employee files the election, taxes are based on the fair market value on the exercise date, with any future appreciation taxed as a capital gain. If the employee does not file an election, taxes are based on the fair market value on the date the restrictions lapse, which will be higher assuming the stock has appreciated in value.

Securities Act of 1933

Often referred to as the "truth in securities" law, the act requires that investors receive financial and other significant information concerning securities being offered for public sale; and prohibits deceit, misrepresentations, and other fraud in the sale of securities.

Securities Exchange Act of 1934

The Congressional act that created the Securities and Exchange Commission. The Act empowers the SEC with broad authority over all aspects of the securities industry. This includes the power to register, regulate, and oversee brokerage firms, transfer agents, and clearing agencies as well as the nation's securities self regulatory organizations (SROs). The various stock exchanges, such as the New York Stock Exchange, and American Stock Exchange are SROs. The National Association of Securities Dealers, which operates the NASDAQ system, is also an SRO. The Act also identifies and prohibits certain types of conduct in the markets and provides the Commission with disciplinary powers over regulated entities and persons associated with them. The Act also empowers the SEC to require periodic reporting of information by companies with publicly traded securities.

Security Clearance

Security Clearances (Classified, Secret, Top Secret) are granted to employees by government agencies and are usually associated with jobs that bring employees into contact with classified government projects or sensitive technologies.

Security Event

In commitment control, events that trigger security authorization checking, such as budget entries, transfers, and adjustments; exception overrides and notifications; and inquiries.

Segmentation

You can “segment” components of pay based on such events as changes in compensation, employee status, or job changes during a pay period. For example, if an individual changes jobs in the middle of a pay period and your organization has a practice of separating components earned in the first job from those earned in the second job, you can set up your system to trigger segmentation of earnings results on the pay slip when there is a change to the job change action/reason field.

Selective Factors

Knowledge, skills, abilities or special qualifications that are in addition to the minimum requirements in a qualification standard, but are determined to be essential to perform the duties and responsibilities of a particular position. Applicants who do not meet a selective factor are ineligible for further consideration.

Selective Merge

In PeopleSoft Enterprise Warehouse the selective merge allows for an additional WHERE clause when you delete a merge.

Self-Service Application

Another name for PeopleSoft's HRMS and ERP applications accessed using a browser.

Sell to Cover Exercise

When an optionee sells a portion of the option shares to cover the exercise cost and any applicable taxes.

Seniority Pay

A premium paid for seniority or for the length of time an employee works for an organization.

Seniority Pay Components

Pay components whose rate codes are assigned to the seniority rate code class SENPAY (provided by PeopleSoft) on the Comp Rate Code page, allowing you to access the seniority pay functionality.

Seniority Rate Codes

A premium paid for seniority or for the length of time an employee works for an organization.

Separate Debit and Credit

A feature in PeopleSoft General Ledger that captures and reports in greater detail the accounting information that resides in balance sheet accounts. It shows the gross debit/credit balances in addition to the net balance for each account stored in the ledger. This feature also supports reversing—debit and reversing—credit journal entries for error correction.

Server Process

A server process is executable code that receives incoming client requests on the application server. The server process carries out a client request by making calls to a service that executes SQL against the database.

Service

A service performs a particular task of an application. Examples of services are MgrGetObj, SQLAccess, RemoteCall, and so on. When a client workstation sends a request to the application server, it sends a service name and a set of parameters, such as "MgrGetObject + parameters". The application server associates the service request with the appropriate server process to complete the transaction.

Service

The PeopleSoft Pension Administration function that determines how much service credit an employee has accrued.

Service Buy Back

The process by which an employee repays a pension plan in order to restore service credit that was forfeited when the employee withdrawal previous contributions. Typically, employees withdraw contributions upon termination and initiate service buy back processing upon rehired.

Service Purchase

The process by which an employee gets additional pension service credit for periods not normally considered eligible. The employee “purchases” this service by paying into the plan.

Service Schedule

A table showing how much service an employee earns based on the number of hours the employee worked during the year or month.

SetID

The label that identifies a TableSet.

Sex Code

Used to indicate gender.

Shape

For a transaction, the set of inputs and outputs for that transaction. For a class, the data members of that class.

Share

A share of a company's stock. Stock options give you the option to purchase a certain number of shares of company stock.

Share Price

The price per share of a company's stock. See, also, "stock price."

Shareholder

Owner of one or more shares of stock in a corporation. Also known as a stockholder or investor.

Shares Available to Issue

The total number of shares authorized, less shares granted, plus cancellations that revert to the Plan pool.

Shares Cancelled

This is usually triggered by a specific event, such as termination of employment in which the unvested shares as of the date of termination are no longer available for future vesting and exercise. These shares are therefore canceled from the option and can be returned to the plan, retired to treasury or allocated back to a group.

Shares Exercisable

The number of shares that are vested and available for exercise.

Shares Exercised

The number of shares purchased upon exercise of a stock option.

Shares Expired

Option shares that no longer are exercisable at the end of the option term. The length of the option term is defined in option agreement. This date is usually the earlier of the exercise period for vested shares after termination of employment or the full length of the option term.

Shares Outstanding

The number of company shares currently held by shareholders, as tracked by the transfer agent

Shift [Time and Labor]

The block of hours that an employee works in a day, such as nine to five, four to eleven, or ten to six. In PeopleSoft Time and Labor, Shift is used as a template of clock hours for scheduling an employee or group of employees to be at work or available to work (on call). Shifts may be constant, rotating, repeating, and/or split; any given shift may or may not have an associated Shift Differential or Bonus. A shift is always associated with a Work Schedule, and consists of clock hour Start and Stop times (two to allow for split shifts), meal periods (two) and relief periods (two).

In PeopleSoft Workforce Analytics, the block of hours that an employee works in a day such as nine to five, four to eleven, or ten to six.

Shift Bonus

A fixed amount (either a flat dollar figure or stated in terms of an employee's rate) paid for working a particular Shift.

Shift Code

A numerical shift identifier that is unique within a SetID.

Shift Differential

Additional compensation paid an employee for time worked during certain shifts. Typically, shift differential is administered as a flat amount per shift, hour worked, and/or as a percentage of the amount paid per shift hour or shift worked.

In PeopleSoft Workforce Analytics, a premium paid for work over regular pay, for which employees on certain shifts may be eligible, such as double-time for late night shifts. Shift differentials are usually stated as an additional rate or factor.

Shift Elements

Individual components of a shift such as TRC start and stop time, duration (see Scheduling)

Shift Name

Customer defined nomenclature for a shift (see Scheduling)

Shift Type

A customer-defined classification associated with a shift. The shift type can be used in the evaluation of rules or exceptions (i.e. On Call) (see Scheduling)

Shift Type [Time and Labor]

Time and Labor defined classification of shifts. Valid shift categories include Flex, General and Elapsed. Shift categories are used in the creation of time reporter schedules (see Scheduling)

Shipping Priority Code

Shipping Priority Codes act as tie breakers during order reservation in PeopleSoft Inventory when different orders are scheduled for the same delivery date and time. When the reservation process in Inventory encounters a situation where there are more orders than available stock, the system reserves the order with the highest priority. If schedules are encountered with the same priority, orders are then considered by order date.

Short Sale

The sale of a security that is not owned or is not delivered at the time of the trade, necessitating its purchase or delivery some time in the future to "cover" the sale. A short sale is usually made with the expectation that the stock value will decline, so that the short seller can eventually cover at a price lower than the original sale, thus realizing a profit. At the time of the short sale, the broker borrows stock to deliver on the settlement date. A short sale can be "naked," in which case the seller does not deliver the shares being sold short and must provide the broker with collateral. Or the short sale can be "against the box," in which case the seller delivers the shares being sold short for the broker to hold "in the box" until the seller chooses to close out the short position.

Short-Swing Transaction

A purchase and sale, or sale and purchase, of the issuer's equity securities by an insider within a period of less than six months. See "Section 16(b)" above.

Short-term Variable Compensation

In PeopleSoft Workforce Analytics, this is cash compensation paid to a worker that is not fixed; includes bonuses and commissions.

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.

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.

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.

Sick Leave

Sick leave is accrued by full-time permanent/seasonal employees at the rate of 4 hours every biweekly pay period; for part-time permanent/seasonal employees, it is accrued at one hour for every 20 hours worked.

Simulated Workforce

In the PeopleSoft Workforce Rewards product, Manage Compensation module, the calculated Simulated Workforce = Existing Employees + New Simulated Employees + Reduced-Employees.

Single Life Annuity

A benefit payable during the lifetime of the participant, with no payments made after the death of the participant. Also referred to as a “life only annuity” or a “straight life annuity.”

Single Signon

This refers to the process by which a user can, after being authenticated by one PeopleSoft application server, access a second PeopleSoft application server without entering a user ID or password.

SIREN Code (Système Informatique pour le Répertoire des Entreprises)

This stands for the Electronic List of Enterprises. The SIREN code is assigned to a company when it registers as a business with the French government, and identifies the purpose of the establishment for regulatory reporting purposes in France.

SIRET (Système Informatique pour le Répertoire des Établissements)

This stands for Electronic List of Entities. In France the SIRET is an identifying number given to a French business by the INSEE, an official statistics and economics organization in France. The SIRET number is a combination of the SIREN and NIC numbers. This number is used by the tax and social security authorities to identify a business enterprise and its entities.

Site Tree

In PeopleSoft eStore, a hierarchical structure that controls navigation, as well as content and behavior within the header, footer, and left margin areas of the web page template.

Slice

The span of time into which an element is segmented as a result of element segmentation. Unlike a segment (or period), a slice does not represent a separate gross-to-net process since it affects only a limited set of elements within a period or segment. Like segments, slices have their own begin and end dates.

Slice Dimension

A model dimension used to restrict user access to the system. For example, a product manager's access to the system can be restricted to only the products he or she is responsible for by defining "Products" as a slice dimension, and assigning this person the members of the "Products" dimension he or she can access.

