

PeopleTools 8.4:
PeopleSoft Application Engine

PeopleSoft®

PeopleTools 8.4: PeopleSoft Application Engine

SKU Tr84APE-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

PeopleSoft Application Engine Preface

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

Chapter 1

Introducing Application Engine

Understanding Application Engine	1-1
Using Meta-SQL	1-1
Application Engine Program Elements	1-2
Special Application Engine Programs	1-4
Daemon Program Type	1-4
Transform Program Type.....	1-6

Chapter 2

Creating Application Engine Programs

Viewing Application Engine Programs	2-1
Using the Definition View	2-2
Using the Program Flow View.....	2-4
Switching Between Definition and Program Flow Views	2-5
Using Refresh.....	2-6
Filtering View Contents.....	2-7
Printing a Program Definition or Flow.....	2-9
Opening or Renaming a Program	2-11
Copying or Moving Program Elements.....	2-12
Testing an Application Engine Program	2-12
Setting Program Properties	2-14
Setting General Properties	2-14

Setting State Records Properties	2-14
Setting Temp Tables Properties	2-16
Setting Advanced Properties	2-18
Adding Sections	2-19
Inserting a Section	2-19
Locating Sections	2-20
Setting Section Properties	2-21
Executing Order of Sections	2-22
Adding Steps	2-23
Inserting Steps	2-23
Setting Step Properties	2-24
Specifying Actions	2-25
SQL Actions	2-26
Do Actions	2-28
PeopleCode Actions	2-31
Call Section Actions	2-31
Log Message Actions	2-33
XSLT Actions	2-34
Executing Order of Actions	2-34
Inserting Actions	2-36
Setting Action Properties	2-36

Chapter 3

Developing Efficient Programs

State Records	3-1
Sharing State Records	3-2
Choosing a Record Type for State Records	3-3
Commit Considerations	3-4
Re-Using Statements	3-5
ReUse (SQL Action Property)	3-5
Using Bulk Insert	3-6
Set Processing	3-7
Using Set Processing Effectively	3-7
Tips for Your Application Engine Programs	3-10
Examples of Set Processing	3-12

Chapter 4

Using Meta-SQL and PeopleCode

Meta-SQL in Application Engine	4-1
%Bind	4-1

%ExecuteEdits	4-4
%Select	4-6
%SelectInit	4-7
%SQL	4-7
%Table	4-8
%TruncateTable	4-8
%UpdateStats	4-9
Application Engine Macros	4-12
Application Engine System (Meta) Variables.....	4-14
Using PeopleCode in Programs	4-16
When to Use PeopleCode	4-16
Environment Considerations.....	4-16
State Records	4-18
IF, THEN Logic	4-18
Scope of Variables	4-19
Action Execution Hierarchy.....	4-19
Using PeopleCode in Loops.....	4-20
AESession Object.....	4-21
Online Application Engine Calls (CallAppEngine).....	4-21
File Layout Object	4-24
Calling COBOL Modules (RemoteCall).....	4-25
PeopleTools APIs.....	4-27
CommitWork	4-28
Notes on Various PeopleCode Objects and Functions.....	4-28
Dynamic SQL.....	4-30

Chapter 5

Managing Application Engine Programs

Running Application Engine Programs	5-1
Using Process Scheduler	5-2
Application Engine Process Request Page.....	5-4
Using PeopleCode.....	5-6
Using the Command Line	5-7
Debugging Application Engine Programs	5-10
Enabling the Debugger.....	5-10
Debugging Options	5-12
Restarting Application Engine Programs	5-18
How Restart Works.....	5-18
When to Use Restart	5-19
Controlling Abends.....	5-21

Restarting an Application Engine Program.....	5-22
When to Restart or Start Again	5-23
Common Restart Errors.....	5-24
Enabling or Disabling Restart.....	5-24
Caching the Application Engine Server.....	5-25
Using CacheBaseDir	5-25
Managing Abends.....	5-26

Chapter 6

Calling Application Engine Programs From COBOL

Adding Copybook.....	6-1
Assigning Copybook Values	6-2

Chapter 7

Tracing Application Engine Programs

Understanding Tracing Application Engine Programs	7-1
Enabling Application Engine Traces	7-1
Locating the Trace File	7-4
Understanding the Trace Results.....	7-4
Trace File Sections	7-7
Tracing Base Temporary Table Usage.....	7-20

Chapter 8

Using Temporary Tables

Understanding Temporary Tables	8-1
Creating Temporary Table Instances.....	8-3
Defining Temporary Tables	8-3
Building Table Instances.....	8-4
Managing Temporary Table Instances	8-6
Assigning Temporary Tables to Programs	8-6
Adjusting Meta-SQL	8-9
Making External Calls	8-11
Sample Implementation.....	8-12
Temporary Table Performance Considerations	8-13
Initial Estimates.....	8-13
Online Temporary Table Allocation	8-14
Viewing Temporary Table Usage.....	8-14
Viewing Online Instance Usage.....	8-15
Viewing the Temp Table Usage Warning Message.....	8-16

Glossary

Index

PeopleSoft Application Engine Preface

PeopleSoft Application Engine is a PeopleTool designed to help you develop, test, and run background SQL processing programs. Its chapters explain the concepts and advantages of Application Engine, how to create programs using the Application Engine Designer, how to run and debug programs, and the use of the special tools to maintain your programs.

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**.

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.
Effective Date	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. <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	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. Language also refers to the language spoken by an employee, applicant, or non-employee.

Field	Definition
Process Frequency group box	Designates the appropriate frequency in the Process Frequency group box: 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 . Always executes the request every time the batch process runs. Don't Run ignores the request when the batch process runs.
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/> For more information about the Report List page, the Process List page, and the Process Scheduler, see Process Scheduler Basics in the PeopleTools documentation . <hr/>
Request ID	A request identification that represents a set of selection criteria for a report or process.
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 Application Engine

PeopleSoft Application Engine is the PeopleTool that you use to develop batch or online programs that perform high-volume, background processing against your data.

This chapter provides an overview of PeopleSoft Application Engine and discusses the following:

- Using Meta-SQL.
- Application Engine program elements.
- Special Application Engine programs.

Understanding Application Engine

PeopleSoft Application Engine comprises two distinct components—a designer where you define your batch program and the runtime where you run and monitor your program.

In PeopleSoft Application Engine, a *program* is a set of SQL statements, PeopleCode, and program control actions (that enable looping and conditional logic) defined in PeopleSoft Application Designer that performs a business process. You can use PeopleSoft Application Engine for straight, row-by-row processing, but the most efficient Application Engine programs are written to perform set-based processing.

PeopleSoft Application Engine does not generate SQL or PeopleCode—it executes the SQL and PeopleCode that you include in an Application Engine action as part of your overall program.

PeopleSoft Application Engine is designed for batch processing where you have a large (or small) amount of data that must be processed without user intervention—for example, calculating the salaries in payroll processing (not printing the checks). Another example might be converting data (money) from one currency to another.

Using Meta-SQL

You can write SQL within PeopleSoft Application Engine, or you can copy SQL statements into Application Engine from any SQL utility with few, if any, changes. This enables you to write and tune SQL statements before you try to incorporate them into an Application Engine program.

Database platforms can have many differing syntax rules, especially in regard to date, time, and other numeric calculations. Generally, you can work around syntax differences using

PeopleSoft meta-SQL, which Application Engine supports. This language is designed to replace RDBMS-specific SQL syntax with a standard syntax, called meta-strings.

In addition, PeopleSoft meta-SQL enables you to dynamically generate portions of your SQL statements. For example, if you want to join two tables on their common keys, use the following code:

```
%Join(COMMON_KEYS, PSAESECTDEFN ABC, PSAESTEPDEFN XYZ )
```

At runtime, the function would be expanded into the following:

```
ABC.AE_APPLID = XYZ.AE_APPLID
AND ABC.AE_SECTION = XYZ.AE_SECTION
AND ABC.DBTYPE = XYZ.DBTYPE
AND ABC.EFFDT = XYZ.EFFDT
```

Application Engine Program Elements

A PeopleSoft Application Engine program is made up of several key elements:

Program	Identifies the set of processes to execute a given task. A program must contain at least one section. The execution of the program always starts with the section defined as “MAIN.”
Sections	Comprises one or more steps and is equivalent to a COBOL paragraph or an SQR procedure. All Application Engine programs must contain at least one section entitled “MAIN.” A section is a set of ordered steps that gets executed as part of a program. You can call sections (and other programs) from steps within other sections.
Steps	The smallest unit of work that can be committed within a program. Although, you can use a step to execute a PeopleCode command or log a message, typically, you use a step to execute a SQL statement or to call another section. The SQL or PeopleCode that a step executes are the actions within the step. When a section gets called, its steps execute sequentially. Every program begins by executing the first step of the required section called MAIN and ends after the last step in the last section completes successfully.
Actions	There are multiple types of actions that you can specify to include within a step. Keep in mind that it is common to have multiple actions associated with a single step. The following sections briefly describe each type of action. These actions are explained in more detail when we

	<p>approach more advanced topics related to building Application Engine programs.</p>
Do Actions	<p>Do actions contain a SQL SELECT statement designed to return results on which subsequent actions depend. For instance, if a SELECT returns no rows, subsequent actions may not need to execute. A Do action is equivalent to a COBOL PERFORM statement and has similar constructs.</p> <p>The four types of Do actions are:</p> <ul style="list-style-type: none"> • Do While • Do When • Do Select • Do Until
SQL	<p>Most SQL actions are comprised of a single SQL statement. These statements can perform the following types of SQL statements:</p> <ul style="list-style-type: none"> • UPDATE • DELETE • INSERT • SELECT <p>The SQL action differs from the Do actions, which also contain SQL, in that the SQL action does not control the flow of the program.</p>
PeopleCode	<p>You can include PeopleCode in the PeopleCode action. Application Engine PeopleCode provides an excellent way to build dynamic SQL, perform simple IF/ELSE edits, set defaults, and other operations that don't require a trip to the database. It also enables you to reference and change active Application Engine state records.</p> <p>Most importantly, PeopleCode provides access to the PeopleSoft Internet Architecture integration technologies such as Application Messaging, Business Interlinks, Component Interface, and XML File Processing.</p>
Log Message	<p>A Log Message action can be used to write a message to the MESSAGE_LOG based on a particular condition in your program. This gives your program multi-language capability. The system stores the message generically as a message set, message number, and parameter values. When the end user views the messages using the Application Engine Messages panel, PeopleTools retrieves the appropriate message string from the Message Catalog based on the user's language preference.</p>

Call Section

You can also insert an action that calls another section. The “called” section can be in the same program as the calling section, or it can be in an *external* program. This enables you to “chunk” your program into more maintainable, reusable pieces. If a section already exists in one program, rather than copying it into another program, just call it.

State Record

A PeopleSoft record that must be created and maintained by the Application Engine developer. This record defines the fields a program uses to pass values from one action to another. Think of the fields of the Application Engine state record as the working storage for your Application Engine program.

An Application Engine state record can be either a physical record or a work record, and any number of state records can be associated with a program. Physical state records must be keyed by process instance.

Note. PeopleSoft Application Engine supports up to 99 levels of nested Call Section actions. For example, the first called section can call a second, which can call a third, and so on, up to 99 calls.

Special Application Engine Programs

There are five types of Application Engine programs, one of which you specify on the Program Properties dialog box for your program definition. They are listed here; the last two are described in more detail in the following section:

- Standard, which is a normal entry-point program.
- Upgrade Only, which is used in PeopleSoft Upgrade utilities.
- Import Only, which is used by PeopleSoft Import utilities.
- Daemon Only, a special type of program used as a daemon.
- Transform Only, another special program type used to support XSLT.

Daemon Program Type

PeopleSoft Application Engine has a new daemon process, called PSDAEMON, that runs continuously when the Process Scheduler is running, and is intended for recurrent jobs. It polls the system checking for certain conditions to be true for events and when true, it schedules a process to handle the event.

PSDAEMON supports limited tracing, because it runs indefinitely. Specifically, it only allows Application Engine tracing at the step and SQL levels, in addition to the standard PeopleSoft SQL and PeopleCode tracing. Other options such as Timings and DB Optimizer tracing are not supported.

You activate PSDAEMON in PeopleSoft Process Scheduler or from the command line.

Note. One PSDAEMON can run for each row in PS_SERVERDEFN. PS_SERVERDEFN.DAEMONENABLED must be set to 1.

Starting PSDAEMON From the Command Line

The command line syntax is:

```
psdaemon -CT <dbtype> -CD <dbname> -CO <userid> -CP <passwd> -R <servername>
```

- R** Use this parameter to query PS_SERVERDEFN, obtaining the Daemon Group, Sleep Time, and Recycle Count (terminate after N iterations).
- <servername>** This parameter is the key to PS_SERVERDEFN. You do not need to pass ProcessInstance (-I) or AE Program ID (-AI).

Starting a Daemon Program from Process Scheduler

Before starting a daemon Application Engine program, you must add the program to a Daemon Group page in Process Scheduler.

The screenshot shows the 'Process Scheduler' interface. On the left, a navigation tree lists various components, with 'Daemon Group' highlighted. The main content area is titled 'Daemon Group Definition' and features two tabs: 'Find an Existing Value' and 'Add a New Value'. Below the tabs, there is a label 'Daemon Procedure Group:' followed by an empty text input field and an 'Add' button. At the bottom of the main area, there are two links: 'Find an Existing Value' and 'Add a New Value'.

Daemon Group Definition



Loading Daemon Programs

See Also

PeopleTools 8.4 PeopleBook: PeopleSoft Process Scheduler, “Setting Server Options,” Setting Daemon Process Options.

Developer Notes About PSDAEMON

The AEDAEMONMGR program is a restartable program and it commits after each daemon procedure. When the PSDAEMON executes, it determines whether it must restart AEDAEMONMGR following a program abend.

If a restart is not required, PSDAEMON assigns a new Process Instance and runs AEDAEMONMGR from the beginning. Because of this design, PeopleSoft Process Scheduler does not have to determine whether PSDAEMON exited due to an error or because it had reached the Recycle Count.

AEDAEMONMGR uses the Daemon Group page value to get related daemon procedures from PS_DAEMONGROUP in order, and then it initiates each procedure. After all procedures have been executed, AEDAEMONMGR logs a sleep message and returns control to PSDAEMON. The sleep time is used only to log an informational message at the end of each cycle, for example "Sleeping for N minutes...". A message is also logged at the beginning of each cycle, so an administrator can monitor the run-time and sleep-time of a specific PSDAEMON process.

If an error occurs in AEDAEMONMGR; if the Recycle Count has been reached; or, if PSSERVERSTAT.DAEMONACTION = '1' (indicating that Process Scheduler is idle), PSDAEMON exits. Otherwise, it sleeps for the requested number of minutes, then calls AEDAEMONMGR again.

See Also

PeopleTools 8.4 PeopleBook: PeopleSoft Process Scheduler, “Setting Server Options,” Defining Daemon Groups.