Slice and Dice

Another term for multidimensional analysis. When your data has three (or more) dimensions, you can think of it as being arranged in a cube (or hypercube), with each side representing a dimension. When you analyze the data, you "slice" off part of the cube or "dice" it to get to an individual cell.

Slotting

In PeopleSoft Workforce Rewards, a process by which the system establishes the target market compensation rates to use for compensating workers in non-benchmark jobs.

Social Security Number

Nine numeric digits assigned to an individual by the Social Security Administration. Also known as a Taxpayer Identification Number (TIN).

Source

The Source table stores valid journal entry and posting sources. These can include job titles (such as CFO), user IDs (such as CLERK123), PeopleSoft General Ledger processes (such as Consolidations), or other applications (such as PeopleSoft Payables).

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.

Sparsity/Density

A multi-dimensional concept of whether data exists at intersections of dimensions. If a cube has many dimensions, but little or no data in some of those dimensions, the cube is considered sparse. Sparse cubes take up unnecessary disk space and reduce calculation performance. The goal is to create dense cubes and only use dimension intersections where data actually exists.

Special Accumulator

A device that accumulates earnings from different sources for a specific purpose. 401(k), pension and retirement plans use special accumulators. A 401(k) plan might use a special accumulator to calculate a deduction using regular, vacation, and overtime earnings. Special accumulators can add to or subtract from a pool of earnings.

Special Payments

A payment that occurs once or under special circumstances (e.g., back pay interest, lump sum leave, bond refund, longevity bonus, compensatory time reimbursement, death payment, severance pay, separation bonus, etc.).

Special Rates

Higher salary rates for specific grade levels and occupational groups determined by OPM for employees working in specific geographic areas. Each area is assigned a separate Schedule Number.

Specialist

A member of a stock exchange who maintains a fair and orderly market in one or more securities. A specialist or specialist unit performs two main functions. executing limit orders on behalf of other exchange members for a portion of the floor broker's commission, and buying or selling for the specialist's own account to counteract temporary imbalances in supply and demand, preventing wide swings in stock prices.

Specialized Experience

Experience that has equipped the applicant with the particular knowledge, skills, and abilities to perform successfully the duties of the position and is typically in or related to the work of the position to be filled.

SpeedChart

A user-defined shorthand key designating 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.

Spending Limits

See Planning Targets.

Split and Join

In PeopleSoft Demand Planning, the process of subdividing a forecast so that multiple users can make changes to their portions of the forecast. After changes are complete, the portions are joined back into a single forecast.

Split Deduction

Deduction that you create by splitting an existing deduction into two deduction items. The new deduction retains the original item ID with an added suffix number.

Split Shift

Periods of productive time split up by period of non-working time; example. a time reporter comes to work as a busboy for the lunch shift from 12-2 p.m. and then returns to work from 6-8 for the dinner shift (see Scheduling)

Split Stream Processing

The matching of a payment's cash information with the payment advice information when they have been received as separate transmissions through EDI and lockbox.

The uniting of the payment cash with the payment advice when they have been received as separate information through EDI.

Spokesmen Committee (Sprecherausschusse)

In Germany the Spokesmen Committee represents the interest of the management in your company before the ownership. The Spokesmen's Committee is consultative in nature, although they play a co-determination role on individual employment contracts, hiring, and dismissals. They also play a role in monitoring employment fairness, equity, and non-discrimination in terms of nationality, race, religion, sex, and age.

Spouse Demonstration J&S

In the PeopleSoft Pension Administration system, an informational-only form of pension payment that tells what the spouse's total benefit would have been if the retiree had chosen the spouse as the beneficiary rather than a nonspouse beneficiary. You cannot pay pension benefits based on this form because it is informational only.

Spouse Eligibility Alias

In PeopleSoft Pension Administration, a Custom Statement that defines any criteria that must be met before the plan will provide an Automatic Spouse Benefit. For example, the plan may require that the employee and spouse be married a full year before they are eligible for an automatic spouse benefit.

Spouse Eligibility Statement

See Spouse Eligibility Alias.

Spread

Depending on the context, either (1) the difference between the bid and asked prices for an over-the-counter stock, or (2) the difference between an option's exercise price and the market price at the time of exercise (i.e., the profit component of the exercise).

SQL Objects

Used to create rules that are more complicated than templates or actions and conditions allow—select statements, insert statements, table joins, and sub-queries

ST (Strategic Trust)

See Strategic Thrust

Staged Date

The date an item was received into the inventory business unit.

Staging ID

An identifier for a putaway plan. The inventory system sequentially assigns Staging IDs when it creates the putaway plan.

Standard Cost

A predetermined, fixed cost associated with an **Inventory Item** or **Forecast Item**, representing detailed estimates of each element of cost entering into the purchasing or manufacturing of an item. Standard cost is used when minor variations in an item's cost are not needed. The use of standard costs also enables management to determine how much an item should cost (Standard), look at how much it does cost (Actual), analyze the differences between the two and their causes (Variances), and compute economic order quantity.

Standard Form (SF)

A standardized form for interagency use by the Federal government. The SF prefix is the most common but not exclusive one in usage.

Standard Price

In PeopleSoft Demand Planning, the standard selling price associated with a **Forecast Item**. The price can be introduced into the system directly in forecast item maintenance or indirectly using the demand transfer interface. At higher levels in the view where there is no standard price available, the summarization function can be set up to develop one.

Standard Unit of Measure

The smallest unit of an item that a PeopleSoft application tracks.

Startup Data

In PeopleSoft Pension Administration, accrued Service, Cash Balance Account, or Employee Account data loaded into the system in the form of an opening balance and “as of” date. The alternative would be to load the entire accrual history.

State Record

The State Record is a PeopleSoft record, keyed by process instance, that must be created and maintained for each Application Engine program. The State Record defines the fields that an Application Engine program uses to pass values from one SQL statement to another.

Static Group

An employee group in Time and Labor that enables you to control its creation and maintenance. The group remains the same at all times until you change it.

Static Policy Controls

Determines how a static (versus time-phased) **Inventory Policy** is to be calculated. Static controls use period and average methods and their arguments.

Statistical Account

An account that has an associated unit of measure, used for tracking and monitoring statistical data. For example, the Workstations account uses EA (each) as a generic unit of measure, while the Floor Space statistical account might use square feet and the Work Days account would use days.

Statistical Code

The unit of measure used for tracking and monitoring statistical data. For example, using a statistical code of WS may represent the number of Workstations.

Statistical Forecast

In PeopleSoft Demand Planning,, a forecast developed at each level of the forecast pyramid and that considers the item’s history in isolation.

Status Checking

In PeopleSoft Projects, a control feature that can be applied to transactions coming into Projects from cost feeder systems. If the incoming transaction does not conform to predetermined status and analysis conditions, an online warning will display or the transaction will be rejected.

Status Position Code

A code that identifies the various conditions of a position, e.g., frozen, classified, etc.

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

A secondary level or subcategory within the primary pay level (depending upon pay plan, different employees may have a different number of steps within their primary pay level).

Step Progression

In PeopleSoft Workforce Analytics, a pay increase granted to an employee or group whose salary plan includes steps within grades. Each step increase is a step up the pay range for the employee.

Stock

In corporate finance, the form in which an owner's interest is represented, distributed in units known as shares.

Stock Administrator

An individual who administers and manages the corporation's benefits and/or equity compensation plans. This individual serves as the contact for transfer agent and broker inquiries. Stock Administrators manage Stock Option Plans, Employee Stock Purchase Plans, Restricted Stock Award Plans, and Stock Bonus Plans.

Stock Appreciation Rights (SAR)

A contractual right to receive, either in cash or employer stock, the appreciation in the value of the employer's stock over a certain period of time. A SAR can be used alone or in tandem with Incentive Stock Options (ISO/SAR) or Nonqualified Stock Options (NQ/SAR). PeopleSoft Stock Administration supports only tandem SAR's.

Stock Awards

Stock allocations that are processed in the Manage Variable Compensation business process. Stock Administration creates stock grants from finalized stock awards.

Stock Exchange

An organized marketplace in which bonds, stocks, and common stock equivalents are traded by members of the exchange, acting as agents (brokers) and as principals (dealers or traders). Such exchanges have a physical location where brokers and dealers meet to execute orders to buy and sell securities. Each exchange sets its own requirements for membership.

Stock Option

A contractual right granted by the company, generally under a stock option plan, to purchase a specified number of shares of the company's stock at a specified price (the exercise price) for a specified period of time (generally five or ten years). Assuming that the exercise price is the

same as the fair market value on the grant date, the option will become more valuable if the fair market value goes up, because the option effectively gives the optionee the right to buy stock in the future at a discount.

Stock Price

The price per share of a company's stock. See, also, "share price."

Stock Purchase Participant

An individual who participates in the corporation's Stock Purchase Plan.

Stock Purchase Plan

A type of broad-based stock plan that permits participants to use payroll deductions accumulated over a period of time to acquire stock from the company.

Stock Split

A change in the capitalization of an issuer that increases or decreases the number of securities outstanding, and adjusts the value of the securities accordingly, without a corresponding change in the assets or capital of the issuer. For example, if an employee has options to purchase 25 shares at \$10 per share and the company has a 2-for-1 stock split, the employee thereafter has the option to purchase 50 shares at \$5 per share.

Stock Swaps

A payment method that can be used to cover the cost of the exercise price and taxes depending on whether it is allowed by the plan. When an employee elects to exercise a stock option by means of a stock swap, they surrender already-owned shares of stock to pay the total required option exercise price and/or taxes for the option being purchased. The surrendered shares are usually valued at the fair market value of the company's stock on the date of exercise.

Stock Trading Symbol

The three or four letter symbol used to identify a company's stock on the stock exchange where it trades. Also known as a "ticker symbol".

Stock Withholding

A cashless method of satisfying the withholding taxes due upon the exercise of a stock option by authorizing the company to withhold from the shares being exercised a number of shares equal to the taxes.

Stockholder of Record

Person or entity, often a broker or the Depository Trust Company, named on the issuer's or transfer agent's stock record books as the owner of shares held in "street name." The stockholder of record acts in part as a way of safekeeping stock certificates that might otherwise be lost by the beneficial owner, and also in order to keep the identity of the beneficial owner confidential from the company.

Stock-In Probability

A replenishment option for defining transfer parameters for PeopleSoft Demand Planning or Inventory Planning upload files. The option is the percentage of time you want to have the item on hand for the **Business Unit** and is used to calculate safety stock.

Stop Time

Out punch

Storage Area

A division of a **Business Unit** used to store material and to track **Inventory Transaction**. Storage areas might include shipping and receiving docks, staging areas, warehouse zones, and inspection and quality control departments. Each storage area can be divided into a maximum of four levels, with each level representing a physical subdivision of the area.

Storage Level

A hierarchical subdivision of a storage area.

Storage Location

The combination of a storage area and that area's most detailed storage level. This is the smallest definable physical space within an **Inventory Business Unit**.

Strategic Initiatives

In PeopleSoft Balanced Scorecard, actions the organization must take to implement strategy. May be temporary or short-term in nature.

Strategic Thrust (ST)

In PeopleSoft Balanced Scorecard, four to five statements or paragraphs that summarize the core components of an organization's strategy. Strategic thrusts describe the key areas across which a scorecard is balanced. They are themes or goals your organization is striving to achieve; more specific descriptions of what you must do to achieve that goal are defined by critical success factors. Key performance indicators may be attached to strategic thrusts as long as there aren't critical success factors below them, but typically strategic thrusts aren't directly associated with key performance indicators.

Strategy Tree

In PeopleSoft Balanced Scorecard, the hierarchical relationships of the objectives your organization is striving to achieve. Used as the foundation for a scorecard, and typically balanced across four major categories: Financial, Customer, Learning and Growth, and Internal Processes. These are made up of Vision, Strategic Thrusts, and Critical Success Factors.

Stratification Engine

A support module that structures the volume of financial accounts and balances at a large financial institution to a manageable scale for processing by the PeopleSoft Funds Transfer Pricing (FTP) and PeopleSoft Risk Weighted Capital (RWC) applications. It categorizes data by a range of values and summarizes data based on rules you define for FTP and RWC.

Stratification Wizard

Stratification Wizard is a tool you can use to quickly create new stratification rules or update the existing rules. Stratification Wizard prompts you for each of the possible source and destination fields, grouping operations, and summarization actions to be performed. Stratification Wizard enables you to stratify your data according to tiers, discrete values, periodic increments, and numeric increments. It also enables you to leave the data aggregated.

Streams

An optional feature that enables you to reduce processing time by processing groups of payees simultaneously.

Street Name

See "Stockholder of Record".

Stretch

In a Goals Matrix performance scale (In PeopleSoft Workforce Analytics), this is the level of performance for which an employee achieves maximum pay out. Performance above this level receives no greater pay out.

Strike Price

The price per share which must be paid in order to exercise the stock option. The strike price is typically the fair market value of the stock on the grant date. Also known as the "exercise" or "grant" price.

String constant

String constants are delimited in PeopleCode by using either single (') or double (") quote marks.

Strip Funding

One of several methodologies used by PeopleSoft Funds Transfer Pricing (FTP) to derive maturity when calculating FTP rates based on matched maturity funding. This approach matches the projected cash for the instrument in each time period, with a specific cost of funds rate for that cash flow. The FTP rate for the instrument is then calculated by weighting the cost of funds rate for the cash flow in each time period by the term of the cash flow.

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.

Style File (Verity)

Collection style refers to a set of configuration options that are used to create the indexes associated with a collection. A collection has one collection style and it is defined in a set of style files before creating the collection.

SubCustomer Qualifier

A value that groups customers into a division for which you can generate detailed history, aging, events, and profiles.

Sub-Process Section

A type of section you can add to a process list. Sub-process sections are especially useful for performing iterative processes such as gross ups (calculating the gross amount for a given net amount). You can include conditional logic within a sub-process section.

Subscription

The process of mapping fields, selecting data parameters and submitting the information to an outside vendor.

Substitute Item

In PeopleSoft Manufacturing, an item that can be used when there are no primary components available in inventory or when there is a long-term shortage of the original item. The substitute item can be defined at three levels: setID, business unit/item, and bill of material/engineering bill of material.

Subtask

A lower-level Planning task in a schedule's hierarchy that rolls up into a parent task. For example, an operation performed on a production ID would be a subtask of the production order.

Summarization Process

See Rollup.

Summary ChartField

A feature for creating 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 Forecast

In PeopleSoft Demand Planning, a type of forecast that results from adding up the adjusted forecast totals from the next lower level, meaning the sum of the children's forecasts for the parent. The summary forecast at level one (1) is always zero since there is not a logical lower level.

Summary Ledger

An accounting feature used primarily in allocations, inquiries, and PS/nVision reporting to store combined account balances from detail ledgers. They increase the 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. In effect, 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.

Summed Adjustment Type

When the system finds multiple summed discounts, they are added together, and applied once.

Super Tree Utility

A PeopleSoft Enterprise Warehouse utility that enables you to combine multiple effective dates of a tree into one. This super tree contains all tree changes for a certain period of time enabling you to analyze effective dated trees more easily.

Supplemental IRR

An IRR type used when a federal employee's retirement deductions were originally under-reported. An agency can create a Supplemental IRR to report the difference in the retirement deduction amount to the Office of Personnel Management (OPM).

Supplemental Tax Method

A payroll tax calculation method that uses a straight percent rather than allowances. The percentage depends on state requirements.

Supply Chain Warehouse

See Warehouses.

Support Costs

Activity costs not directly connected to production.

Support Modules

The support modules are a collection of engines and analysis models that derive values, rates, financial calculations, and prices. PeopleSoft Analytic Applications take this data and use it for further processing. The support modules perform processes that are used in the financial services industry. They are only utilized by two of the PeopleSoft Analytic Applications: Risk Weighted Capital (RWC) and Funds Transfer Pricing (FTP).

Support Team

A group of people working together to sell to and/or support a customer. You can assemble support teams and associate them with ship to customers, quotes, sales orders, and **Buying Agreement**.

Supporting Element Overrides

Provide a mechanism to override various supporting element types, such as brackets, dates, durations, formulas, and variables, at various different levels.

Supporting Elements

Supporting elements are building blocks for other elements. In PeopleSoft Global Payroll, they are used in combination with other elements to create rules. They are not stand-alone. Typical supporting elements are arrays, brackets, rounding rules, and fictitious calculations.

Suspend Exercise

As a condition of a leave of absence, a company may stipulate to restrict the exercise of shares during the leave or for a period of time. Only applicable if the Stock Action is LOA.

Suspend Vesting

As a condition of a leave of absence, a company may stipulate to suspend vesting of shares. Only applicable if the Stock Action is LOA. If a company does not Suspend Vesting then the Vest Deferral Grace Period and Service Rule are not applicable.

Suspended Item

In PeopleSoft Demand Planning, an item suspended by the system. The suspension is due to the lack of demand for the number of periods defined for the item's **Control Group**.

System Element

In PeopleSoft Global Payroll, system elements are delivered and maintained by Peoplesoft. There are two types of system elements: database system elements and system-computed elements. Database system elements contain payee-related data that can be used frequently in a calculation, such as department ID, location, and personal data. System-computed elements are automatically populated by the payroll process.

System Functions

A list of all activities that the system supports, along with their associated General Ledger distributions.

System-Defined Count

The PeopleSoft Inventory feature that employs user-defined criteria to begin the count creation process.

System-Defined History

Any statistical information updated by the posting and aging programs, maintained to reflect customer credit standing.

T**T+3**

The obligation in the brokerage business to settle securities trades by the third day following the trade date. "Settlement" occurs when the seller receives the sales price (less the broker's commission) and the buyer receives the shares.

Table

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

Table Lookup

A utility in PeopleSoft Pension Administration that finds an unknown value based on a known one in your data set. For example, you can look up an interest rate based on a year, or an actuarial factor based on an employee's age.

TableMaps

In the PeopleSoft Enterprise Warehouse, TableMaps define the physical relationships between related tables. TableMaps allow you to define "families" of related tables and the columns that define the key relationships between the tables.

TableSet

A group of rows across control tables identified by the same SetID.

TableSet Sharing

Specifying the control table data for each business unit so that redundancy is eliminated.

Tardy

The circumstance when a time reporter reports for work after the scheduled start time

Target

In a Goals Matrix performance scale (In PeopleSoft Workforce Analytics), this is the performance level your organization establishes as the norm for performance and pay out.

Target Cost

A desired target cost (for production, engineering or marketing) is found by subtracting the desired profit margin from a competitive or estimated price.

Target Currency

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

Target End Date

The intended end date for employee schedules in Time and Labor reporting. You establish a target end date, and depending on the work schedule templates, the application determines what the actual end date should be.

Target Grade

Highest obtainable grade for a position.

Target Matrix

In PeopleSoft Workforce Rewards, the Target Matrix defines the level of award to be paid based upon a predetermined level of performance that a defined measure is evaluated against.

Target Plan

In PeopleSoft Workforce Rewards, a plan for distributing compensation awards, in which the level of the award is linked directly to a predetermined level of performance that a defined measure is evaluated against. For example, a Target Plan for a Sales group might be as follows: “As a group, increase the business unit’s net income by 10% and each member of the group will be eligible for an award equal to 5% of base pay.”

Target Rate

In PeopleSoft Workforce Rewards, calculated market rates you choose to use as new target pay rates for the jobs in your organization. Think of these target rates as pay guidelines, in support of your company’s overall pay strategy.