Transform Program Type

These type of programs enable different systems to communicate with one another by transforming messages into appropriate formats. When you specify an Application Engine as a

Transform Only program, you must specify an Action Type of XSLT. You can use transform programs to do any of the following:

- Apply a *transformation* to a message to make its structure comply with the target system's requirements.
- Perform a *data translation* on a message so its data is represented according to the target system's conventions.
- Determine whether to pass a message through to its target, by *filtering* it based on its content.

See Also

PeopleTools 8.4 PeopleBook: PeopleSoft Integration Broker, "Applying Transformation, Translation, and Filtering" Developing Transform Programs.

CHAPTER 2

Creating Application Engine Programs

An Application Engine program includes a logically ordered set of sections, steps, and actions. An executable program must contain at least one section, called MAIN, used to identify the starting point of the program; it should contain at least one step; and each step should contain at least one action.

This chapter discusses how to:

- View Application Engine programs.
- Filter view contents.
- Print program and flow definitions.
- Copy or move definitions.
- Test an Application Engine program.
- Set program properties.
- Add sections.
- Add steps.
- Specify actions.

Viewing Application Engine Programs

PeopleSoft Application Designer enables you to display your program in the following two views:

- Definition
- Program Flow

This section also discusses how to:

- Switch between Definition and Program Flow views
- Use Refresh

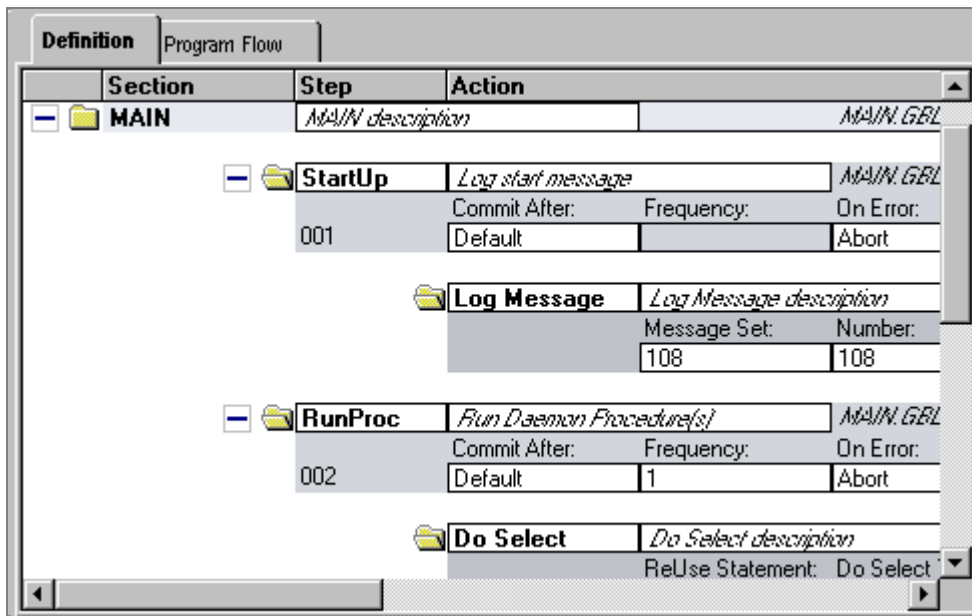
Using the Definition View

The Definition view is where you create your *definitions* within a defined hierarchical structure. Within the metaphor of the definition view, *nodes* represent the definitions. A node is the visual representation of a section, step, or action that you can select, collapse, modify, and so on.

The sections that appear in the Definition view do not necessarily appear in the order that they will execute. To see the actual order in which the sections, or the entire program for that matter, executes, switch to the Program Flow view.

Besides using the mouse, you can navigate in this view using the keyboard:

- Press CTRL+HOME to scroll to the top of the program definition and select the first node.
- Press CTRL+END to scroll to the end of the program definition and select the last visible node.
- Press TAB to move from the currently selected field to the next updateable field.
- Press CTRL+DOWN ARROW to move from the currently selected node to the next node.
- Press CTRL+UP ARROW to move from the currently selected node to the previous node.



PeopleSoft Application Designer Definition view

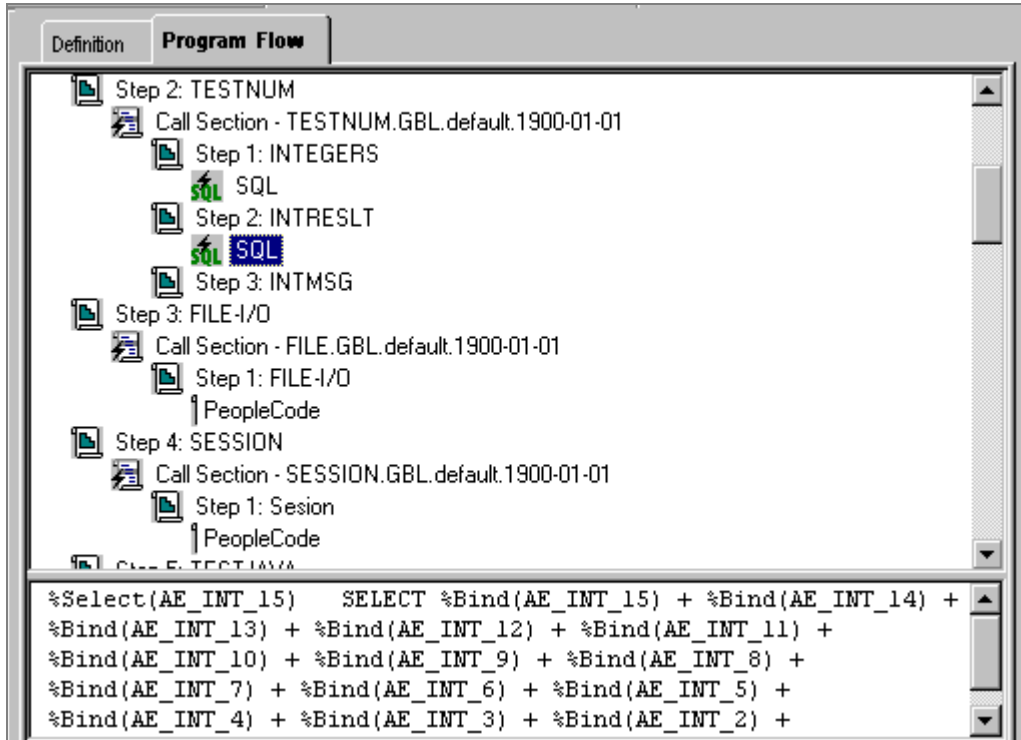
Definition View Pop-up Menu

The following table contains each menu item on the pop-up menu (right-click in the window) that you see while the definition view is active. Keep in mind that certain menu items are enabled only when a particular definition is selected.

View PeopleCode	Launches the PeopleCode Editor with the appropriate PeopleCode loaded. Enabled when a PeopleCode action is selected.
View SQL	Launches the SQL Editor with the appropriate SQL loaded. Enabled when an action containing SQL is selected.
View XSLT	Launches the SQL Editor with the related XSLT text loaded. Enabled for Transform program types only, when an XSLT action is selected.
Cut	Removes the selected item and copies it to a “clipboard”. Here, “clipboard” refers to a PeopleTools-only repository for sharing PeopleTools objects. You <i>cannot</i> copy or paste into another program.
Copy	Copies the selected item.
Paste	Pastes the contents of the “clipboard” (the most recently cut or copied item) to the current location of the cursor.
Delete	Removes the currently selected node from the program definition.
Refresh View	Refreshes the current view and reorders the Definition objects as necessary.
Show Comment	Reveals the comments associated with the selected definition object.
Insert Section	Inserts a new Application Engine section into the current program where the cursor is positioned. This option is enabled only when you have MAIN or another section selected.
Insert Step/Action	Inserts a new Application Engine step <i>and</i> action within the currently selected Application Engine section. This option is enabled only when you have a section or a step selected.
Insert Action	Inserts a new Application Engine action within the currently selected step. This option is enabled only when you have a step or action selected.
Jump to This Program Flow	Switches to the Program Flow view with the first occurrence of the currently selected definition in focus.
Print	Displays the print dialog box for the definition view.

Using the Program Flow View

The Program Flow view is a read-only view that shows the expected sequence of steps to be executed at runtime for the program you are developing.



PeopleSoft Application Designer Program Flow view

You can control the amount of detail that appears for each definition by clicking on it to expand to the next level, as needed. You can also view the SQL or PeopleCode in the lower (splitter) window area by clicking on this.

To display the pop-up menu for a node, right-click it. When using the pop-up menus, you do not have to select the node first.

You can also display the comments associated with definitions by selecting View, Show All Comments, or for a particular node, right-click on it and select Show Comment from the pop-up menu.

You can double-click on SQL or PeopleCode statements to launch the editors.

Program Flow Pop-up Menu

The following table contains each menu item on the pop-up menu in the Program Flow view.

View PeopleCode	Launches the PeopleCode Editor with the appropriate PeopleCode loaded. Enabled when a PeopleCode action is selected.
------------------------	--

View SQL	Launches the SQL Editor with the appropriate SQL loaded. Enabled when an action containing SQL is selected.
Refresh View	Refreshes the current view and reorders the Definition objects as necessary.
Show Comment	Reveals the comments for a single definition object that appears in the Program Flow view.
Jump to This Definition	Switches to the definition view with the first occurrence of the currently selected definition object in focus.
Print	Launches the print dialog for the program view.

Switching Between Definition and Program Flow Views

By default, navigation within either view does not affect the currently active row in the other view. This enables you to retain your place in one view while scrolling around in the other.

To switch between the two views, you can use any of the following methods.

View Tabs	As with any tabbed interface, if you click on the tab label, the associated view interface becomes active. When you return to the previous view, it remains positioned on the current or last selected node within the program when you switched. This is true whether you highlighted the item or just placed the cursor within an edit box.
View Menu	To switch between views using menu options, first select a section or step in the current view (note that selecting an Action does not enable this functionality—you can only jump from parent nodes). Then select View, Jump to Program Flow or View, Jump to Definition depending on the view that is currently active. When you select one of these menu options, the focus of the target view depends on what you have selected in the previous view. For example, if you have section C, step 4 highlighted in the Definition view, and you select View, Jump to Program Flow, section C, step 4 will be the focus of the Program Flow window. If the item selected resides in a program that is not already open, Application Engine opens the appropriate program, then navigates to the requested node in the view window.
Pop-Up Menus	Using the pop-up menus associated with each view, you can achieve the equivalent result as the View menu option. To activate the pop-up menu, just right-click within the desired target in either view.

Within the Program Flow View

While you are in the Program Flow view, you can select these options:

Go to Next Reference	Select this to switch to the next reference of a particular definition object if one exists. This helps you to quickly navigate through a program. For instance, if references to section C, step 4 appear three times because there are multiple calls to this object at runtime, you select Go to Next Reference to quickly and easily navigate to each reference, or call.
Jump to this Definition	Select this to go directly to the definition node in the Definition view that pertains to the current selection in the Program Flow view.

Using Refresh

As you develop an Application Engine program you may be inserting, renaming, and deleting definitions from the program. In a large program, it can be easy to lose your place or become disoriented. The Refresh option reorders all the nodes for the current definition according to the following logic:

- For standard program definitions, the MAIN section is always displayed first (Library program types do not contain a MAIN section because they contain only callable sections). The remaining sections appear alphabetically by name, which makes it easier to locate a given section within the program definition. Keep in mind that the system, at runtime, executes sections through Call Section actions within steps, not by the order in which they are defined.
- Steps are never automatically reordered in the definition view, and, at runtime, they execute in the sequence in which you define them.
- Actions are always logically reordered, based on their action type, which defines their runtime sequence.

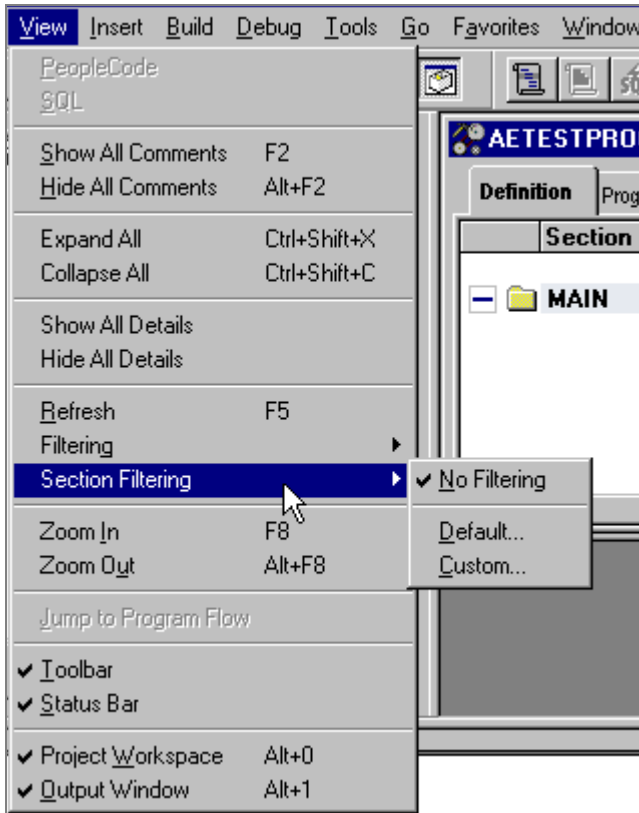
Note. When you save a modified definition, the system automatically refreshes the view.

PeopleSoft Application Engine inserts any delete requests for a given section into the current project, regardless of the Tools, Options setting in Application Designer.

For example, suppose you delete a section node from the current Application Engine program, and then you re-insert a section node and rename it to the same name as the section you just deleted. In this case, the section object *will not be* inserted into the project regardless of your Tools, Options setting. This is because a delete action already exists for this object. To resolve this situation either manually remove the delete request before inserting the new copy request or manually reset the proper flags in the upgrade project that changes the action type from delete to copy.

Filtering View Contents

Section filtering options enable you to filter the current view so that you see only sections and steps based on specified criteria.



View, Section Filtering options

To enable or modify the filtering options, select View, Section Filtering. The dialog box shows the following filtering options:

No Filtering

Select this option if you want to see all objects in your program regardless of any section attributes, such as Market, Database Type, Effective Date, and Effective Status.

Default

Displays the definition filter according to the Default filtering criteria. If you change the value of any filter option and click OK, you have defined a Custom filter.