Target View

In PeopleSoft Demand Planning, the of the views being reconciled during the **Cross-View Reconciliation** process. During the process, fields for reconciliation are defined for both a source and target view.

Tariff

In Germany a Tariff is a contract between the employee's unions and the employers' association, defining labor agreements on issues such as standard working hours, income, and vacation. This contract is valid for all of Germany for the business or industrial sector the company is working in (such as **Banking** or **Metal**).

Tariff Area

In Germany, additional labor agreement terms beyond those in the Tariff, such as salary plans or employee reviews, can also be applied based upon the Tariff Area. The Tariff Area is often split along regional lines (such as **Bavaria** or **Berlin**).

Task

See **Manufacturing Task** and **Planning Task**

Task

A piece of work assigned to or demanded of a person; a unit of work (see Time Reporting)

Task Entity

Individual component of a task; for example, Project ID, Activity ID, Work order, Department, Company, Business Unit (see Time Reporting).

Task Profile

A way of viewing or establishing where to allocate employee task information for a day and time. The task profile fields that appear on the page are established by the *Task Profile Template*.

Task Profile

Entity that establishes the default values for optional and required task elements. This can be for single or multiple tasks. (i.e. default values based on hours allocation, percentage distribution, equally distributed or by prompt) (see Time Reporting)

Task Rules

A methodology that is applied to scheduled, reported and payable time to allocate or redistribute task assignments (see Time Administration)

Task Transfer

Department transfer

Task Values

The customer defined value for a specific task element i.e. Customer 1, project 1, etc (see Time Reporting)

Taskgroup

Identifies the default time reporting templates, task template, and task profile(s) for time reporters that share the same task reporting requirements.

Taxable Benefits

Any employer contributions that are subject to Federal Withholding Tax.

TDS (Transfert de Données Sociales)

In France the TDS is a social security transfer report, submitted on magnetic media to the government.

Team Member

An individual who is part of a support team. Each team member may be in a commissionable or non-commissionable role.

Technical Scenario

In PeopleSoft Enterprise Warehouse technical scenarios allow you to set up the object type values that the Resolver uses to chunk the record/TableMap you'll resolve. Technical scenarios allow technical users to define chunking criteria that enable chunks to run in parallel. This allows for parallel data processing.

Template

A portal template is simply HTML code, associated with a web page, to define the style and layout of the page. Templates allow a developer to build an HTML page by combining HTML from a number of sources. Templates do two basic things: define the layout of the page, and define where to get HTML for each part of the page.

Template Pagelet

One piece of an overall template. For example, in a given template, there may be one template pagelet for the universal navigation header and one template pagelet for the target content.

Temporal Constraint

A relationship between Planning tasks that defines their sequence and timing in a schedule. Temporal constraints cannot be violated by the Optimizer. PeopleSoft Planning constraints include finish to start, start to start, finish to finish, start to finish.

Temporary Continuation of Coverage (TCC)

The TCC program, as prescribed by the OPM, requires Federal agencies to provide to separating Federal employees the opportunity to temporarily continue their FEHB coverage for up to 18 months (unless involuntarily separated because of gross misconduct), provided the individual pays the full cost of coverage, including both the employee and government share and a two percent administrative charge. Agencies may elect to provide this service in-house or enter into cross-servicing agreements with another Federal agency.

Tenor

Used by the PeopleSoft Funds Transfer Pricing (FTP) application to refer to the maturity of an instrument. It represents the length of time an instrument is available as either a source or use of funds. The FTP application calculates the transfer price for an instrument, based on the marginal cost of funds of similar liquidity and tenor.

Term Certain and Continuous Payment Option

See Certain and Continuous Payment Option.

Term Certain Payment Option

See Certain Only Payment Option.

Termination

A transaction in which an employee ceases to be an employee of the corporation.

Threshold

In a Goals Matrix performance scale (In PeopleSoft Workforce Analytics), this is the minimum threshold for adequate performance, the designated level of performance below which it is inappropriate to pay incentives.

Think-time process

Think-time functions suspend processing either until the user has taken some action (such as clicking a button in a message box), or until an external process has run to completion (for example, a remote process).

Three-Tier

A three-tier architecture introduces an intermediary application server between the client workstation and the database server to improve performance. Within PeopleSoft, the application server sends the SQL to the database and then returns results to the client in the form of lightweight Tuxedo messages.

Threshold Checking

In PeopleSoft Projects, a control feature that can be applied to transactions entered directly into Projects or integrated into Projects via the INTFC_PROJ_RES table. If the transaction exceeds a predefined tolerance, a warning will display or the transaction will be rejected.

Thrift Savings Plan (TSP)

A voluntary retirement savings and investment plan for Federal employees administered by the Federal Thrift Investment Board.

Ticker Symbol

The three or four letter symbol used to identify a company's stock on the stock exchange where it trades. Also known as a "stock trading symbol".

Tiers

In the financial services industry, Tiers are ranges that you set up for stratifying your instrument data (such as amounts, rates, and numbers) into specific groups. You define Tiers within Stratification Engine.

Time Administration

A process which provides four [separate] different online tools for creating, maintaining, and applying an organization's compensation, task, and exception rules to both reported and scheduled time. templates, actions and condition, SQL objects, and user exits

Time and Labor Period

A distinct, configurable period of time used by all the PeopleSoft Time and Labor processes (see Time Reporting)

Time and Labor User

Either a Time Reporter or a Time Manager

Time Capture Device

Third party system or methodology for collecting elapsed or time capture device time, i.e., time capture device, IVR, Fax, etc. (see Time Reporting)

Time Collection

A Time and Labor feature that collects positive and exception time reports, applies appropriate business rules and edits to the reported time to ensure validity and reasonableness, and returns errors and questionable items to the time reporter for correction or scrutiny. Time collection is also responsible for scrutinizing future (previously) posted time information for correctness when those reports are ready for use.

Time Collection Device

A group of time collection device lumped together and named for ease of assignment to employees. In other words. clock group 1 is made up of clock 1, and clock 2. Employee 123 is assigned to clock group 1 and can then punch in at either clock 1 or clock 2. (see Understanding Time Collection Device)

Time Collection Device time

Reporting time by recording actual starts and stop times (see Time Reporting)

Time Dimension

Determines how date-related information is presented in a **Cube View**. This dimension defaults to a two-level hierarchy consisting of the **Inventory Policy** year and a standard period, such as monthly.

Time Fence

In PeopleSoft Planning, a user-defined parameter that specifies the business rules to be used in the generation of the plan. PeopleSoft Planning time-fence types include start of time, end of time, planning close date (demand time fence), purchase order fence, leveling fence, action message cutoff, and planning time fence.

Time Manager

An individual who supervises Time Reporters

Time Period

A period of Time used in Time and Labor rules processing. You can categorize time periods in terms of days, weeks, or months. You establish day, week, or month-type periods for use when you apply rules for compensation, holidays, and so on.

Time Report

A payroll time and/or labor distribution time report for an employee for any date within the employee's current period.

Time Reporter

Any employee or contractor for who time is reported or generated in PeopleSoft Time and Labor.

Time Reporter Information

Values associated with the Time Reporter that are displayed when entering or viewing reported time and facilitate the processes of Time Reporting and Time Management (see Time Reporting)

Time Reporting

Any information required by a business unit that can be attributed to an individual employee (worker/contractor) and can be expressed in hours.

Time Reporting Code

A hybrid of two PeopleSoft objects: the Payroll Earnings Type and the Human Resources Absence Type. The Time Reporting Code represents the level at which a business actually needs to track employee time to support all of its administrative and compensation needs.

Time Reporting Code Type

Categorization of a time reporting code. Valid categories include: units, amounts, hours or a combination of hours and amounts (see Time Reporting)

Time Reporting Group

See Group [Time and Labor].

Time Segment

For Service, Cash Balance Accounts, and Employee Accounts, employees can accrue benefits differently at different times. The period of time during which employees use a particular rule is that rule's time segment.

TimeSpans

Relative periods, 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.

Tolerance

In PeopleSoft Projects, a value that is defined at either the project or activity level as either a percentage of a project's funding or an actual amount.

In commitment control, the percentage over budget that you allow, excluding revenues applied to increase budget limits, before the system creates an exception.

Total Authorized But Unissued

The combined total number of shares from Shares Available to Issue plus Total Options Outstanding.

Total Compensation

In PeopleSoft Workforce Analytics, this is generally the officially recognized compensation provided an employee in the course of their employment with an organization; includes both direct compensation and benefits compensation.

Total Compensation Management

The ability to track and report on all types of cash programs, non-cash programs, benefits and deferred compensation for all current workers, ex-workers and individuals associated with ex-workers who receive compensation due to the employment of the ex-worker.

Total Non-Compensation

In PeopleSoft Workforce Analytics, generally, this is the often unrecognized compensation an employee receives in the course of their employment with an organization; it includes Learning and Development compensation such as training, and Workplace Environment Compensation such as telecommuting privileges or other prerequisites.

Total Options Outstanding

The number of company shares currently held by shareholders as tracked by the transfer agent. Derived by using the number of Prior Outstanding plus Grants, less Exercises and less Cancellations.

Total Rewards

In PeopleSoft Workforce Analytics, this is the total rewards provided to an employee by their employment with an organization; it includes their officially recognized total compensation, and less often recognized total non-compensation.

Tour of Duty

The scheduled days and hours per day of attendance at a duty station for an employee.

Tracking Signals

PeopleSoft Demand Planning, a forecasting tool that detects bias in the forecast and provides an early warning of an unstable forecast. There are six tracking signals associated with each **Forecast Item** that correspond to the six most recent historical periods.

Trade Payment

An authorization for a customer deduction in a Promotion application.

Training Report 2483

The Training Report 2483 is a French regulatory report used to declare vocational training your company has provided to your employees. It is also known as the Declaration 2483 Report. The purpose of the report is to receive tax deductions from the government based upon the amount of money your company has spent on training.

Transaction

A named command with optional named and typed inputs and outputs. The associated external system or the Business Interlink Plug-in understands this command. The types of inputs and outputs are based on a set of generic types.

Transaction

A named command with optional named and typed inputs and outputs. The associated external system or the Business Interlink Plug-in understands this command. The types of inputs and outputs are based on a set of generic types.

See also **Inventory Transaction** or, for PeopleSoft Projects, **Resource Transaction**.

Transaction catalog

Lists transactions used to interface to the external system.

Transaction Code

In PeopleSoft Projects, an additional field on each resource transaction that is used in conjunction with accounting entry templates. Transaction codes enable you to deal with exceptions to your accounting entry templates without having to create additional transaction types. You can set up separate accounting entry templates for resource transactions containing the transaction codes you create. The accounting entry templates for those resource transactions lines can then use the same transaction types, but specify different accounts.

In PeopleSoft Asset Management, transaction codes identify special asset transactions and are used in conjunction with transaction type to create accounting entries.

Transaction Code

Identifies what action has taken place against the position.

Transaction Costing

See Multidimensional Costing

Transaction Currency

In the financial services industry, the original currency in which a company conducts its business activities. When a company has multinational operations, it may use different transaction currencies. These are translated to the base currency for consolidation and reporting of financial results.

Transaction Date

The date a transaction actually occurred as opposed to the date the transaction is recognized—the accounting date (although the two dates can be the same).

Transaction Dated

Data aggregated over a date range.

Transaction group

The package can contain one or more transaction groups. Each transaction group is a set of transactions of the same type, with the same trading partners involved.

Transaction Loader

The SQR in PeopleSoft Asset Management that transfers load lines from the Loader tables into the PeopleSoft Asset Management Tables as assets and open transactions.

Transaction Tables

In the PeopleSoft Enterprise Warehouse, these are tables that contain dynamic information and are keyed by business units.

Transaction Type

The building blocks of accounting entry templates in PeopleSoft Asset Management and Projects. For each transaction type you create you define specific transaction lines. The transaction lines are then transferred into accounting entry templates. In the accounting entry templates each transaction line is assigned a specific general ledger account.

Transactional System

A business application for performing the business transactions that keep your company running. Transactional applications, and the databases that support them, are optimized for quick transaction processing. Because they are constantly changing and are not optimized for data retrieval, transactional system databases are not usually the best source of data for analysis.

Transfer Agent

An individual or firm who that keeps a record of your shareholders and the number of shares they own. Transfer Agents also issue new share certificates and cancel old certificates. Unlike Brokers, Transfer Agents are not responsible for selling stocks. Instead they are primarily concerned with maintaining records on all stocks which your company has issued.

Transfer Forecast

In PeopleSoft Inventory Planning, a Generation process option that transfers the forecast from the target view in Demand Planning forecasts. The process only transfers items from Demand Planning that have been set to update the **Inventory Policy**.

Transfer Punch

The start of a work period that specifically denotes a change in task and usually compensation-related characteristics

Transfer Type

An interunit transfer setting PeopleSoft Production Planning and Enterprise Planning use to determine where it will obtain item data for transfer tasks. If the type is a supply or demand transfer task, the Planning engine only processes the transfer item for a single location, reducing the time for plan processing. If the transfer type value is both, the Planning engine processes the transfer item using data from both the To and From units.

Transfer Worksheet

A work space for transferring an open item from one customer to another.

Transferable Stock Options

Options that may be transferred by the optionee, generally only to a family member or to a trust, limited partnership or other entity for the benefit of family members, or to a charity.

Translate Table

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

Translate Table

A system edit table that stores codes and translate values for the miscellaneous fields on the database that do not warrant individual edit tables of their own. In most cases PeopleSoft maintains the Translate Table.

Transport Rate

The Transport is a statutory deduction in France. Each establishment has a rate, and the URSSAF notifies establishments of this rate on a yearly basis. This deduction is used by the region to subsidize transportation, and maintain and build roads.

Transportation Lead Times

The transportation lead time is the in-transit interval from the date and time a shipment leaves your warehouse (**Inventory Business Unit**) to the date and time it arrives at your customer's receiving dock. The transportation lead time is used in calculating the scheduled shipment and scheduled arrival dates on the order when you enter either a requested arrival date or a requested shipment date.

Travel And Relocation Date

Length of time an employee must remain in the Government after the Government has paid to relocate him/her from one official duty station to another or for initial appointment.

TRC Program

A program that runs the level at which an organization actually needs to track employee time to support all of its administrative and compensation needs. TRCs are assigned to TRC Programs, which are ultimately assigned to workgroups. Multiple Workgroups can share these TRC Programs.

Treasury Interface files

These are DOS-based files generated by PeopleSoft in accordance with FMS file layouts for transmission of payment data to one of the FMS' Regional Financial Centers.

Treasury Position Code

In the financial services industry, this is a lookup code used for off-balance sheet treasury position accounts, such as foreign exchange, derivatives, precious metals, or any other account position that is the result of trading room and treasury operations.

Treasury Stock

Shares of a company's stock that have been repurchased or otherwise reacquired by the company and are "held in treasury." Whether the treasury shares count as "issued" or as "outstanding" shares of the company is a matter of state corporate law. Generally, a company may not vote its own shares held in treasury.

Treasury Stock Method

The method of calculating primary and fully diluted earnings per share when common stock equivalents such as unexercised stock options exist. Required under generally accepted accounting principles.

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.

Tree Compare Utility

A PeopleSoft Enterprise Warehouse utility that enables you to compare effective dates for trees. The results page shows nodes that have been added, deleted, or moved from one parent to another. You may also view the detail objects that have changed.

Tree control

Tree Control is a hierarchical search tool that you can embed in a panel. Tree Controls give the user a view of hierarchical data structures and enable them to drill down through the hierarchy to a particular row of data.

Tree Denormalizer

The Tree Denormalizer Application Engine process converts trees into multi-column data format so they can be used by third-party OLAP or ROLAP tools.

Trigger

See Event Trigger.

Trustee Extract

A PeopleSoft Pension Administration data extract containing data that a third party needs in order to produce pension checks.

Turnover Costing

In PeopleSoft Workforce Analytics, this is a calculation of the cost to the organization of employee turnover, in dollars.

Turnover Rate

In PeopleSoft Workforce Analytics, the rate that employee's are leaving the company.

TUXEDO

BEA's middleware product used to manage transaction queues, server process initiation, system administration, time-outs, data encryption, compression, logging and other application server processing.

Two-Tier

A two-tier architecture refers to the traditional client/server model in which a client workstation connects to and sends SQL directly to the database server.

Type of Appointment

Indicates the specific type of appointment, e.g., part-time permanent, full time temporary, etc.

U

Underlying Security

The security underlying a stock option that an optionee has the right to buy, or the security underlying a convertible security.

Underpayment Adjustment Limit

The maximum amount or percent above which underpayment adjustments are not allowed for a given business unit.

Underwater Option

When the current market price is below the option exercise price. When an option is underwater, it would cost more than the underlying stock is worth to exercise the option. Such options are also described as being "out-of-the-money."

Underwriter

An investment banking firm that actually buys the shares from the company in a public offering and then resells them (at a slightly higher price) to its customers.

Unexpected Losses

In the financial services industry, these occur when the economic capital is exhausted and the insolvency rate is exceeded. Unexpected losses are determined by a targeted insolvency rate (confidence level); for example, a 99.7% confidence level indicates that there is a 0.03% estimated probability that the unexpected losses will exceed economic capital (or shareholder equity).

Union Code

Part of a group of defaults assigned to job codes. Union code may be used by human resources to group similar jobs or bargaining units together, dependent on individual company parameters.

Unit Code

In the financial services industry, Unit Code is used as an alternate means of measuring the relative size of companies participating in external surveys. A typical measure would be the number of employees in a company. The concept of unit is generic enough that the units can be other measures besides number of employees. For example, in the hospital industry the unit could be the number of hospital beds. Or in the hotel industry the unit could be the number of rooms.

Unit of Measure (UOM)

A type of unit used for quantifying in PeopleSoft systems. Depending on the application, units of measure might describe dimensions, weights, volumes, or amounts of locations, containers, or business activities. Examples include inches, pounds, workhours, and standard cost dollars.

Unit of work

Each transaction group includes one or more individual units of work. A unit of work is a single transaction that you want to commit or rollback as a whole.

Unitize Assets

The process of unitizing a single load line, usually originating from a different application, into multiple assets in PeopleSoft Asset Management.

Univariate Forecasting Technique

In Enterprise Planning and Simulation, the Univariate Forecasting Technique is a forecasting method that uses only the recorded history for the value to forecast its future.

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.

Unscheduled Punch

A punch that is made by a time reporter who was not scheduled (see Time Reporting)

Unvested Shares

Unvested stock options are options that have not vested and, therefore, are not exercisable.

URI

A URI does not include the query string (the text following a ? on the URL). You can think of it as a subset of the URL that points to the resource, but does not include any parameters being passed to that resource. From the above example, the URI portion of the URL is as follows:

`http://serverx/InternetClient/InternetClientServlet`

URL

In this document, the term URL refers to the entire query string. The following is an example of a URL:

`http://serverx/InternetClient/InternetClientServlet?ICType=Script&ICScriptProgramName=W
EBLIB_BEN_401k.PAGES.FieldFormula.iScript_Home401k`

URSSAF Code

The URSSAF is the body responsible for ensuring payment of Social Security contributions by all French employers.

Useful Life

The amount of time an asset may be depreciated.

User Data

PeopleSoft Demand Planning, data held in user-defined fields. These fields provide for storage of additional data that is not supplied by the standard set of fields in the system. The fields can also become part of the key for the **Forecast Item** at each level within the **Forecast View**.

User-Defined History

A summary of customer receivables activity that is defined by the user.

User-Field Code

PeopleSoft Demand Planning, a definition of a set of user-defined fields that contain data specific to the installation.