Custom

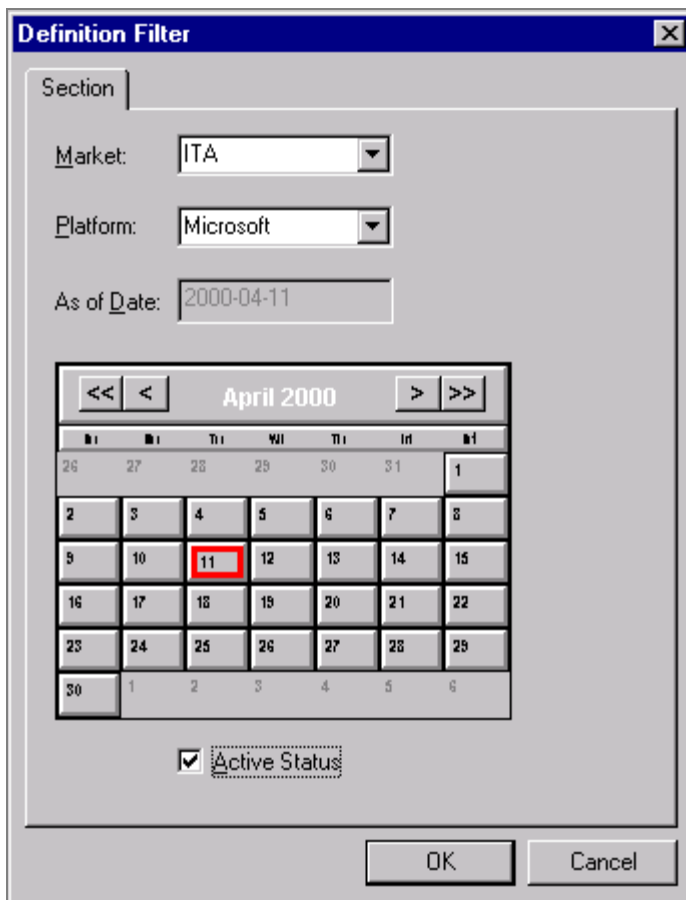
Displays the definition filter dialog box so that you can define custom filtering options for the current view.

Behavior of Section Filtering Options

- The default is No Filtering, therefore *all* section definitions are included in this view.

- If you select Custom filtering, the default filtering options are displayed while you're in that session of designer/ not persistent (or filtering options saved from a previous session. If you modify these filtering options and click OK, the new options are stored as the currently active options and the view is updated accordingly.
- If you select the Default filter option, the *original* default options display in the dialog box. After clicking OK, the view redisplay with only those sections that qualify. However, if you change the default options and do not click OK, these options are stored as a Custom filtering request and the view redisplay as necessary.
- If no platform-specific Section is defined for the target filter value, the default (base platform) is always included because this more accurately represents the PeopleSoft Application Engine runtime behavior.

If you select Section Filtering, Default, or Section Filtering, Custom, the following dialog box appears:



The image shows a 'Definition Filter' dialog box with the following fields and controls:

- Section** (tabbed area)
- Market:** ITA (dropdown menu)
- Platform:** Microsoft (dropdown menu)
- As of Date:** 2000-04-11 (text input)
- Calendar:** A calendar for April 2000 with the date 11 highlighted in red.
- Active Status:** A checked checkbox.
- Buttons:** OK and Cancel.

Definition Filter dialog box

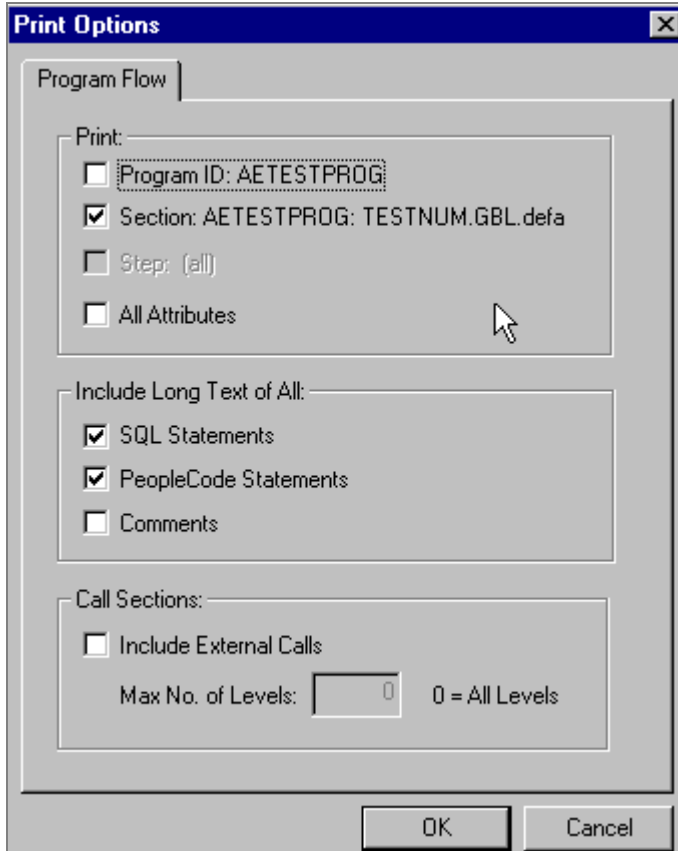
In this example, only definitions that represent the following criteria appear in the definition and program flow views.

Market	Select a market code to see only those definitions within that market. To see all market-related definitions for a program, you could update the default profile, or define a custom Filter, selecting (<i>none</i>) from the Market drop-down list. In the previous example, sections pertaining only to the Italian market (Market code “ITA”) are shown.
Platform	Select the platform filtering. In the previous example, sections that are defined only for the Microsoft SQL Server platform are shown. <i>Default</i> displays those sections defined to be database platform independent (default platform). Specific platforms: DB2/MVS, DB2/Unix, Informix, Oracle, Microsoft, Sybase.
As of Date	Select the date filtering. In the previous example, sections with an As of Date equal to or greater than April 7 th , 2000 are shown. <i>None</i> displays all sections regardless of their effective date.
Active Status	Select this check box to show active section definitions.

Note. All filtering options pertain only to section-level nodes.

Printing a Program Definition or Flow

You can print the program definition or program flow depending on which view you are in when you select print.



Print Options dialog box for program flow

To print an Application Engine program definition or flow:

1. Select Print from the pop-up menu in the Program Flow or Program Definition view.
2. Select the options that you want printed.

Program ID	Select this check box to print the whole program.
Section	Select this check box to print the currently selected section.
Step	Select this check box to print the currently selected step (all steps are printed if Section is selected).
All Attributes	Select this check box to print all detail level attributes for the specified node (and its children).
SQL Statements	Select this check box to print, for every SQL type action, the text of each SQL statement.
PeopleCode Statements	If selected, prints the text of the PeopleCode statements for every PeopleCode action.
Comments	If selected, prints the long description comments for the selected node (and its children).

Include External Calls	If selected, prints the section detail of all external calls.
Max No. of Levels	Specify the maximum number of recursive levels to print for the specified call sections.

Opening or Renaming a Program

This section describes basic tasks such as opening, copying, and deleting programs.

To create a new program definition:

1. Select File, New, or press CTRL + N.
2. On the New dialog box, select *App Engine Program*, and click OK.
3. Specify the appropriate values in the Program Properties dialog box, and click OK.
4. Save and name your program.
5. Enter the name of your program in the Save Name As edit box, and click OK to return to the Definition view.

To open an existing program:

1. Select File, Open.
2. On the Open Definition dialog box, select *App Engine Program* from the Definition Type drop-down list box.
3. Enter the new program name in the Save Name As edit box on the Save As dialog box, and click OK.

To rename a program:

1. Select File, Rename.
2. On the Rename dialog box, make sure *App Engine Program* appears as the Definition Type.
3. In the box that contains the results of your search, click on the program that you want to rename.
4. Click Rename.
5. Place the cursor in the box that appears around the highlighted program name.
6. Enter the new name for the program.
7. Click Rename again, and respond appropriately on the Confirm Rename dialog box.

Note. All static references in other programs to the renamed program, the system modifies automatically. For instance, if you call the renamed program from another Application Engine program, the Call Section action in the calling program is modified to reflect the new program name. All the sections and steps are saved under the new name. Only one occurrence of a program name can exist for a given database.

If the renamed program is called in a dynamic Do, the reference is *not* automatically modified. You should also manually check and modify any embedded references to the new program name in CallAppEngine or other PeopleCode functions.

Copying or Moving Program Elements

The following procedures apply to sections, steps, and actions. Note that when these functions are performed for a given object, the result applies not only to the selected object, but also includes its defined children, if they exist. Also note that all references to menu items apply not only to the main menu bar items, but also to their related items in the context menu, where applicable.

To copy a definition:

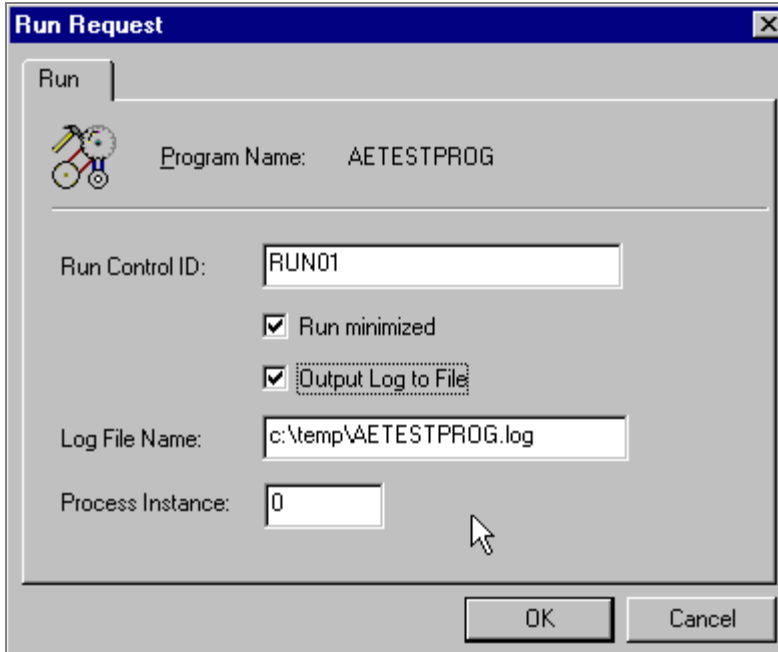
1. Select the definition, or right-click to open the pop-up menu.
2. Select Edit, Copy from the drop-down menu.
3. Position the cursor where you would like to put the copied definition select Edit, Paste from the drop-down menu.

To move a definition:

1. Select the definition object, or right-click the object to open the pop-up menu.
2. Select Edit, Cut from the drop-down menu.
3. Position the cursor in the target location within the program and select Edit, Paste from the drop-down menu.

Testing an Application Engine Program

After creating or modifying your program, you can test it while in PeopleSoft Application Designer in 2-tier mode.



Run Request dialog box

To run an Application Engine program in 2-tier:

1. Select Edit, Run Program from the PeopleSoft Application Designer toolbar.

The Run Request dialog box appears.

2. Enter the desired values. These values are passed runtime parameters to the initiated PSAE runtime executable when the OK button is clicked.

Run Control ID	Enter the run control ID of the program that you are testing.
Run Minimized	Select this to have the window of the requested process minimized when it is submitted to run.
Output Log to File	Select this if you want the output log to be written to a file.
Log File Name	Specify the log file name (enabled only when Output Log to File is selected).
Process Instance	Specify the process instance for this run request, or let default to zero if instance number is not needed.

3. Click **OK** to submit the run request.

Setting Program Properties

When you have an Application Engine program open in PeopleSoft Application Designer you can view and modify the properties assigned to an entire program just as you would a step or a section.

To view or modify the properties associated with a program, click the **Properties** button or select File, Definition Properties while the program is open. You can also press ALT-ENTER. After doing so, the Program Properties dialog box appears and includes the following four tabs:

- General
- State Records
- Temp Tables
- Advanced

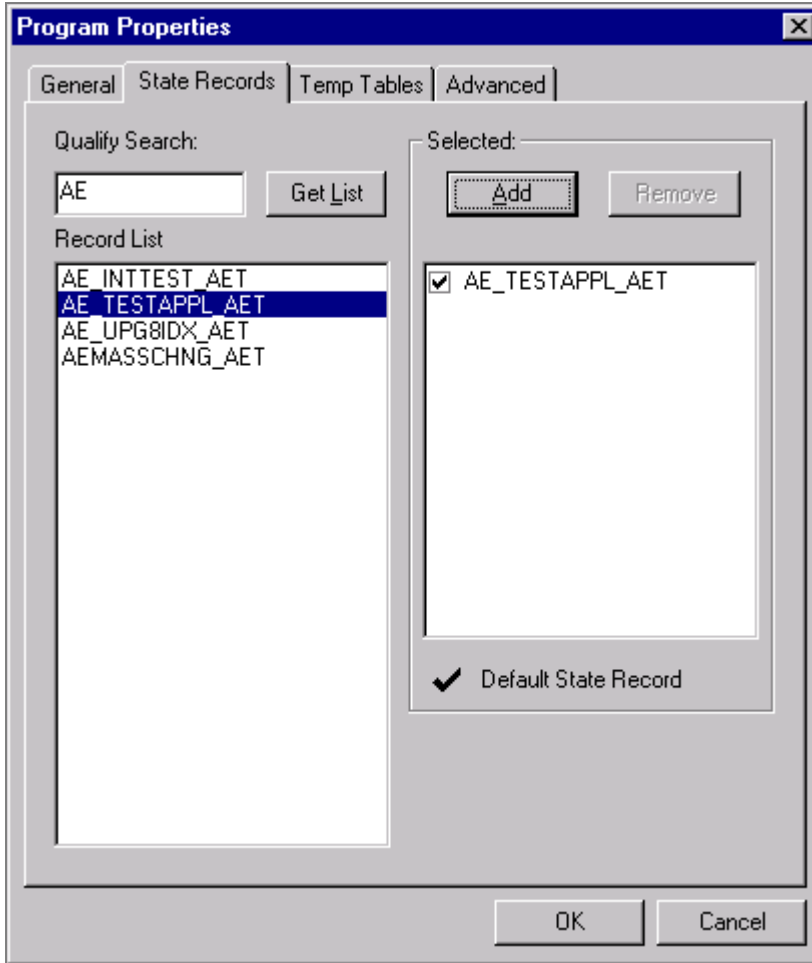
Setting General Properties

On the General tab you can specify certain identification values for your Application Engine program.

Description	Enter a descriptive name for your program. This edit box has a 30-character limit.
Comments	Describe what the program is designed to do or add any other useful comments that development or administrative staff may need to see.
Owner ID	Enter the Owner ID for the program. Owner ID is a way to identify which definitions are owned by which PeopleSoft applications, such as General Ledger, Accounts Receivables, and so on. The values in the drop-down list box are translate table values associated with the OBJECTOWNERID field. This field is optional.

Setting State Records Properties

Access the State Records tab.



Program Properties dialog box: State Records tab

The State Records properties define the following:

- | | |
|-----------------------|--|
| Qualify Search | This edit box enables you to enter any wildcard characters or complete table names to limit the results that appear in the Record List. By default, the Record List box contains all record names that end with the extension AET. This extension identifies the record within the system as an Application Engine record. |
| Get List | Click this button to populate the Record List box. To limit the length of the returned list of entries, enter any qualification text in the Qualify Search edit box prior to clicking this button. |
| Record List | This text box contains the results of your state record search. |
| Selected | This group box is where you select state records for use with a particular program. Use the Add button to include selected records from the Record List into the Selected list. Use the Remove button to remove selected records |

from the Selected list box. Indicate which state record will act as the default state record by selecting its check box. For your default state record, you need to reference only fieldnames in your PeopleCode and SQL (for the active program). When you reference a non-default state record, you do so by using *rename.fieldname*.