V**Valuation**

The way a company represents the value of a non-monetary award such as stock.

Value allocation

A process in PeopleSoft Workforce Planning, by which you assign an overall monetary value to a competency strategy for your organization, and assign a weight or importance to the roles,

competencies and accomplishments in the strategy. The system then allocates a breakdown of the value to roles, competencies, and accomplishments in the strategy based on their relative weighting or importance.

Value Object

In the PeopleSoft Enterprise Warehouse, Value Objects are a metadata layer that provides descriptive information about fields and values. Value Objects are used as constants in Data Manager target object definition.

Variable

Temporary storage for use or defined information used in the creation and application of rules (see Time Administration)

Variable [Global Payroll]

An element type that defines and stores values such as a character, date, or number. You can use variables to create generic formulas for situations where you use the same values over and over again in a calculation.

Variable Compensation

In PeopleSoft Workforce Analytics, direct compensation that is not fixed, that is paid out in variable amounts, such as bonuses and commissions.

Variable Plan.

A plan in which either the number of shares and/or the price at which they will be issued is not known on the grant date.

VAT Account Type

A code that identifies the different types of accounting entries that must be created for VAT transactions. These codes are also used to categorize transactions in the VAT transaction table. The account type is used in conjunction with the VAT code and VAT transaction type to determine the VAT ChartFields used for a given VAT accounting entry.

VAT Apportionment

For mixed activity, VAT apportionment is the mechanism that allows you to specify the ratio of taxable activity to exempt activity for individual ChartFields.

VAT Calculation Method

Options are Net or Gross. When calculating VAT at net, the early payment discount is applied to the goods amount before calculating the VAT. The amount of VAT calculated using this method is the amount that is to be paid, regardless of whether the early payment discount is actually taken at time of payment. When calculating VAT at gross, the VAT is initially calculated based on the gross transaction amount. The early payment discount is not taken

into account at this point. However, in some countries an adjustment is made to the VAT amount at the time of payment, if the early payment discount is taken.

VAT Calculation Type

Options include Exclusive or Inclusive. If exclusive, the VAT amount is stated separately from the goods amount. If inclusive, the VAT is not stated separately but is included with the goods amount.

VAT Code

The tax code used to define a percentage the system uses to determine the VAT amount. The VAT code is similar to the sales and use tax code, with a few exceptions. The tax authority tied to the VAT code generally consists of a single authority, and the ChartFields for a VAT code don't reside with the tax authority but are determined by the combination of the VAT code, VAT account type, and VAT transaction type.

VAT Declaration Point

When VAT transaction information is declarable for reporting purposes. Options include Invoice or Payment. If you choose invoice, the system will recognize VAT at invoice time; if you choose payment, the system will recognize VAT at the time of payment.

VAT Entity

The level or entity within an organization at which VAT reporting is performed. VAT entities can be registered for VAT in multiple countries, but only one country can be designated as the VAT entity's home country. VAT and Intrastat reporting information and VAT default information are defined for each country in which the VAT entity is registered. You may also specify any VAT exceptions—either exoneration or suspension from paying VAT—for any country in which the entity is registered.

VAT Exempt Supply or Purchase

A transaction where the product or item is non-taxable or exempt from VAT. No VAT code is associated with the transaction. Although no tax is applied to the transaction, the transaction is still logged in the VAT transaction table.

VAT Exonerated

A transaction where the purchaser has been determined as not subject to VAT. For these cases, there may be an exoneration certificate number tied to the purchasing entity (either the customer or the VAT entity) as proof of exoneration. A zero-rated VAT code should be associated with transactions where exoneration applies. The transaction is still logged in the VAT transaction table, but no tax is applied.

VAT Rebate Percent

Within Canada, for Public Service Bodies, the percentage of VAT that is not normally recoverable but which may be refunded in the form of a tax rebate.

VAT Recoverability Percent

The percent of VAT that's recoverable.

VAT Registration Countries

Country codes associated with a VAT registration number for a particular customer or VAT entity.

VAT Transaction Table

Stores detailed transaction information for VAT reporting. It is the primary source of information for all VAT reports. Each application is responsible for writing to this table and also to a cross-reference table used to link entries in the VAT transaction table with entries within each application.

VAT Transaction Type

Used to categorize VAT transactions according to particular VAT accounting and reporting requirements. The VAT code and the VAT transaction type are used in conjunction with the VAT account type to obtain the ChartFields for accounting entries.

VAT Treatment

A description of how the transaction must be treated for VAT purposes. This is used to determine how VAT defaults are applied, what accounting entries are required, and how and if the transaction is reported on the VAT return.

VAT Use ID

A code used to identify the type of activity in which a purchased good or service will be used, and therefore to determine a recoverability percent and a rebate percent (when applicable) that will be applied to a transaction line. Activities are categorized as taxable, exempt, or mixed. Where activity is mixed, you may associate either the ratio of taxable activity to exempt activity directly with the Use ID, or you may indicate that this ratio is determined at the ChartField level.

VdkVgwKey

A key within a Verity BIF file for every document to be indexed. VdkVgwKey values must be unique across all collections that will be searched in any one application.

Vendor Draft

A draft issued by a vendor. PeopleSoft Receivables generates vendor drafts, provides a flexible worksheet environment for approval management, and enables discounted or standard submission for bank processing. PeopleSoft Payables receives vendor drafts and associates the appropriate vouchers.

Verity

The third-party search engine integrated with the PeopleSoft Portal.

Verity Fields

Verity fields are stored in the collection for retrieval and searching, and can be returned on a results list. Fields are defined in the BIF file and stored in the collection for retrieval and searching, and can be returned on a results list. Fields, like date and numeric fields can be used with the comparison operators (<,<=,>,>=).

Verity Thesaurus

The custom thesaurus consists of lists of synonyms defined in a synonym control file and can be used for synonym searching. After defining synonym lists in the control file, you use the mk Syd utility to create a custom thesaurus (a control file which has the .syd extension) that the search engine uses.

Verity Topics

Verity applications can provide end users with predefined search criteria called *topics*. A topic is a named object that represents a concept, or subject area and can be used for synonym searching. It consists of words and phrases grouped together using the Verity query language in a tree-like structure. When provided, topics can be shared by all users.

Verity Zones

Zones are specific regions of a document to which searches can be limited. When the zone filter is used, the Verity engine builds zone information into the collection's full-word index. The index, enhanced with zone information, permits quick and efficient searches over zones. Searching a zone is faster than field searching. Zones are defined in the DAT file. The contents of a zone cannot be returned in the results list of an application.

Version

There can be up to five budget versions for each Budget Center level in a Budgeting Model. Budget versions are used to perform what-if analysis and comparisons of budget amounts before the user selects one version to submit as the Budget Center's budget plan. PeopleSoft Budgeting-specific.

Vest Deferral Grace Period

The specified period of time within which an optionee must return from leave to avoid having the vesting deferred. Only applicable if the Stock Action is LOA. Suspend Vesting must be selected for this rule to be applicable.

Vest Deferral Grace Period Service Rule

If the company provides a vest deferral grace period, they may stipulate that only certain individuals are eligible for the grace period based on service with the company. Only applicable if the Stock Action is LOA. Suspend Vesting must be selected for this rule to be applicable.

Vest Immediately

A stock option plan may provide that upon specific types of terminations, or upon a case by case scenario, all unvested shares held by an individual can be made immediately vested as of a specific date. Some companies' plans provide that under certain circumstances, such as retirement, the vesting of option shares accelerates upon termination of employment. When this occurs, you must modify the vesting schedule before you terminate the individual.

Vested Shares

Option shares that are free of any ownership restriction. Generally, vested exercised shares are fully owned by the optionee, free from restrictions and freely tradable.

Vested Termination

The termination of an employee who has a vested benefit. The benefit is deferred until the participant reaches retirement age. The employee is considered "Terminated Vested," "Term Vested," or simply "TV."

Vesting

The method by which a granted option becomes free of all restrictions and the Optionee has full rights to the shares.

Vesting Schedule (Template)

A convenient way to set up the framework for a vesting schedule that can be uniformly applied to individual options. When you grant stock options, you define a vesting schedule to determine the default-vesting schedule for the option.

Vesting Service

The service used to determine an employee's vesting percentage. Rules for accruing vesting service may be different from rules for accruing other plan service credits.

VETS-100 Federal Contractor Report

This report is required of employers in the United States. It lists federal job classifications, and the number of employees and new hires in the last 12 months who are special disabled military veterans or Vietnam era military veterans. It also provides totals for each job classification of both veterans and non-veterans who hold these jobs.

View

PeopleSoft Demand Planning, a multilevel forecast structure. Each view is associated with a unique view ID and includes information that defines the view and structure type. The three types of views are working, disbursement, and dynamic.

For PeopleSoft Budgeting see Budget View.

Virtual Tasks

In Time and Labor, Virtual task data is associated with a taskgroup profile that defines common characteristics for a given Taskgroup and Task Profile ID. A single row of data is linked to multiple Earnings records for multiple employees. By minimizing the physical storage of daily task data we provide enhanced performance without limiting its functionality.

Vision

In PeopleSoft Balanced Scorecard, the overall mission of an organization. Usually the highest level on a strategy tree. Vision is optional; you aren't required to have a vision component on each strategy tree.

See also Strategy Tree

Volume

Total share volume traded in a stock during market hours.

W**WA (Workforce Analytics)**

See PeopleSoft Workforce Analytics

Waiver Of An OPM Qualification Standard

Involves setting aside requirements in a published standard to place an employee in a particular position, usually to avoid some kind of hardship to the employee, such as in cases of RIF or administrative error on part of the agency. Extra training and/or skills development may be needed to help the employee adjust to the new position. Waivers are granted by OPM or an agency, as appropriate, on a case-by-case basis, and do not directly affect other positions in the organization.

Warehouses

A warehouse reporting and analysis solution that supports the specific PeopleSoft business application that warehouse is using. It consists of predefined ETL maps, data warehouse tools, and Data Mart definitions. The warehouses we deliver are: PeopleSoft Financials Warehouse, PeopleSoft HRMS Warehouse, PeopleSoft CRM Warehouse, and PeopleSoft Supply Chain Warehouse.

Warning Exception

A transaction that exceeds the available funds but is allowed to continue to be posted against the budget. Warnings are informational only.

Warrant

A type of security, usually issued together with a bond or preferred stock, that entitles the holder to buy a proportionate amount of common stock at a specified price, usually higher than the market price at the time of issuance, for a period of years or to perpetuity. A warrant is usually issued as a sweetener, to enhance the marketability of the accompanying fixed income securities. Warrants are freely transferable and are traded on the major exchanges.

WCB

In Canadian provinces the Worker's Compensation Board (WCB) operates as an independent board, and thus would have different requirements in each province. For example, in British Columbia the organization is called the Worker's Compensation Board of British Columbia and in the Province of Quebec, the board is known as Commission de la Santé et de la Sécurité du Travail (CSST).

Weight

In PeopleSoft Planning, a user-defined value for the constraints that can be violated, determining how the schedules score will be calculated. Violations that are more critical to your schedule merit a higher weight.

Weight and Volume Pricing

You can price shipments by weight or volume to create price prices. Weight and Volume pricing requires using estimated shipments.

Weighted Average Cost of Funds

The projected principle payments for an instrument are used to derive a series of matched maturity funding rates, which in turn are used to calculate the overall base PeopleSoft Funds Transfer Pricing (FTP) rate. The Weighted Average Cost of Funds (WACF) method calculates a weighted average FTP rate where each of the funding rates is weighted by the principle payment amount and the term to maturity of the payment.

WFA (Workforce Analytics)

See PeopleSoft Workforce Analytics

WGI Due Date

Identifies the date of an employee's next within grade increase. Current policy is that the step increase is implemented on this date automatically unless prevented by the processing of an unsatisfactory performance appraisal.

WGI Non-Creditable Days

Total number of days that cause the WGI due date to be adjusted forward.

Whole Calendar Month

An instruction telling the system to use every day in each month for this time period. The system fills in the last day of the period according to the information you have entered.

Wildcard

You can replace the right-hand characters in a search field with a percent (%) wild card to query a range of values beginning with the remaining, left-hand characters. For example, by entering '2%' in a six-character field, you will receive a range of available values, such as 200000 through 299999 or 2aaaaa through 2zzzzz.

Window Period

The ten-day period, from the third to twelfth day after public release of a company's financial statement, when insiders may exercise their stock-appreciation rights without violating Securities and Exchange Commission rules for short-term trading.

Windows Client

Traditional PeopleSoft 32-bit client. Windows clients connect to the application server domain (Tuxedo) using a port number (or connection string) specified in PeopleSoft Configuration Manager.

WIP Replenishment Method

Designates how the PeopleSoft Flow Production request is communicated. For a replenishment method of Inventory, the Workflow, Pull Ticket, and Pull List replenishment methods are available. With feeder line replenishment, you can only use Pull Tickets.

WIP Replenishment Mode

Determines how PeopleSoft Flow Production is triggered to generate a replenishment request for an item. Replenishment options include Backflush, Manual, and Kanban Card.

WIP Replenishment Source

Determines where you send your PeopleSoft Flow Production replenishment request and what source supplies your WIP location. Options include Feeder, Inventory, and Vendor.

Withdrawal

An election not to continue participation in a stock purchase plan.

Withdrawal of Contributions

In a pension plan, the act of returning pension contributions, with interest, to an employee who is terminating. An employee who withdraws contributions typically forfeits all service associated with those contributions. If the employee is later rehired, repayment of contributions and interest typically reinstates the forfeited service.

Withholding

A deduction taken by employers out of taxable income of an individual. Typical withholding taxes include federal income taxes, federal social security, Medicare taxes, and state and local income taxes.

Within Grade Increase (WGI)

A longevity-based increase in salary based on predetermined time in grade requirements and acceptable performance.

Work Council (Comité d'Enterprise)

In France it is mandatory for companies with more than 50 employees to elect a Work Council to represent the employees in negotiations with management.

Work Effort

See Activity Type.

Worker

In PeopleSoft Workforce Analytics, workers are defined as anyone who performs functions for the organization, and receives compensation from the organization's operating expense funds in return. Workers can be direct employees or independent contractors. This includes individuals contracting business directly from the company or through an agency.

Work Group

In PeopleSoft Enterprise Performance Management, the work group is a grouping of employees that share a similar activity profile.

Work Period

A Days On/Days Off template; the smallest unit of time that a business uses to communicate with their employees regarding when to be and/or not to be at work (that is, time working and time not working). The work period can be any number of hours. Until clock hour reporting is implemented, the application does not care about the number of hours. The initial Time and Labor product will apply the work period to a calendar day.

Work Queue

In PeopleSoft Demand Planning and Inventory Planning, a feature for reviewing and working with exceptions created during the processing of forecasting and inventory data.

Work Schedule

A template consisting of a sequence of work periods (days) on and off, and the number of scheduled hours per work period. Work Schedules and Work Periods should not be confused with calendar days.

Worksheet

A way of presenting data to the user through a BAM interface that enables users to do in-depth analysis using pivoting tables, charts, notes, and history information.

Work Templates

Work templates describe your employee's work patterns. Work templates could apply to individuals or entire organizations. For instance, 9 AM to 5 PM, Monday through Friday is a fairly standard working week in organizations.

Workday

A 24-hour period rounded by daybreaker with one or more associated shifts (see Scheduling)

Workday Override

A function that allows a Time Manager to override a Time Reporter's schedule for a single workday. For example, Jane's long-term schedule assignment is Monday – Friday, 8.00 to 17.00. Due to an increase in production demand, her manager needs to schedule her to work 7.00 to 18.00 on Thursday, 16 March 2000. Her manager needs to be able to make this change to her schedule in the PeopleSoft Time and Labor system, so when Jane checks her schedule for this week, she'll see the revised schedule.

Worker

Workers can be defined as anyone who performs functions for the organization and receives compensation from the organization's operating expense funds in return. Workers can be direct employees or independent contractors (includes individuals contracting business directly from the company or through an agency).

Workers Compensation

The days an employee is on LWOP due to sustaining an injury or illness while on the job.

Workflow

The background process that creates a list of administrative actions based on your selection criteria and specifies the procedure associated with each action.

Workflow

The background process that creates a list of administrative actions based on your selection criteria and specifies the procedure associated with each action.

Workforce Monthly Report (Déclaration Mensuelle Obligatoire des Mouvements de Main D'oeuvre)

In France, companies that employ 50 or more employees are required to submit the Workforce Monthly Report to the Administrative Division of the Ministry of Work and Social Relations. The report contains workforce information for a given establishment of a company, including

the total number of employees and details of employees who have joined or left the establishment during the month.

Workgroup

A user-defined group of employees who share identical compensation rules. A workgroup may be equivalent to all the employees in a business enterprise, all employees in a Paygroup, all employees belonging to the same Union or Union Local, or all employees who work at a specific work location.

Worklist

The automated "to do" list that Workflow creates. From the Worklist you can directly access the panels you need to perform the next action, and then return to the Worklist for another item.

Worklist

The automated "to do" list which 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.

Works Councils (Betriebsrat)

In Germany, the works councils for your company are internal committees elected by the employees that represent the interests of salaried and hourly paid employees, other than management. Every work location in your company has its works council (this would be the local works council) and the company as a whole has a central works council.

Work-Study Program

Government or non-government programs supervised work experience related to a student's course of study and are a part of, or a supplement to, education. Federal student-trainee programs are examples of such programs.

X

Y

Yearly Maximum Pensionable Earnings (YMPE)

Amount set by the government upon which Canadian Pension Plan (CPP) contributions are made.

Z

Zero-Based Budgeting

A budgeting option that builds a budget from the ground up starting with zero values. This is in contrast to an incremental budget that is based upon using prior year actual or budget values as starting point. PeopleSoft Budgeting-specific.

Zero-Rated VAT

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.

Zip Code

The term for postal codes in the United States.

Index

@

@ operator 6-30

A

access classes 10-1

accessing

- component PeopleCode 2-11
- component record field PeopleCode 2-8
- component record PeopleCode 2-9
- field object 10-13
- level 0 10-6
- menu item PeopleCode 2-13
- message PeopleCode 2-15
- page PeopleCode 2-12
- panel group record field PeopleCode 2-9
- PeopleCode Debugger 13-1
- PeopleCode programs 11-3
- record field PeopleCode 2-4
- record object 10-11
- row object 10-10
- rowset object 10-6
- SQL editor 4-3

Activate event 11-34

ambiguity in field references 9-7

Application Packages

editing 5-5

Application Packages editor

considerations 5-4

understanding 5-1

application server

DLL functions 12-2

assigning

objects 7-5

attachment functions

coding considerations 8-45

filenames for PutAttachment 8-46

naming considerations 8-46

overview 8-36

subrecord 8-43

using 8-43

work record 8-43

B

backing up PeopleCode programs 2-18

Boolean constants 6-15

Boolean operators 6-31

buffer fields

referring to 9-15

Business Interlink class

generating PeopleCode 3-14

C

class

definition of 7-1

comments 6-5

comparing rowsets to scrolls 9-3

comparison operators 6-30

compile all PeopleCode programs 13-15

component buffer

contents 9-3

contextual buffer field references 9-7

contextual row references 9-6

current context 9-5

processing order 9-6

record fields 9-3

referring to fields 9-15

resolving ambiguity 9-7

scroll path syntax 9-9

structure 9-1

component build processing

add mode 11-21

update mode 11-18

Component Interface class

generating PeopleCode 3-15

component processor 11-1

behavior 11-7

events outside flow 11-1

component processor events 11-33

Activate event 11-34

event order 11-4

FieldChange event 11-34

FieldDefault event 11-35

FieldEdit event 11-36

FieldFormula event 11-36

ItemSelected event 11-37

PostBuild event 11-37

PreBuild event 11-37

PrePopup event 11-38

RowDelete event 11-38

RowInit event 11-39

RowInsert event 11-40

RowSelect event 11-41

SaveEdit events 11-42

SavePostChange event 11-43

SavePreChange events 11-43

- SearchInit event 11-44
- SearchSave event 11-45
- Workflow event 11-45
- concatenating strings 6-29
- conditional statements 6-8
- constants
 - Boolean 6-15
 - null 6-16
 - numeric 6-15
 - string 6-15
 - user-defined 6-16
- contextual buffer field references 9-7
- contextual references
 - data buffer access classes 10-21
 - resolving ambiguity 9-7
- contextual row references 9-6
- control statements 6-8
- copying PeopleCode programs 2-18
- cross reference reports 13-23
- current context of component buffer 9-5
- CurrentRowNumber
 - using 9-17