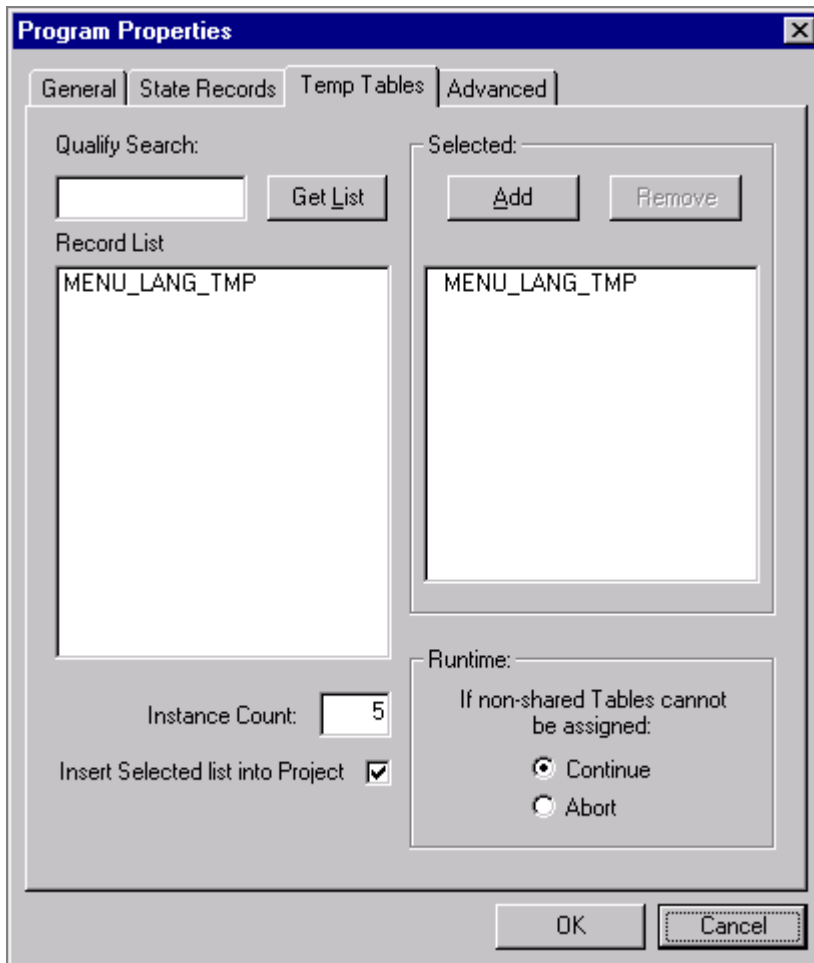
Setting Temp Tables Properties

Temporary Tables, also referred to as Temp Tables store transient or "intermediate results" during a program run.

See Also

PeopleTools 8.4 PeopleBook: PeopleSoft Application Engine, Using Temporary Tables.

Note. You must have already defined the required temporary tables in your database prior to associating them with an Application Engine program.



Program Properties dialog box: Temp Tables tab

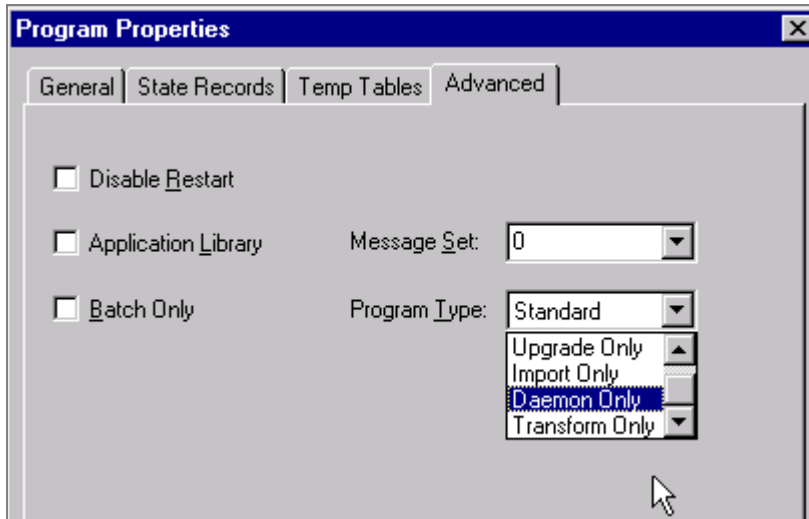
The Temp Tables properties are:

Qualify Search	This edit box enables you to enter any wildcard characters or complete table names to limit the results that appear in the Record List. By default, the Record List box contains only records that are of type "Temporary Table". This is an attribute applied at the time you create the Record in PeopleSoft Application Designer.
Get List	Click this button to populate the Record List box. Enter any qualification text in the Qualify Search edit box to limit the search prior to clicking this button
Record List	This text box contains the results of your Temp Table search.
Selected	This group box is where you select temporary tables for use with a particular program. Use the Add button to include selected records that appear in the Record list. Use the Remove button to exclude selected records that appear in the Selected list box.
Instance Count	The Instance Count value controls the number of physical tables to be created for each dedicated table for this program during the SQL Build procedure in PeopleSoft Application Designer. Typically, you would set this number to equal the maximum number of parallel program runs that you anticipate. For instance if you expect up to five instances of the same program to run simultaneously, then you would set the Instance Count to 5.
Insert Selected List into Project	If the active Application Engine program definition belongs to a Project, then you can opt to include the dedicated temporary tables for this program within the same project.
Runtime	The Runtime options enable you to control how an Application Engine program behaves in the event that an instance of its specified dedicated temporary tables are not available. If you select Continue , then Application Engine uses the base version, or undedicated version, of the temporary tables. If you select Abort , then the program exits with an error message.

Note. If the table is keyed by PROCESS_INSTANCE, and the application SQL includes the Process Instance in the WHERE clause, then the table can be shared by multiple processes. The best performance, however, occurs when a program runs against a dedicated temporary table instance.

Setting Advanced Properties

Access the Program Properties dialog box.



Program Properties dialog box: Advanced tab

The Advanced properties are:

Disable Restart

Select this to disable the Application Engine built-in restart capabilities for a particular program.

Application Library

In some cases, you may want a program to contain only a collection, or “library,” of common routines (in the form of “callable” sections) that you do not want to run as a standalone program. When sections are defined as “Public,” other programs can call the sections, or routines, that exist in the “library” at runtime. Because this type of program is not designed to run as a standalone program, it does not require the MAIN Section, or initial entry point. Setting this option, renames or removes any existing MAIN section.

Note. Application Libraries are the appropriate location to store a collection of shared Application Engine program sections. It is not intended to store a specific SQL action within a section. To share common SQL, use the SQL repository.

Batch Only

Select this check box for batch only programs. Batch only programs are not executed from the CallAppEngine() PeopleCode function. Any dedicated temporary table used for Batch Only programs do not have online instances created.

Message Set

Specify the message set value that you want assigned as the default message set number for this Application

Engine program. The system uses this Message Set value for all Log Message actions where the message set isn't specified.

Program Type

Select from these options:

Standard: Standard entry-point program.

Upgrade Only: Used by PeopleSoft Upgrade utilities only.

Import Only: Used by PeopleSoft Import utilities only

Daemon Only: Use for daemon type programs.

Transform Only: Support for XSLT Transform programs.

Adding Sections

An application engine program section comprises one or more steps and is somewhat equivalent to a COBOL paragraph or a PeopleCode function. You can create sections that are platform-independent or platform-specific, intended for a particular market, and effective-dated.

Whenever you create a new program, you simultaneously create a section, called MAIN. The MAIN section identifies the program's entry point so that it can be called by another program.

In this section, we discuss how to:

- Insert a section.
- Locate sections.
- Set section properties.
- Know the execution order of sections.

Inserting a Section

To insert a section:

1. Select Insert, Section, or select Insert Section from the pop-up menu.

The default name for a section that you insert is Section N , where N is an incremented number that attempts to provide a unique name for each section object. Unless you rename your sections according to a custom, naming scheme, the sections you add are named Section $N+1$, where n is the last section you inserted. Consequently, you get Section1, Section2, Section3, and so on.

The designer inserts the new section directly beneath all the subordinate objects within the highlighted object's owning section. For instance, in the previous example, if Section2 were selected instead of Section3, then Section4 would be inserted *between* Section2 and Section3 rather than *after* Section3.

Note. Section names are always reordered alphabetically at save time to make it easier to locate a given section by name. However, order of execution is dependent on internal Call Section references, and is therefore completely independent of the order that sections are inserted and displayed.

2. Enter the remaining section property values.
3. Save the program.

Locating Sections

The following topics describe methods you can use to locate references to sections within an entire database as well as a method for easily navigating to particular section contained within a large program.

Finding Call Section References

The following procedure describes how you generate a list of all the references to a particular section. The list applies only to Application Engine programs defined within a single database.

To locate section references:

1. Open the program containing the shared or "called" section.
2. Select Edit, Find References.
3. The Find Definition References dialog box appears.
4. On the Call Sections tab, select the appropriate section from the Section name drop-down list box, or enter the name directly.

By default, the current program name and section "MAIN" is provided in this dialog.

5. Click OK.
6. In the output window, view the generated list.

The output window lists the programs and sections that call a particular program. This list also shows the total call references made to a particular section. Call Sections within the current program appear first in the list.

Double-click an item in the output window list to automatically navigate the definition view to that calling section.

Finding Sections Within the Current Program

Within large and more complicated Application Engine programs, such as those upgraded from a previous release, it is not uncommon to have over a hundred sections. Rather than scrolling through a large program, use the Go To Section feature.

Note. This feature applies only to the current program. For example, you cannot use Go To Section to navigate to a section in another program.

To automatically navigate to a selected section:

1. Select Edit, Go To Section.

The Find Definition References tab appears.


2. On the Go To Section tab, select the appropriate section from the **Section name** drop-down list box or enter the name of the section.

3. Click **OK**.

The Definition View scrolls to the first occurrence of the section with the name you selected.

Setting Section Properties

You can find all of the controls that specify section properties in the Definition view. For example, for each Section included in your program there will be a node, as shown in the following example, from which you specify all of the attributes that you'd like to associate with a particular Section.

 MAIN	<input type="text" value="MAIN description"/>	MAIN.GBL(base).1900-01-01				
	Market:	Platform:	Effective Date:	Effective Status:	Section Type:	Auto Commit: Access:
	GBL	(base)	01/01/1900	Active	Prepare Only	<input type="checkbox"/> After Step <input type="checkbox"/> Public

Section Object

The values you specify at the section level, generally apply to all the objects contained within that section.

The following table contains a brief description of each option.

Section Name	Be as descriptive as you can, keeping in mind that you only have an 8-character limit. Develop a naming convention and be consistent throughout your projects.
Description	In this edit box you can enter a more descriptive name for the section. This edit box has a 30-character limit.
Market	Select the Market for which the section is intended. If a particular Market is irrelevant to your batch program, keep the default Market value of Global (GBL).
Platform	Select the target database platform for which this Section definition is to execute from the drop-down list box. Leave this as “default” for all Sections whose defined Actions are not specific to any given database platform. That is, only change this value to a specific platform if there are Actions defined in this section that are to be

	performed only for the selected database platform.
Effective Date	To make a particular section Effective Dated (Active or Inactive based on a specified date), enter the target date in this edit box.
Effective Status	Specify whether a section is active or not. Active refers to being enabled at runtime. Inactive Sections are bypassed at runtime.
Section Type	<p>In the case of an abnormal termination of the program, the value of this system field specifies whether you must restart the section.</p> <p>If a section controls a procedure that, if not run to completion, could corrupt or desynchronize your data, select <i>Critical Updates</i>. Otherwise, use the default value of <i>Preparation Only</i>.</p>
Auto Commit	Specify the commit level for the section by selecting this check box. You can opt to have no commit or you can have Application Engine commit after the step successfully completes.
Access	<p>Controls the Call Section action. You can specify whether a section can be called from another program. To specify Private, deselect the Public check box. To specify Public, select the Public check box. The check box enables you to toggle between <i>Private</i> and <i>Public</i> access:</p> <p>Unselected (off). Indicates that a particular section can be called only from “internal” sections, or sections defined within the same program. To restrict access to the current program only, make sure that <i>Public</i> is not selected.</p> <p>Selected (on). Indicates that a section can be called from both “internal” and “external” sections. To enable Public Access, select the <i>Public</i> check box.</p>

Executing Order of Sections

A section is unique based on the program and section names, and based on its intended database platform and effective date. You can also create unique market-specific sections. When you execute an Application Engine program, it executes sections based on the following order of precedence:

- If a section for the current market exists, execute it. Otherwise, execute the default GBL (global) market section.
- If a section for the current platform, or database exists, execute it. Otherwise, execute the default database platform section.
- If multiple effective-dated sections exists, execute the section with the most recent effective date, based on the current (run) date.

For example, suppose you have two versions of a particular section: SECT01 for Public Sector and SECT01 for Global use. If you request to run the Public Sector version of the program, PeopleSoft Application Engine executes the Public Sector version of SECT01. If the program is running on Oracle, PeopleSoft Application Engine then looks for an Oracle version of the SECT01 for Public Sector.

Adding Steps

After you define a section, you must define the steps within that section. Each step contains actions, which execute SQL statements, do action statements, message log statements, or PeopleCode.

A step represents the smallest unit of work that can be committed in a program. When you first create a program, you have a default MAIN section and step, initially named Step01.

In this section, we discuss how to:

- Insert steps.
- Set step properties.

Inserting Steps

To insert a step:

1. Highlight the section or step that you want to immediately precede the new step.

For example, if you want the new step to be the *first* step in the section, select the section node. Otherwise, select the existing step that you want the new step to follow. The insert always occurs *after* the currently selected node.

2. Select **Insert, Step/Action**.

By default, the steps will be given a default name of *StepN+1* beginning with *Step01*. This name is normally renamed by the user to better define the type of actions this step contains.

Note. The designer continues to increment the step Name until it has a unique step Name within a particular section. If the designer is unable to create a unique name after 50 attempts, a new step is not inserted. Consequently, you should devise your own naming convention to define more descriptive step names.

3. Specify a step name and the remaining values.

If you want to rename the step name, position the cursor in the step name edit box and enter a custom name. PeopleSoft recommends only accepting the default name for building quick, simple programs and for training purposes.

Setting Step Properties

The following table contains the properties available for steps.