D

- data buffer classes
 - current context 10-21
 - example 10-3
 - traversing hierarchy example 10-14
- data buffer model 10-2
- data types 6-1
 - conventional 6-2
 - object 6-2
- date and time operators 6-29
- debugging
 - subscription PeopleCode 13-14
- default processing
 - field-level 11-12
 - panel group level 11-12
- deferred processing mode 11-30
- DLL functions
 - application server 12-2
- dot notation
 - syntax 9-9

E

- errors and warnings
 - restrictions 8-59
- Evaluate
 - more examples 6-8
- Exec function
 - OLE automation 8-48
- expressions 6-14

F

- field modification processing 11-22
- FieldChange event 11-34
- FieldDefault event 11-35
- FieldEdit event 11-36
- FieldFormula event 11-36
- fields
 - converting strings to field references 6-30
 - naming 6-26
 - reference syntax 6-25
- file attachment
 - chunking considerations 8-41
 - coding considerations 8-45
 - configuring multiple application servers 8-39
 - debugging 8-37
 - naming considerations 8-46
 - overview 8-36
 - putattachment filenames 8-46
 - using non-DNS URLs 8-43
 - using NT4 ftp server 8-42
 - using the attachment functions 8-43
 - viewing considerations 8-39
- file class
 - generating PeopleCode 3-17
- Find In tool 13-20
- fonts
 - PeopleCode editor 3-13
- formatting
 - PeopleCode 3-10
- functions
 - calling 6-12
 - declaring 6-12
 - defining 6-12
 - naming conflicts 6-13
 - parameters 6-12
 - recursive 6-21
 - return values 6-13

G

- GenerateTree function
 - building HTML tree page 8-21
 - customizing PeopleCode 8-25
 - end-user actions 8-24
 - FieldChange example 8-30
 - HTML tree 8-19
 - mouse-over ability 8-34
 - PostBuild example 8-27
 - rowset records 8-22
 - using 8-19
 - visual selection node indicators 8-35

H

HTML
tree 8-19

I

ImageReference field 8-13
implementing
modal transfers 8-9
multi-row insert 8-12
inserting rows using PeopleCode 8-18
ItemSelected
event 11-37
processing 11-28

L

language constructs 6-7
assignment statements 6-6
comments 6-5
legal names
record fields 6-25
local variables
considerations 6-18
lifetime 6-18
logical operators 6-31
loops
For 6-10
Repeat 6-10
scroll levels 9-17
While 6-10

M

math operators 6-28
message definitions
accessing PeopleCode 2-15
Messages
debugging 13-14
Metastrings 6-24
modal transfers
considerations 8-10
implementing 8-9
multiple occurs levels 11-46
multiple scroll levels
effects on PeopleCode execution 11-46
multi-row insert
considerations 8-13
implementing 8-12

N

name

references 6-26
reserved 6-27
null constant 6-16
numeric constants 6-15

O

object
assignment 7-5
definition of 7-2
methods 7-4
properties 7-3
working with 7-2
object datatypes 6-2
occurs levels
multiple 11-46
OLE automation
data types 8-48
sharing object instances 8-48
using 8-47
WinExec 8-48
operators
@ 6-30
Boolean 6-31
date and time 6-29
math 6-28
relational 6-30
string concatenation 6-29

P

passing objects 7-7
PeopleBooks
printed, ordering iii
PeopleCode
accessing 2-2
accessing external functions 3-11
compiling all programs 13-15
component processor 11-1
current context 9-5
finding strings in 13-20
how programs are triggered 11-2
inserting rows 8-18
PeopleSoft Internet Architecture applications
12-1
using drag-and-drop 3-13
PeopleCode Debugger
accessing 13-1
debugging tips 13-9
DoModal consideration 13-10
environment 13-14
features 13-2
log file interpretation 13-18
log options 13-16
messages 13-14
options 13-10

- single debugger considerations 13-5
 - subscription PeopleCode 13-14
 - variables panes 13-6
 - PeopleCode editor
 - choosing a font 3-13
 - context-sensitive help 3-12
 - definition list 3-2
 - drag-and-drop 3-11
 - event list 3-6
 - find and replace 3-9
 - formatting statements 3-10
 - functions 3-8
 - generating Business Interlink template 3-14
 - generating Component Interface template 3-15
 - generating definition references 3-14
 - generating file template 3-17
 - replace dialog box 3-9
 - using 3-7
 - validating syntax 3-9
 - window 3-1
 - PeopleCode language
 - comments 6-5
 - PeopleCode programs
 - automatic backup 2-18
 - copying 2-18
 - saving 2-17
 - PeopleSoft Internet Architecture
 - deferred processing mode 11-30
 - processing considerations 11-30
 - search page 12-2
 - using GetHTMLText 8-17
 - using GetJavaScriptURL 8-18
 - using HTML area 8-16
 - using PeopleCode 8-14
 - using style property 8-15
 - writing PeopleCode applications 12-1
 - pop-up menus
 - processing 11-27
 - PostBuild event 11-37
 - PreBuild event 11-37
 - PrePopup event 11-38
 - processing
 - deferred mode 11-30
 - processing sequences
 - component build in add mode 11-21
 - component build in update mode 11-18
 - default processing 11-11
 - field modification 11-22
 - ItemSelected 11-28
 - pop-up menu 11-27
 - pushbuttons 11-27
 - Row Delete 11-25
 - Row Insert 11-24
 - Row Select 11-20
 - Save 11-28
 - search in add mode 11-16
 - search in update mode 11-13
 - PSOPTIONS table
 - considerations 12-4
 - pushbuttons
 - processing 11-27
- ## R
- record field
 - legal names 6-25
 - recursive functions 6-21
 - relational operators 6-30
 - Remote Call 8-62
 - and Process Scheduler 8-65
 - components of 8-63
 - PeopleCode API 8-63
 - programming guidelines 8-66
 - remote program API 8-64
 - reserved words 6-27
 - restrictions on function and method use 8-1
 - CallAppEngine 8-7
 - Component Interface 8-7
 - data buffer fields 8-5
 - DoSave 8-5
 - Errors and warnings 8-5
 - GetGrid 8-8
 - GetPage 8-8
 - Publish method 8-9
 - record object 8-6
 - ReturnToServer 8-8
 - SeachInit 8-7
 - SQL object 8-6
 - think-time 8-2
 - WinMessage and MessgeBox 8-3
 - Row Delete processing 11-25
 - Row Insert processing 11-24
 - Row Select processing 11-20
 - RowDelete event 11-38
 - considerations 11-38
 - RowInit event 11-39
 - exception 11-39
 - RowInsert event 11-40
 - rows
 - referring to 9-14
 - RowSelect event 11-41
 - rowset
 - comparing to scroll 9-3
 - example 10-17
 - instantiating using non-panel buffer data 10-23
 - standalone rowsets 8-53
- ## S
- Save processing 11-28
 - SaveEdit event 11-42
 - SavePostChange event 11-43
 - SavePreChange event 11-43
 - saving PeopleCode programs 2-17

- scroll levels
 - looping through 9-17
 - multiple 11-46
 - referring to 9-13
- scroll path syntax 9-9
- search PeopleCode 13-20
- search processing
 - add mode 11-16
 - update mode 11-13
- SearchInit event 11-44
- SearchSave event 11-45
- Select method
 - child rowsets 8-51
 - select record 8-51
 - syntax 8-50
 - using 8-49
- SelectNew method
 - using 8-49
- shared objects considerations 6-23
- SQL
 - Metastrings 6-24
- SQL definition 4-3
- SQL editor
 - accessing 4-3
 - properties 4-2
 - using 4-1, 4-7
- standalone rowsets 8-53
 - adding child rowsets 8-54
 - CopyTo method 8-54
 - Fill method 8-53
 - reading from files 8-57
 - writing to files 8-55
- starting
 - third-party applications from PeopleCode 8-48
- string concatenation 6-29
- string constants 6-15
- subroutines 6-8
- syntax
 - scroll path 9-9
 - validating with PeopleCode editor 3-9
- system edits 11-22

T

- think-time functions 8-2

- time and date operators 6-29
- Tuxedo *See* Remote Call

U

- user-defined constants 6-16
- using
 - Application Packages editor 5-1
 - attachment functions 8-43
 - GenerateTree function 8-19
 - ImageReference field 8-13
 - inserting rows 8-18
 - OLE functions 8-47
 - PeopleCode editor 3-7
 - select method 8-49
 - SelectNew method 8-49
 - standalone rowsets 8-53

V

- validating syntax
 - PeopleCode editor 3-9
- variables
 - considerations 6-18
 - declaration 6-17
 - declaring 6-20
 - initializing 6-20
 - passing to functions 6-22
 - restrictions 6-21
 - scope 6-17
 - system 6-24
 - user-defined 6-22

W

- warnings and errors
 - restrictions 8-59
- WinExec function
 - OLE automation 8-48
- Workflow event 11-45