Step Name	According to your step-naming convention, uniquely identify each step. This edit box has an 8-character limit.
Step Description	This edit box has a 30-character limit. Here, you may want to briefly describe the step's purpose.
Commit	<p>Specify the commit level for the step. The following briefly describes the commit options at the step level:</p> <p><i>Default.</i> The step inherits what ever commit level you have specified for the section in which the step resides.</p> <p><i>Later.</i> Select to postpone the commit until a subsequent commit occurs, as specified by developers. Again, here you can override the section-level commit, if it happened to be set to <i>After Step</i>.</p> <p><i>After step.</i> Select if you have a commit level of <i>None</i> specified at the section level. This way you can override the section-level commit, and commit a specific step within a section with no other commits.</p>
Frequency	<p>This becomes enabled only when a step contains one of the following actions: Do While, Do Select, or Do Until. This control accepts only numeric values. The value indicates the frequency with which Application Engine should commit. If non-zero, PeopleSoft Application Engine commits every <i>N</i> iterations, and then again after the last iteration.</p>
On Error	<p>Specify how PeopleSoft Application Engine should respond to an error at the step level. The On Error routine behaves the same for both SQL and PeopleCode actions. The program only aborts on errors, not warnings.</p> <p><i>Abort:</i> The application terminates with an error message.</p> <p><i>Ignore:</i> The program continues but logs an error message.</p> <p><i>Suppress:</i> The program continues and presents no error message.</p> <p><i>SQL.</i> Usually the program abends if a SQL Prepare (compile) or execute fails. If you select Ignore or Suppress, errors on executes are suppressed, but errors on compiles still cause the program to abend. Thus, if you select ReUse on an update statement, it fails on the compile if the SQL is incorrect, but it does not fail on a duplicate key error, or some other such error on the execute.</p>

	<i>PeopleCode</i> : PeopleCode error in program—if the return code satisfies the statement "If (nRet & PCM_ERROR)."
Status	Specify whether to activate a step or make it inactive. If the step is currently applicable to your program (and working) you'll probably want to keep it <i>Active</i> . However, if you're still developing or testing the step you may want to make it <i>Inactive</i> . The check box toggles the status of a step.

Note. The On Error property does not apply to compile errors (for example, specifying erroneous SQL statements). It checks only for execution type errors. If your program has a syntax error, the program abends.

Specifying Actions

There are eight types of actions that you can include within a step, and a step can contain multiple actions. Which actions you choose to define for a step depends on the type of results that your program requires at each stage of execution.

The only mutually exclusive actions within a single step are Call Section and SQL Statement; in other words, you cannot add a Call Section action to a step that already contains a SQL Statement action and visa versa. Furthermore, you can include only one of each action type within a single step. Because there are eight types of actions, and two are mutually exclusive, the maximum number of actions a single step can contain is seven.

The following types of actions are available for an Application Engine program:

- SQL
- Do (When, While, Select, Until)
- PeopleCode
- Call Section
- Log Message
- XSLT (enabled for Transform Only program types)

This section also discusses how to:

- Know the execution order of actions.
- Insert actions.
- Set action properties.

See Also

PeopleTools 8.4 PeopleBook: PeopleSoft Integration Broker, Applying Transformation, Translation and Filtering.

SQL Actions

This is the default action type for the first action within a given step. Use this action to perform the following SQL commands on multiple rows:

- UPDATE
- INSERT
- DELETE
- SELECT

Note. Before you insert SQL (View, SQL) into a SQL action within a new application engine program, you must have previously saved the program. This is required because the program name you use to save this definition with is used to relate your program with the SQL objects you are about to create. The same is true for inserting new PeopleCode.

With a SQL action, you use the SQL Editor to create and modify your SQL statement. For example,

```
%Select (AF_PERFM_AET.PREV_ASOF_DT)
SELECT %DateOut (ASOF_DT)
FROM PS_AF_FCST_SCHT%Bind (EPM_CORE_AET.TABLE_APPEND,NOQUOTES)
WHERE AFDEFN_ID = %Bind (AF_CORE_AET.AFDEFN_ID)
AND ASOF_DT = (
SELECT MAX (ASOF_DT)
FROM PS_AF_FCST_SCHT%Bind (EPM_CORE_AET.TABLE_APPEND,NOQUOTES)
WHERE AFDEFN_ID = %Bind (AF_CORE_AET.AFDEFN_ID)
AND ASOF_DT < %Bind (AF_PERFM_AET.ASOF_DT))
```

Note. If you intend to include multiple SQL statements within a single application engine action you should use the meta-SQL construct %EXECUTE. The previous sample SQL statement sample contains bind variables from a previous application engine action.

ReUse Statement

ReUse is an option you can enable to optimize the SQL components of your batch program. ReUse converts any %BIND references to state record fields into real bind variables (:1, :2, and so on), enabling the Application Engine runtime process to compile the statement once, dedicate a cursor, and then re-execute it with new data as often as your program requires. When you are using SQL to process a large volume of rows, one at a time, inside a fetch loop,

compiling each statement you issue can be a considerable performance issue. ReUse is a way to combat potential performance decreases.

Note. You can have PeopleSoft Application Engine recompile a reused statement by using the %ClearCursor function.

You have the following options to apply ReUse to your SQL actions.

Bulk Insert	When used in conjunction with statements like INSERT INTO tablename (field1, field2...) VALUES (%BIND(ref1), %BIND(ref2)), Bulk Insert offers the most powerful degree of performance enhancements related to ReUse. This option turns on ReUse, and, in addition, it holds all the data in a buffer and performs an insert only after a large volume of rows has gathered in the buffer. The number of rows allowed to gather in the buffer depends on your database platform. Keep in mind that storing data in the buffers is applicable only if you've selected Bulk Insert <i>and</i> the SQL is an INSERT statement. For statements other than INSERT/...VALUES, the Bulk Insert option is ignored.
No	Select this option to disable ReUse. With ReUse off, the Application Engine runtime process recompiles the SQL statement every time the loop executes. By default, ReUse is disabled.
Yes	Select this option to enable basic ReUse functionality.

Note. ReUse can offer significant performance gains. However, PeopleSoft recommends *not* using it if %BIND variables are building parts of the SQL statement or are in the field list of a SELECT statement (this does not apply if you use the STATIC option in %BIND).

No Rows

If the SQL (INSERT, UPDATE, and DELETE) associated with the SQL action does not return any rows, you must specify what the Application Engine program should do.

For example, you could use this in a case where you INSERT into a temporary table, and then you intend to perform further operations on the inserted rows (provided that some rows meet the criteria). If the initial INSERT...SELECT provides no rows, you could save the program from having to re-SELECT on the temporary table prior to executing another operation, or you could also prevent the program from performing set operations on the table when there won't be any qualifying rows.

The following list contains the options that you have when no rows are returned.

Abort	The program terminates.
Section Break	PeopleSoft Application Engine exits the current section immediately, and control returns to the calling step.
Continue	The program continues processing.
Skip Step	PeopleSoft Application Engine exits the current step immediately and moves on to the next step. Application Engine ignores the commit for the current step at runtime. If the current step contains only one action, use Skip Step only to bypass the commit.

Note. Using No Rows in conjunction with a Truncate Table operation is unreliable. Some database platforms report "zero rows affected" for truncates, regardless of how many rows were in the table.

Do Actions

There are four types of PeopleSoft Application Engine actions that, although distinct from the others, can be grouped together as:

- Do When
- Do While
- Do Select
- Do Until

Use these actions to control the execution of your program. With these action types you can control the execution of subsequent sections, actions, or SQL statements depending on the results of a “Do” SQL statement in the form of a SELECT. If you were coding in COBOL, you would perform similar actions using the IF and WHILE functions.

Any of the Do actions can control the execution of a section, a SQL statement, a PeopleCode program, or a Log Message. For example, a Do Select can execute a SQL statement for each row returned by the included SELECT.

DO When

- The Do When action is a SELECT statement that allows subsequent actions to be executed if any rows of data are returned.
- This action is similar to a COBOL “IF” statement. A Do When statement runs *before* any other actions in the step. If the Do When statement returns any rows, the next action is executed. If the Do When conditions are not met, the remaining actions within that step are not executed. Your program executes a DO When action only once when the owning step executes.

- The only property that you can specify for the Do When action is the ReUse Statement property, which applies to all SQL-based actions.

Do While

- The Do While action is a SELECT statement that, if present, runs *before* subsequent actions of the step. If the Do While does not return any rows of data, the action terminates. The Do While is identical to the COBOL “WHILE” function. In other words, the subsequent actions within the step are executed in a loop as long as at least one row is returned by the SELECT statement for the DO While action.
- In short, if the Do While does not return any rows, the step is complete.
- The only property that you can specify for the Do While action is the ReUse Statement property, which applies to all SQL-based actions.

Do Select

- The Do Select action is a SELECT statement that executes subsequent actions once for every row of data that the Do Select returns. For instance, a Do Select can execute a SQL statement for each row returned from the SELECT statement. The subsequent actions within the step are executed in a loop based on the results of the SELECT statement. The type of the Do Select determines the specific looping rules.

When you insert a Do Select action, you must specify the action properties described in the following topics.

ReUse Statement	As with any SQL-based action, with a Do Select you have the option of specifying the ReUse Statement level.
Do Select Type	When you add a Do Select action, you can specify the ReUse Statement option available for all SQL-based actions, but you also need to specify the Do Select Type.

The following list describes the three types of Do Selects you can select:

Select/Fetch	With this option, PeopleSoft Application Engine opens a cursor for the Do Select, then, within that cursor, Application Engine performs a Fetch for each iteration of the loop to get each row from the SELECT. When a FETCH results in an “end of table”, the looping is complete. You can’t restart this type of SELECT statement because PeopleSoft Application Engine does not perform a checkpoint or a commit within the step containing this action while Select/Fetch is running. Ultimately, your program ignores the commit settings at runtime until the outermost Select/Fetch completes.
Re-Select	For each iteration of the loop, PeopleSoft Application Engine opens a cursor and fetches the first row. This means that, with Re-Select, your program processes the “first row” returned from the Select statement. It also

means that the cursor gets re-opened for each iteration of the loop. With this type of Fetch, you will typically want some aspect of the loop to eventually cause the Select to return no rows. Otherwise, there is no mechanism in place by which to exit the loop. This type of Do Select *is* restartable.

Restartable

This option is similar to the Select/Fetch in that PeopleSoft Application Engine opens the cursor associated with the Do Select once, and then it performs a Fetch on each iteration of the loop to get each row from the SELECT. However, unlike the Select/Fetch option, you can restart this type of Select and Fetch because PeopleSoft Application Engine performs a checkpoint in the middle of the step. Keep in mind that Application Engine will only *treat* this loop as if it is restartable. Developers need to make sure that the SQL they include within this action is such that, upon restart, it recognizes where the previous run failed and where to restart processing. There are various techniques you can use to achieve this, such as employing a “processed” switch, or to base the next Select on the key.

Special Concerns With DO Select

PeopleSoft Application Engine does not commit a step containing a Do Select with the *Select/Fetch* option enabled until the entire step completes successfully, regardless of what other options you have selected.

For example, suppose at the step level you specified to commit every 100 iterations of the step. One of the actions of this step is a Do Select with Select/Fetch chosen. Because PeopleSoft Application Engine will not checkpoint or commit while the Do Select is active, the transaction performed by the actions within a step will not be committed until the entire step completes successfully. Again, this is also true if any sections are called from inside the loop.

Do Until

- A Do Until action is a Select statement that runs *after* each action when a step completes. If the Select returns any rows of data, the step terminates.
- Use a Do Until if you want the “processing actions” to execute at least once, and to execute over and over until a certain condition is true, such as until a Select returns some rows.
- You can also use a Do Until to stop a Do Select prematurely. For example, if the Select for the Do Until does not return any rows, then the actions in the step are repeated (except if a Do When appears in the step). Normally, a Do Select continues until no rows are returned. If any rows of data are returned, the Do Select stops and the step is not repeated.

- The only property that you can specify for the Do Until action is the ReUse Statement property, which applies to all SQL-based actions.

PeopleCode Actions

Use this action type to insert PeopleCode within your Application Engine program. You can invoke the PeopleCode Editor directly from the designer interface to code your PeopleCode programs.

With a PeopleCode action, there is only one property that you can specify—On Return.

Use the On Return value to determine how your Application Engine program reacts based on the return of your PeopleCode program. The On Return setting takes effect if your PeopleCode program issues a “return 1” or “exit 1.” You can use the TRUE keyword in place of a non-zero numeric return.

The following list contains the valid options for On Return:

Abort	The program issues an error and exits immediately.
Break	The program exits the current step and section, and control returns to the calling step.
Skip Step	The program exits the current step, and continues processing at the next step in the section. If this is the last step in the section, the calling step resumes control of the processing.

The following is an example of PeopleCode within an Application Engine program.

```
If All (FSI_CORE_AET.PF_CONSTRAINT_CODE) Then
    Exit True;
Else
    Exit False;
End-If;
```

Call Section Actions

Use the Call Section action to call another section defined in an Application Engine program. You can call a (local) section defined within your current program and you can make external calls to a section defined in another Application Engine program.

The external section you intend to call must have its access property set to *Public*. If a section’s access property is set to *Private*, that section can be called only from within the same program. By default, a section’s access property is *Private*. If you attempt to make a call to a section that does not allow external calls, you receive an error message at runtime.

Note. You can call only programs that reside within the same database as the “calling” program.

Section Name

The names that appear in the drop-down list box are those that are defined in the program that appears in the Program ID edit box. If you want to call a section that is defined in an external program, always select the program name in the Program ID edit box prior to selecting the section name.

Also use the Call Section action to call an entire external program. First select the Program ID, then select section MAIN. At runtime, this call executes the entire program defined by the value in Program ID.

Note. PeopleSoft Application Designer does not prevent you from calling the main section of the current program or the current section. For instance, Section1 can contain a step that has a local Call Section reference for Section1. This enables recursive calls, and should therefore be used with caution.

Program ID

Because you can call sections defined in the current program or within external programs, you must first specify the Program ID of the program containing the section you intend to call.

The default value is (*current*). If you call a section defined in another program, make sure that you first select the appropriate external program from the drop-down list box. The drop-down list box contains the names of all program definitions that currently exist in the database.

Dynamic

Use the AE_APPLID and AE_SECTION fields in the state record to execute different sections depending on the conditions a program encounters during runtime.

These two fields must be defined on the default state record for the program. If AE_APPLID is not present or is blank (at run time), the current program is substituted for the AE_APPLID value. If AE_SECTION is not present or is blank, an error occurs.

When issuing a dynamic call, *both* the section and the Program ID must be dynamically set. You enable a dynamic call by first having your program store different section names in the AE_SECTION field, and different program names in AE_APPLID field. The values you insert in these fields are normally based on various conditions met within your program. You then create a Call Section action that calls the section name defined in the State record field by selecting the **Dynamic** check box.

Selecting dynamic automatically populates the AE_SECTION field with the symbolic value %Section, and the Program ID field with the symbolic value %AEAPPLID. At runtime, the program calls the section name stored in AE_SECTION that belongs to the program name defined by AE_APPLID.

Program Properties of Called Sections

When you call a section defined in an external program, the current program (the program containing the defined Call Section) defines the properties that apply to the running process. Suppose tracing is enabled for the current program, but tracing is disabled for the called program section. In this case, the called program has the trace option enabled at runtime because it inherits the calling program's properties.

For example, if program A calls program B *and* program B calls program C, then the properties of A apply to both programs B and C. The calling program always controls the properties for the called program. In this case, program A controls the properties for program B, and because program B inherits the properties of program A, when program B calls program C, program A's properties *also* apply to program C.


Note. Although program properties are inherited, note that state records do not follow this inheritance model.

Sharing State Records of Called Programs

When you call a program from another program, the called program's default state record becomes active until processing returns to the initial program. However, all of the state records associated with both programs are available. State records that are common between the two programs "share" values. This means that if you need to communicate between the two programs, as in share %BIND variables, you must define the same state records in both programs.

Log Message Actions

Use this type of action to write a message to the Message Log. The Message Log refers to the PeopleTools table (PS_MESSAGE_LOG) where execution messages reside. Any substitution parameters are written to PS_MESSAGE_LOGPARM.

 Log Message	Log Message description		
	Message Set:	Number:	Parameters:
	10662	278	%BIND(AF_CORE_AET_AFDEFN_ID)

Log Message action

You can use the Log Message action to insert any type of messages you need. Typically, Log Message writes error messages to the Message Log, but you could also write informational or status messages.

Note. You can also use MessageBox PeopleCode to populate PS_MESSAGE_LOG instead of using the Log Message action. This enables you to easily record errors encountered within Application Engine PeopleCode programs.

Message Set, Number

Identifies the specific message defined in the Message Catalog.

Parameters

These are a comma-delimited list of values to substitute for the message variables (%1, %2, and so on) in the message text. These parameters can be hard-coded values or %Bind references. The information specified is inserted in the PS_MESSAGE_LOG at runtime, and any %Bind values are replaced by the current state record field values. You can then view the logged messages from the Process Monitor page.

Note. Both message set and number can be a hard-coded numeric value or a "[record.]field" reference for dynamic message logging (record is optional; if missing, the default state record is assumed).

For example, using message set 1012, number 10 = "The total number of %1 rows exceeds the control count value, %2," and the following parameters:

```
Invoice, %Bind(CONTROL_CNT)
```

Suppose you run this program with the CONTROL_CNT field value of 120. When the process completes, the following message would be included on the Process Details dialog box in Process Monitor:

```
The total number of Invoice rows exceeds the control count value, 120.
```

XSLT Actions

These are used for transformation programs only.

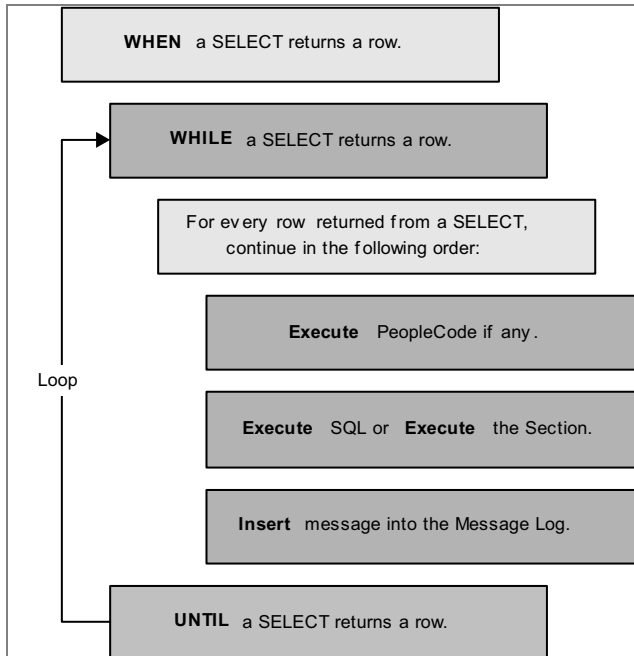
See Also

PeopleTools 8.4 PeopleBooks: PeopleSoft Integration Broker, "Applying Transformation, Translation, and Filtering," Using XSLT for Transformation.

Executing Order of Actions

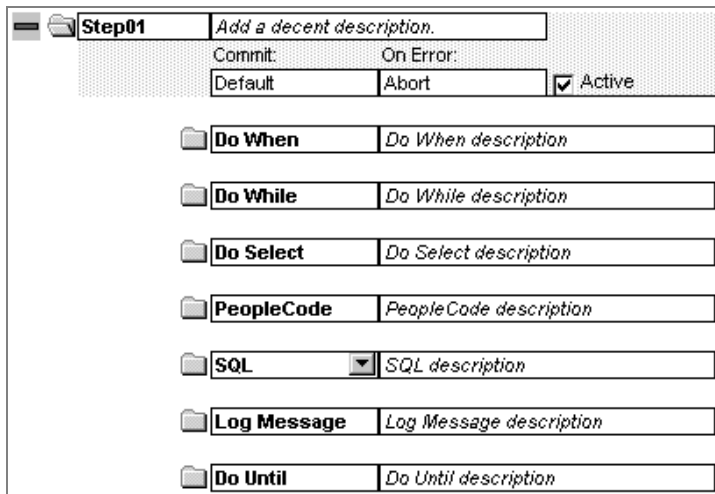
At runtime, the system evaluates actions by type and executes them within a strict hierarchy. For example, if both a Do When and PeopleCode action exist within a given step, Application Engine always executes the Do When first.

The following example depicts the sequence and level of execution for each type of action [we need to add XSLT to this picture]:



Action execution hierarchy

Notice that as you add actions to a step in the definition view, the actions are initially inserted after the selected definition (the owning step or a prior action). However, following a save request or a refresh of the view, the designer reorders all actions to match the execution hierarchy. This feature helps you visualize the sequence in which each step of your program logic executes.



Actions in execution order in designer view [need to include XSLT]

Note. In the previous example, the SQL action and a Call Section action are interchangeable, in that one or the other would fill that position in the hierarchy. Only one of these two action types can appear within a step as they are mutually exclusive.

Remember the following when inserting actions:

- You cannot have more than one action of a specific type within the *same* step.
- You cannot have a SQL action and a Call Section action within the *same* step.
- You can only define XSLT type actions for programs defined as Transformation types (see Program Properties).

Inserting Actions

To insert an action:

1. Highlight the step in which you want to insert the action.
2. Insert the action.

You do this using one of the following methods:



- Select Insert, Step/Action.
 - Right-click on the step and select Insert Step/Action from the pop-up menu.
3. Select the action type from the drop-down list box, or when current action type is selected, type the first character or so of the desired action type, then press TAB. The first (or only) type qualified by your entry is updated in this control.
 4. Enter a description of the action in the description edit box.
 5. Specify the appropriate properties for the action you selected.

Setting Action Properties

To modify action properties, you must have the definition view active. Because there are a variety of actions that you can include within a step, there are different sets of properties specific to a particular action type. Depending on what action type you select, the properties that appear will change.

For example, you can specify the reuse feature with a SQL action. However, because this feature doesn't apply to a PeopleCode action, you would instead need to specify how to respond to the PeopleCode program's return value.

The following example depicts how you can select action-specific properties for different action types.

 Do Select	<input type="text" value="Do Select description"/>
	ReUse Statement: <input type="text" value="No"/> Do Select Type: <input type="text" value="Select/Fetch"/>
 Call Section	<input type="text" value="Call Section description"/>
	Section Name: <input type="text" value="STATS"/> Program ID: <input type="text" value="AETESTPROG"/> <input type="checkbox"/> Dynamic

Actions and associated properties

PeopleCode and all SQL action types invoke the related PeopleTools Editor to define or maintain the related text. The following table describes the properties you can select for each action type. The actions are arranged according to properties that apply to them.

Action Type	Object Type	Configurable Properties	Editor
Do When Do While Do Until	SQL Select (Iterative)	ReUse Statement: Selecting this means the database engine needs to compile the SQL only once, which reduces SQL overhead. Choices are Yes, No, and Bulk Insert.	SQL Editor
Do Select	SQL Select	ReUse Statement: (See Do When, Do While, and Do Until.) Do Select Type: You must specify one of the following: Select/Fetch, Reselect, or Restartable.	SQL Editor
PeopleCode	PeopleCode	On Return: If your PeopleCode program provides a false result, you can have PeopleSoft Application Engine respond by doing one of the following: Abort, Break, or Skip Step.	PeopleCode Editor
SQL	SQL Statement	Reuse Statement: (See Do When, Do While, and Do Until.) No Rows: If your SQL statement doesn't affect any rows, you can specify that PeopleSoft Application Engine responds in one of the following ways: Abort, Section Break, Continue, or Skip Step.	SQL Editor

Action Type	Object Type	Configurable Properties	Editor
Call Section	Application Engine Section	<p>Section Name: If you are calling another section, you need to specify the appropriate section here. Only the names of Public Sections defined in the program identified in Program ID will appear in the Section Name drop-down list box.</p> <p>Program ID: Because you can call sections in the current program or sections that exist within other programs, you must specify the Program ID, or Program Name, of the program containing the section you intend to call. The drop-down list box contains all of the program definitions that currently exist.</p>	N/A
Log Message	Message Log	<p>Message Set: Identifies a message set within the Message Catalog.</p> <p>Number: Identifies a particular message within a Message Set.</p> <p>Parameters: A list of comma-delimited values to substitute for %1, %2, and so on markers in the message text.</p>	N/A
XSLT	XSLT	There are no additional properties defined for XSLT types.	SQL Editor

CHAPTER 3

Developing Efficient Programs

This chapter covers the following topics:

- State records.
- Commit considerations.
- Re-using statements.
- Bulk insert.
- Set processing.

State Records

You assign variables for your Application Engine program through state records, while sections, steps, and actions pass values to subsequent program steps through state records.

You can have any number of state records associated with a particular Application Engine program. However, only one record can be the default state record. You can specify both work (derived) and “physical” (SQL table) records to be used as state records. The only difference is that derived state records cannot have their values saved to the database at commit time, and so the values would be lost during a restart. Therefore, PeopleSoft Application Engine erases the contents of derived state records at commit time if Restart is enabled for the current process.

A PeopleSoft Application Engine state record must have a process instance defined as the first field and the *only* key field and the state record name must end with `_AET`.

Not all the database columns referenced in your program must be in the state record, just the columns that must be selected into memory so those values can be referenced in a subsequent program action. You may also want to include additional fields to hold pieces of dynamic SQL, to use as temporary flags, and so on.

PeopleSoft Application Engine supports long fields unlike COBOL or SQR. However, it allows only one long field per state record. You set a maximum size for the field in PeopleSoft Application Designer and make sure that the data space is compatible with the size of the field that you set.

PeopleSoft Application Engine also supports image fields and long text fields.

The screenshot shows a window titled "AETESTPROG_AET (Record)" with a "Record Fields" tab. Below the tab is a table with the following data:

Num	Field Name	Type	Len	Format
1	PROCESS_INSTANCE	Nbr	10	
2	AE_INT_1	Nbr	1	
3	RECNAME	Char	15	Upper
4	AE_CNV_IN_FLD_NM	Char	18	Upper
5	AE_CNV_OUT_FLD_NM	Char	18	Upper
6	AE_CNV_FILL_CHAR	Char	1	Upper
7	AE_CNV_JUSTIFY	Char	1	Upper
8	AE_STRIP_ZERO	Char	1	Upper

Sample state record

During batch processing, PeopleSoft Application Engine automatically performs all state record updates. When a program starts, it inserts a row into the state record that corresponds to the Process Instance assigned to that program run. PeopleSoft Application Engine updates the record whenever a commit operation occurs. When restart is enabled and a commit occurs, all state records that have been updated in memory are written to the database, except for derived state records, which are initialized instead.

After the program completes successfully, PeopleSoft Application Engine deletes the corresponding row in the state record. There is only one row in the state record for each process instance, or program. Multiple programs can use the same state record, and each program has its own row based on the unique process instance key.

To set values in the state record, you use the %SELECT construct in a SQL statement or write PeopleCode that references the state field with the standard "record.field" notation. To reference fields in the state record, use the %BIND construct.

Note. When a state record is a work record, no database updates can be performed on the record. Consequently, if your program is restartable, your code must take into account that this "memory" will be lost if the program abends. PeopleSoft Application Engine automatically re-initializes fields on all work records after each commit.

In this section we discuss how to:

- Share state records.
- Choose a record type for state records.

Sharing State Records

State records can be used by multiple sections and by multiple programs. When you call a section in another program, any additional state records defined for that program (as in state records that are not already in use by the calling program) are initialized, even if the program has been called previously during the run. However, state records that are common to both programs retain their current values.

To reference variables that exist within a state record, use the following:

```
%BIND(fieldname)
```

Unless a specific record name is specified preceding the fieldname, %BIND references the default state record. To reference a state record other than the default, use the following:

```
%BIND(recordname.fieldname)
```

In the case of a called program or section, if the called program has its own default state record defined, then PeopleSoft Application Engine uses that default state record to resolve the %BIND(fieldname). Otherwise, the called program inherits the calling programs default state record. In theory, the called program does not require a state record if all the fields it needs for processing exist on the calling program's state record.

For those state records that are shared between programs (during an "external" Call Section), any changes made by the called program remain when control returns to the calling program. This means that any subsequent actions in the calling program can access residual values left in the common state records by the called program. This can be useful at times to return output values or status to the calling program, yet it can also cause unforeseen errors.

Generally, a called program should not share state records with the caller unless you need to pass parameters between them. Most programs have their own set of state records unless a program calls another program that requires specific input or output variables. In that case, you must include the state record of the called program into the calling program's state record list, and make sure to set the input values before issuing the Call Section.

Choosing a Record Type for State Records

As a general rule, if you want preserve any state record field values across commits in your program, then you should store those values in a state record with a Record Type of SQL Table. Only derived work type state records store values that don't need to be accessed across commits. Derived work records are, however, an excellent choice for temporary flags and dynamic SQL "containers" that are set and then referenced immediately. Because these values aren't needed later, you don't want to have to save them to the database at each commit. When you create your state record in PeopleSoft Application Designer, you should have an idea regarding how your state record will be used. With this information, you can select the appropriate Record Type to build.

With Application Engine programs, state records that are derived work records function the same as those which are of a SQL Table type. However, there is one notable distinction: unless you have disabled Restart, derived work records have their field values re-initialized after each commit. Therefore, unless you anticipate this behavior, you may encounter problems. One quick way to diagnose such a problem is to examine a trace. Typically, you see %BINDs resolved to values prior to a commit, and then after the commit, they have no value.

This behavior is necessary to ensure consistency in the event of an abend and restart. During the restart, PeopleSoft Application Engine begins, or restarts, at the point of the last successful commit and restores the values of any state records with corresponding database tables. Derived/Work records aren't associated with a physical database table, and consequently they can't be restored in the event of a restart.

Commit Considerations

For new Application Engine programs that you develop, by default, the commit values at the section and the step level are turned off. This means that, by default, *no* commits occur during the program run, except for the implicit commit that occurs after the successful completion of the program.

It's up to you to divide your program into logical units of work by setting commit points within your program. Typically, after PeopleSoft Application Engine completes a self-contained task, it might be a good time to commit. How often you apply commits affects how your program performs in the event of a restart. For set processing programs, commit early and often. For row-based processing, commit after every *n* iterations of the main fetch loop that drives the process.

If you have a step with a Do While, Do Until, or a Do Select action, you can set the frequency option, which drives your commit level. This enables you to set a commit at the step level that occurs after a specified number of iterations of your looping construct. Application Engine programs commit whenever they are instructed to do so, and because of this there's nothing preventing a developer from having both the frequency option enabled as well as having other individual commits inside of a loop.

The only restriction for batch runs occurs when you have restart enabled, and you are inside a Do Select that is of the Select/Fetch type (instead of "Re-select" or "Restartable"). With Select/Fetch, all commits inside the loop are ignored, including the commit frequency if it's set.

"Restartable" is similar to Select/Fetch, except that you are implying to PeopleSoft Application Engine that your SQL is structured in such a way that it filters out rows that have been "processed" and committed. This enables a successful restart. One technique for accomplishing this is to have a processed flag that you check in the where clause of the Do Select, and you perform an update inside the loop (and before the commit) to set the flag to "Y" on each row you fetch.

The commit logic is designed to do a commit regardless of whether any database changes have occurred. The program commits as instructed by the developer, except when the program is restartable *and* at a point where a commit would affect restart integrity—inside a non-restartable Do Select, for example.

When you have a step set to commit by default, it means that the step's commit frequency is controlled by the section's auto commit setting. If the section is set to commit after every step, then the program commits. Otherwise, the program never commits unless the step is explicitly set to commit afterward.

Note. The Commit After, Later setting at the step level enables you to override the section setting if you don't want to commit after a particular step.

%TruncateTable Considerations

Some databases, such as Oracle, issue an implicit commit for a "truncate" command. If there were other pending (uncommitted) database changes, the results would differ if an abend

occurred after the %TruncateTable. To ensure consistency and restart integrity, PeopleSoft Application Engine checks the following:

- Whether there are pending changes when resolving a %TruncateTable.
- If the program is at a point where a commit isn't allowed.

If either condition is true, PeopleSoft Application Engine issues "delete from" syntax instead.

"No Rows" Considerations

The default for the No Rows setting (on the Action) is Continue. This setting controls how your program responds when a statement returns no rows. If you leave it set to Continue for commit reasons. In the case of %UpdateStats you may want to set No Rows to Skip Step and thus skip the commit. For example, suppose you have a single INSERT into a table, followed by an %UpdateStats. If the stats were current before the INSERT and the INSERT affects no rows, then the %UpdateStats is unnecessary.

Re-Using Statements

One of the key performance features of PeopleSoft Application Engine is the ability to “re-use” SQL statements by dedicating a persistent cursor to that statement. The following topics describe some items to keep in mind if you decide to take advantage of this feature.

ReUse (SQL Action Property)

Unless you select the ReUse property for you SQL Action, %BIND fields are substituted with literal values in the SQL statement. This means that the database has to recompile the statement every time it is executed.

However, selecting ReUse converts any %BIND fields into real bind variables (:1, :2, and so on), enabling Application Engine to compile the statement once, dedicate a cursor, and re-execute it with new data multiple times. This reduction in compile time can result in dramatic improvements to performance.

In addition, some databases have SQL statement caching. This means that every time they receive SQL, they compare it against their cache of previously executed statements to see if they have seen it before. If so, they can reuse the old query plan. This works only if the SQL text matches exactly. This is unlikely with literals instead of bind variables.

When using ReUse, keep the following items in mind:

- ReUse is valid only for SQL Actions.
- Use ReUse only if you do not use Bind variables for column names.
- Use ReUse only if you have no %BINDS in the Select list.

- If the SQL is dynamic, as in you are using %Bind to resolve to a value other than a standard Bind value, *and* the contents of the Bind will change each time the statement gets executed, then you can't enable ReUse. In this situation the SQL is different each time (at least from the database perspective) and therefore can't be "reused."
- If you use the NOQUOTES modifier inside %Bind, there is an implied STATIC. For dynamic SQL substitution, the %Bind has a CHAR field and NOQUOTES to insert SQL rather than a literal value. If you enable ReUse, this means the value of the CHAR field gets substituted inline, instead of using a Bind marker (as in :1, :2, and so on). The next time that Application Engine Action executes, the SQL that it executes will be the same as before, even if the value of a static bind has changed.
- To re-prepare a ReUsed statement from scratch, because one of the static binds has changed and the SQL has to reflect that, use %ClearCursor.

If you are running DB2 on OS/390 or AS/400, use ReUse only when you are not using %BINDS as operands of the same operator, as shown in the following example:

```
UPDATE PS_PO_WRK1
      SET TAX = %BIND(STATE) + %BIND(FED)
```

This causes Error -417. You can modify the SQL so that you can use ReUse successfully. Suppose your program contains the following SQL:

```
UPDATE PS_PO_WRK1
      SET TAX = 0
      WHERE %BIND(TAX_EXEMPT) = %BIND(TAX_STATUS)
```

If you modify it to resemble the following SQL, ReUse will work:

```
UPDATE PS_PO_WRK1
      SET TAX = 0
      WHERE %BIND(TAX_EXEMPT, STATIC) = %BIND(TAX_STATUS)
```

Using Bulk Insert

By buffering rows to be inserted, some databases can get a considerable performance boost. PeopleSoft Application Engine offers this non-standard SQL enhancement on the following databases: Oracle, Microsoft SQLServer, and DB2. This feature is named Bulk Insert. For those database platforms that do not support bulk insert, this flag is ignored.

You should consider using this feature *only* when the INSERT SQL is called multiple times in the absence of intervening COMMITs.

PeopleSoft Application Engine ignores the Bulk Insert setting in the following situations:

- SQL is not an INSERT.
- SQL is other than an INSERT/VALUES statement that inserts one row at a time. For instance, the following statements would be ignored: INSERT/SELECT, UPDATE, or DELETE.

- SQL does not have a VALUES CLAUSE.
- SQL does not have a field list before the VALUES clause.

In the previous situations, PeopleSoft Application Engine still executes the SQL; it just doesn't take advantage of the performance boost associated with Bulk Insert.

To re-prepare or flush a Bulk Insert statement because one of the static binds has changed and the SQL has to reflect that, use %ClearCursor. A flush also occurs automatically before each commit.

Set Processing

Set processing is an SQL technique used to process groups, or sets of rows at one time rather than processing each row individually. Set processing enables you to apply your business rule directly on the data (preferably while it resides in a temporary table) in the database using an update or insert/select statement. Most of the performance gain is because the processing occurs in the database instead of loading the data into the application program, processing it, and then inserting the results back into the database tables. Because the data never leaves the database with set processing (whether it remains in the same table), you effectively eliminate the network round trip and database API overhead.

Note. Because the updates in set processing occur within the database, PeopleSoft encourages you to use temporary tables to hold transient data while your program runs. Although temporary tables are not required for set processing, they are often essential to achieve optimum performance in your batch program.

In this section, we discuss the following topics:

- How to use set processing effectively.
- Tips for your Application Engine programs.
- Examples of set processing.

Using Set Processing Effectively

Keep the following in mind if you are developing new or upgrading older Application Engine programs to adhere to a set-based model.

SQL Expertise

Even if you're developing row-by-row programs with Application Engine, you should be a SQL expert. With set-based programs, this is especially true. The following concepts are particularly important:

- Group by and having clauses
- Complex joins
- Sub-queries (correlated and non-correlated)
- Tools for your database to analyze complex SQL statements for performance analysis

Typically, you use these SQL constructs to refine or filter the set to contain only the rows that meet particular criteria. Keep in mind that SQL is what you code with PeopleSoft Application Engine, and Application Engine passes that SQL directly to the database where it gets processed. If you have a complex SQL statement that works functionally, it may not necessarily perform well if it is not properly tuned.

Planning

Well-constructed, robust, and efficient Application Engine programs are usually the product of a detailed planning stage where loops, program flow, the use of temporary tables, sections, steps, and so on, are discussed.

In an ideal situation, PeopleSoft suggests that you address batch processing as a whole while you are designing the system. Sometimes, systems analysts and developers focus primarily on the online system during the database design, and then they consider the batch component within the existing database design. Set processing works best in an environment where the data models are optimized for set processing.

For example, you could have a separate staging table for new data that hasn't been processed rather than having numerous cases where existing rows in a table get updated. In set processing, it is much easier to process the data after moving it to a temporary table using an insert or select statement rather than just using an update. PeopleSoft recommends that you avoid performing updates on "real" application tables, and try to perform your updates on temporary tables. To minimize updating "real" application tables, structure your data model to prevent that.

Another important consideration is keeping historical data separate from active transactions. After the life cycle of given piece of transaction data is over, so that no more updates are possible, consider moving that data to an archive or history table and deleting it from the real transaction table. This will keep the number of rows in the table to a minimum, which improves performance for queries and updates to your active data.

Temporary Tables

Although temporary tables are not necessarily required for set processing, you will find that well designed temporary tables complement your set-based program in a variety of ways.

Creating temporary tables enables you to achieve one of the main objectives involved with set based processing—the processing remains on the database server. By storing transient data in temporary tables, you avoid the situation where the batch program fetches the data, row-by-row, and runs the business rule, processes the data, and then passes the updated data back to the database. If the program were running on the client, you encounter performance issues due to the network roundtrip and the diminished processing speed of a client (compared to the database platform).

Your temporary tables should be designed to accomplish the following:

- Hold transaction data for the current run or iteration of your program.
- Contain only those rows of data affected by the business rule.
- Present key information in a de-normalized or “flattened” form, which provides the most efficient processing.
- Switch the keys for rows coming from the master tables if needed. A transaction may use a different key than what appears on the master tables.

De-normalized Tables

The most efficient temporary tables store the data that a program needs to access in a de-normalized or flattened form. Because most programs need to access data that resides in multiple tables, it is more sensible to consolidate all of the affected and related data into one table, the temporary table. It’s much more efficient for the program to run directly against the flattened temporary table rather than relying on the system to materialize complex joins and views to retrieve or update necessary data for each transaction.

If your program requires the use of a complex view to process transactions, then PeopleSoft recommends that you resolve the view into a temporary table for your program to run against. Each join or view that needs to materialize for each transaction consumes system resources and affects performance. In this approach, the system applies the join or view once (during the filtering process), populates the temporary table with the necessary information that the program needs to complete the transaction, and then runs the program against the temporary table as needed.

For example, consider the following situation:

- A program needs to update 10,000 rows on the Customer table, which contains 100,000 rows of data.
- The Customer table is keyed by SetID.
- To complete the transaction, the program references data that resides on a related table called PS_SETCNTRL_REC.
- PS_SETCNTRL_REC is used to associate SetID and BUSINESS_UNIT values.
- The transaction is keyed by BUSINESS_UNIT.

Given that set of circumstances, the most efficient processing method would be similar to the following:

- Isolate affected or necessary data from both tables, and insert that into the temporary table. Now, instead of dealing with a 100,000-row CUSTOMER table and a join to a related table, the program faces a 10,000-row temporary table that contains all of the required data to join directly to the transaction data, which can also be in a temporary table. If all necessary columns reside on the temporary tables, the program can modify all the rows at once in a simple UPDATE statement.

Note. The previous bullet item presents two different uses of temporary tables. In one situation, the temporary table is designed to hold setup/control data in a modified form. In the other situation, the temporary table is designed to hold transaction data in a denormalized form, perhaps with additional “work” columns to hold intermediate calculations.

- Make sure the data appears in a denormalized form for optimum processing.
- Because the transaction is keyed by BUSINESS_UNIT, so should the temporary table that holds the control data. In this case, the table that holds the control data is the CUSTOMER table.

Tips for Your Application Engine Programs

When you write set-based programs in PeopleSoft Application Engine, there are some general guidelines to keep in mind.

Hybrid Application Programs

A set-based program is not an all-or-nothing situation. There are some rules that call for row-by-row processing, but our experience reveals that these rules are the exception. When an exception arises, PeopleSoft recommends not to let the exception ruin the chances of a "mostly" set-based program. You can have a row-by-row component within a mostly set-based program.

For example, suppose your program contains five rules that you'll run against your data. Four of those rules lend themselves well to a set-based approach while the fifth requires a row-by-row process. In this situation, PeopleSoft suggests that you run the four set-based steps or rules first, and then run the row-by-row portion last to resolve the exceptions. Although it's not pure set-based processing, you still obtain better performance than if the entire program used a row-by-row approach.

And, when performing a row-by-row update, reduce the number of rows and the number of columns that you select to an absolute minimum to decrease the data transfer time.

For logic that cannot be coded entirely in set, try to process most of the transactions in set, and process only the exceptions in a row-by-row loop. A good example of an exception is the sequence numbering of detail lines within a transaction when most transactions have only a single detail line. You can default the sequence number on all the detail lines to 1 in an initial set-based operation, then execute a select statement to retrieve only the exceptions (duplicates) and update their sequence numbers to 2, 3, and so on.

Avoid the tendency to expand row-by-row processing for more than what is necessary. For example, just because you're touching all of the rows of a given table in a specific row-based process, it doesn't mean that you gain in efficiency by running the rest of your logic on that table in a row-based manner.

When updating a table, it's OK to add another column to be set in the update statement. However, PeopleSoft does not recommend adding another SQL statement to your loop just because your program happens to be looping. If you can apply that SQL in a set-based

manner, in most cases, you will achieve better performance with a set-based SQL statement outside the loop.

Filtering

Using SQL, filter the "set" to contain only those rows that are affected or meet the criteria and then run the rule on them. Use the where clause to minimize the number of rows to reflect only the set of affected rows.

Two-Pass Approach

Use a two-pass approach wherein the first pass runs a rule on all of the rows, and the second pass resolves any rows that are exceptions to the rule. For instance, bypass exceptions to the rule during the first pass, and then address the exceptions individually in a row-by-row manner.

Parallel Processes

Divide sets into distinct groups, and then run the appropriate rules or logic against each set in parallel processes. For example, in terms of employee data, you could split the population into distinct sets of "hourly" and "salary," and then you could run the appropriate logic for each set in parallel.

Flat Temporary Tables

"Flatten" your temporary tables. The best temporary tables are de-normalized and follow a flat file model for improved transaction processing.

For example, Payroll control data might be keyed by Set ID and Effective Dates rather than by Business Unit and Accounting Date. Use the temporary table to denormalize the data, and switch the keys to "Business Unit" and "Accounting Date." Afterwards, you can construct a straight join to the Time Clock table, keyed by "Business Unit" and "Date."

Techniques to Avoid

- If you have a series of identical temporary tables, examine your refinement process.
- Insert/updates from one table to the next during the refinement process.
- Don't attempt to accomplish a task that your database platform does not support, as in complex mathematics, non-standard SQL, and complex analytical modeling. Use "standard" SQL for set processing.
- Although sub-queries are a useful tool for refining your set, make sure that you're not using the same one multiple times. If you are using the same sub-query in more than one statement, you should probably have denormalized the query results into a temporary table. The key is to identify the sub-queries that appear frequently and, if possible, "flatten" or denormalize the queried data into a temporary table.

Examples of Set Processing

The following sections each contain an example of set processing.

Payroll

In this example, suppose the payroll department needs to give a \$1000 salary increase to everybody whose department made more than \$50,000 profit. The following pseudo code enables you to compare the row-by-row and set-based approaches.

- **Row-by-Row**

```
declare A cursor for select dept_id from department where profit > 50000;
open A;
fetch A into p_dept_id
while sql_status == OK
    update personnel set salary = (salary+1000) where dept_id = p_dept_id;
    fetch A into p_dept_id;
end while;
close A;
free A;
```

- **Set-Based**

```
update personnel set salary = (salary + 1000)
    where exists
        (select 'X' from department
         where profit > 50000
         and personnel.dept_id = department.dept_id)
```

Note. The previous example employs a correlated sub-query, which is important in set-based processing.

Using Temporary Tables

One technique to improve your database performance is to use a temporary table to hold the results of a common sub-query. Effective Dating and Set ID indirection are common types of sub-queries that you can replace with joins to temporary tables. With the joins in place, you can just access the temporary table instead of doing the sub-query multiple times. Not only do most databases prefer joins to sub-queries, but if you combine multiple sub-queries into a single join as well, the performance benefits can be significant.

In this SetID indirection example, you see a join from a transaction table (keyed by BUSINESS_UNIT and ACCOUNTING_DT) to a setup table (keyed by SETID and EFFDT).

To accomplish this using a single SQL statement, you need to bring in PS_SET_CNTRL_REC to map the business unit to a corresponding SetID. This is typically done in a sub-query. You also need to bring in the setup table a second time in a sub-query to get the effective date (MAX(EFFDT) <= ACCOUNTING_DT). If you have a series of similar statements, this can be a performance issue.

The alternative is to use a temporary table that is the equivalent of the setup table. The temporary table would be keyed by BUSINESS_UNIT and ACCOUNTING_DT instead of SETID and EFFDT. You populate it initially by joining in your batch of transactions (presumably also a temporary table) once, as described previously, to get all the business units and accounting dates for this batch. From then on, your transaction and setup temporary tables have common keys, which allows a straight join with no sub-queries.

The example in this topic (taken from PeopleSoft Receivables and modified slightly for clarity) may help to illustrate these concepts. The following paragraphs provide some context for the examples.

The original setup table (PS_ITEM_ENTRY_TBL) is keyed by SETID, ENTRY_TYPE and EFFDT.

The de-normalized temporary table version (PS_ITEM_ENTRY_TAO) is keyed by PROCESS_INSTANCE, BUSINESS_UNIT, ENTRY_TYPE and ACCOUNTING_DT, and carries the original keys (SETID and EFFDT) as simple attributes for joining to other related setup tables, as in PS_ITEM_LINES_TBL for this example.

If the program references the setup table in only one insert/select or select statement, you wouldn't see increased performance by de-normalizing the temporary table. But, if several SQL statements are typically executed in a single run, all of which join in the same setup table with similar SetID and effective date considerations, then the cost of populating the temporary table up front provides long-term advantages.

- Before

```

INSERT INTO PS_PG_PENDDST_TAO (...)

SELECT
. . . . .
  ( (I.ENTRY_AMT_BASE - I.VAT_AMT_BASE) * L.DST_LINE_MULTIPLR *
L.DST_LINE_PERCENT / 100 ),

  ( (I.ENTRY_AMT - I.VAT_AMT) * L.DST_LINE_MULTIPLR * L.DST_LINE_PERCENT / 100
),
. . . . .
FROM PS_PENDING_ITEM I, PS_PG_REQUEST_TAO R, PS_ITEM_LINES_TBL L,
     PS_ITEM_ENTRY_TBL E, PS_SET_CNTRL_REC S, PS_BUS_UNIT_TBL_AR B
. . . . .

WHERE

  AND L.ENTRY_REASON = I.ENTRY_REASON
  AND L.SETID = E.SETID
  AND L.ENTRY_TYPE = E.ENTRY_TYPE
  AND L.EFFDT = E.EFFDT
. . . . .

  AND E.EFF_STATUS = 'A'
  AND S.RECNAME = 'ITEM_ENTRY_TBL'

```

```

AND S.SETID = E.SETID
AND S.SETCNTRLVALUE = I.BUSINESS_UNIT
AND E.ENTRY_TYPE = I.ENTRY_TYPE
AND E.EFFDT = ( SELECT MAX(EFFDT) FROM PS_ITEM_ENTRY_TBL Z
                WHERE Z.SETID = E.SETID
                  AND Z.ENTRY_TYPE = E.ENTRY_TYPE
                  AND Z.EFF_STATUS = 'A'
                  AND Z.EFFDT <= I.ACCOUNTING_DT )
AND B.BUSINESS_UNIT = I.BUSINESS_UNIT
/

```

- After

```

INSERT INTO PS_ITEM_ENTRY_TAO
. . . . .

SELECT DISTINCT %BIND(PROCESS_INSTANCE), I.BUSINESS_UNIT, I.ACCOUNTING_DT,
E.ENTRY_TYPE ...
. . . .

FROM PS_PENDING_ITEM I, PS_PG_REQUEST_TAO R,
     PS_ITEM_ENTRY_TBL E, PS_SET_CNTRL_REC S, PS_BUS_UNIT_TBL_AR B

WHERE R.PROCESS_INSTANCE = %BIND(PROCESS_INSTANCE)
AND R.PGG_GROUP_TYPE = 'B'
AND I.POSTED_FLAG = 'N'

AND R.GROUP_BU = I.GROUP_BU
AND R.GROUP_ID = I.GROUP_ID

AND E.EFF_STATUS = 'A'
AND S.RECNAME = 'ITEM_ENTRY_TBL'
AND S.SETID = E.SETID
AND S.SETCNTRLVALUE = I.BUSINESS_UNIT
AND E.ENTRY_TYPE = I.ENTRY_TYPE
AND E.EFFDT = ( SELECT MAX(EFFDT) FROM PS_ITEM_ENTRY_TBL Z
                WHERE Z.SETID = E.SETID
                  AND Z.ENTRY_TYPE = E.ENTRY_TYPE
                  AND Z.EFF_STATUS = 'A'
                  AND Z.EFFDT <= I.ACCOUNTING_DT )
AND B.BUSINESS_UNIT = I.BUSINESS_UNIT
/

INSERT INTO PS_PG_PENDDST_TAO (...)

SELECT ...

```

```

    ( (I.ENTRY_AMT_BASE - I.VAT_AMT_BASE) * L.DST_LINE_MULTIPLR *
L.DST_LINE_PERCENT / 100 ),
    ( (I.ENTRY_AMT - I.VAT_AMT) * L.DST_LINE_MULTIPLR * L.DST_LINE_PERCENT / 100
),
. . . . .

FROM PS_PENDING_ITEM I, PS_PG_REQUEST_TAO R, PS_ITEM_LINES_TBL L,
     PS_ITEM_ENTRY_TAO E

. . . . .

WHERE

. . . . .

AND L.ENTRY_REASON = I.ENTRY_REASON
AND L.SETID = E.SETID
AND L.ENTRY_TYPE = E.ENTRY_TYPE
AND L.EFFDT = E.EFFDT

. . . . .

AND E.BUSINESS_UNIT = I.BUSINESS_UNIT
AND E.ACCOUNTING_DT = I.ACCOUNTING_DT
AND E.ENTRY_TYPE = I.ENTRY_TYPE
/

```

Platform Issues

Set processing does not necessarily behave the same on every database platform. Coincidentally, on some platforms, set processing can encounter performance breakdowns! Some platforms do not optimize update statements that include sub-queries.

For example, the environments that are accustomed to updates with sub-queries would get all the qualifying Dept IDs from the Department table, and then, using an index designed by an application developer, update the Personnel table. Other platforms would read through every employee row in the Personnel table and query the Department table for each row.

On platforms where these types of updates are a problem, try adding some selectivity to the outer query. In the following example, examine the SQL in the "Before" section, and then notice how it is modified in the "After" section to run smoothly on all platforms. You can use this approach to workaround platforms that have difficulty with updates that include sub-queries.

Note. In general, set processing capabilities vary by database platform. The performance characteristics of each database platform differ with more complex SQL and the set processing constructs discussed in this chapter. Some database platforms allow additional set processing constructs that enable you to process even more data in a set-based manner. In cases where performance needs improvement, you must tailor or tune the SQL for your environment. The document is intended to provide a general foundation for set processing. PeopleSoft assumes that you are familiar with the capabilities and limitations of your database platform and can recognize, through tracing and performance results, the types of modifications you need to incorporate with the basic set processing constructs described.

- Before

```
UPDATE PS_REQ_LINE
SET SOURCE_STATUS = 'I'
WHERE
EXISTS
(SELECT 'X' FROM PS_PO_ITM_STG STG
WHERE
STG.PROCESS_INSTANCE =%BIND (PROCESS_INSTANCE) AND
STG.PROCESS_INSTANCE =PS_REQ_LINE.PROCESS_INSTANCE AND
STG.STAGE_STATUS = 'I' AND
STG.BUSINESS_UNIT = PS_REQ_LINE.BUSINESS_UNIT AND
STG.REQ_ID = PS_REQ_LINE.REQ_ID AND
STG.REQ_LINE_NBR = PS_REQ_LINE.LINE_NBR)
```

- After

```
UPDATE PS_REQ_LINE
SET SOURCE_STATUS = 'I'
WHERE
PROCESS_INSTANCE = %BIND (PROCESS_INSTANCE) AND
EXISTS
(SELECT 'X' FROM PS_PO_ITM_STG STG
WHERE
STG.PROCESS_INSTANCE =%BIND (PROCESS_INSTANCE) AND
STG.PROCESS_INSTANCE =PS_REQ_LINE.PROCESS_INSTANCE AND
STG.STAGE_STATUS = 'I' AND
STG.BUSINESS_UNIT = PS_REQ_LINE.BUSINESS_UNIT AND
STG.REQ_ID = PS_REQ_LINE.REQ_ID AND
STG.REQ_LINE_NBR = PS_REQ_LINE.LINE_NBR)
```

Note. This assumes that the transaction table (PS_REQ_LINE) has a PROCESS_INSTANCE column to lock rows that are “in process.” This is another example of designing your database with batch performance and set processing in mind.

This modification enables the system to limit its scan through PS_REQ_LINE to only those rows that the program is currently processing. At the same time, it enables a more “set-friendly” environment to first scan the smaller staging table and then update the larger outer table.

CHAPTER 4

Using Meta-SQL and PeopleCode

This chapter discusses the following topics:

- Meta-SQL constructs that are unique to Application Engine processing.
- PeopleCode in Application Engine programs.
- Dynamic SQL.

Meta-SQL in Application Engine

Note. The SQL Editor does not validate all of the meta-SQL constructs, such as %Bind and %Select—messages might appear stating these constructs are invalid.

This section discusses these meta-SQL functions:

- %Bind
- %Execute Edits
- %Select
- %SelectInit
- %SQL
- %Table
- %TruncateTable
- %UpdateStats

%Bind

Retrieves a field value from a state record and can be used anywhere in a SQL statement. When executed, %Bind returns the value of the state record field identified within its parentheses.

```
%BIND ( [recordname.] fieldname [, NOQUOTES] [, NOWRAP] [, STATIC] )
```

Parameters**Recordname**

The name of a state record. If you do not specify a particular state record, PeopleSoft Application Engine uses the default state record to resolve the %BIND(fieldname).

Fieldname

The field defined in the state record.

NOQUOTES

If the field specified is a character field, its value is automatically enclosed in quotes unless you use the NOQUOTES option. Use NOQUOTES to include a dynamic table and field name reference, even an entire SQL statement or clause, in an Application Engine SQL action.

NOWRAP

If the field is a date, time, or date-time, the system automatically wraps its value in %datein() or %dateout(), unless you use the NOWRAP option. Therefore, if the state record field is populated correctly, you don't need to be concerned with the inbound references, although you can suppress the "In" wrapping with the NOWRAP modifier inside the %BIND. Furthermore, PeopleSoft Application Engine skips the "In" wrapper if the %BIND(date) is in the select field list of another %Select statement. This is because the bind value is already in the "out" format, and the system "selects" it into another state record field in memory. As such, in this circumstance there is no need for either an "Out" wrapper or an "In" wrapper. For example,

First SQL Action:

```
%Select(date_end)
```

```
SELECT %DateOut(date_end )
```

```
FROM PS_GREG
```

Second SQL Action:

```
INSERT INTO ps_greg
```

```
VALUES(%Bind(date_end))
```

STATIC

The STATIC parameter enables you to include a “hard-coded” value in a reused statement. For %BINDs that contain dynamic SQL, this parameter must be used in conjunction with NOQUOTES for proper execution of a re-used statement.

Example

```

UPDATE PS_REQ_HDR
  SET IN_PROCESS_FLG = %BIND(MY_AET.IN_PROCESS_FLG) ,
      PROCESS_INSTANCE = %BIND(PROCESS_INSTANCE)
  WHERE IN_PROCESS_FLG = 'N'
      AND BUSINESS_UNIT || REQ_ID
      IN (SELECT BUSINESS_UNIT || REQ_ID
          FROM PS_PO_REQRCON_WK1
          WHERE PROCESS_INSTANCE = %BIND(PROCESS_INSTANCE))

```

In the previous example, %BIND(PROCESS_INSTANCE) assigns the value of the field PROCESS_INSTANCE in the default state record to the PROCESS_INSTANCE field in table PS_REQ_HDR.

The %BIND function is also used in a WHERE clause to identify rows in the table PS_PO_REQRCON_WK1, in which the value of PROCESS_INSTANCE equals the value of PROCESS_INSTANCE in the default state record.

Notes about %BIND

- Typically, when you use %BIND to provide a value for a field or a WHERE condition, the type of field in the state record that you reference with %BIND must match the field type of the corresponding database field used in the SQL statement.
- On most platforms, you can't use a literal to populate a LONG VARCHAR field. You should use the %BIND(record.fieldname) construct.
- In the case of an external call to a section in another program, if the called program has its own default state record defined, then PeopleSoft Application Engine uses that default state record to resolve the %BIND(fieldname). Otherwise, the called program inherits the calling programs default state record.
- All fields referenced by a %SELECT command must be defined in the associated state record.
- You *must* use the Date, Time, and DateTime output wrappers in the SELECT list that populates the state record fields. This ensures compatibility across all supported database platforms. For example,

- First SQL Action:

```

%Select (date_end)
  SELECT %DateOut (date_end )
      FROM PS_EXAMPLE

```

- Second SQL Action:

```

INSERT INTO PS_EXAMPLE
  VALUES (%Bind (date_end))

```

Handling Bind Variables and Date Wraps

The behavior of bind variables within PeopleSoft Application Engine PeopleCode and "normal" PeopleCode is exactly the same.

Alternately, if you compare Application Engine SQL to PeopleCode (of any type), then the system processes bind variables differently. There are the following options.

- **Option 1.** If you use the following approach

```
AND TL_EMPL_DATA1.EFFDT <= %P(1))
```

Then in PeopleCode you issue

```
%SQL(MY_SQL, %DateIn(:1))
```

This assumes that you have referenced the literal as a bind variable.

Or in Application Engine SQL you issue:

```
%SQL(MY_SQL, %Bind(date_field))
```

- **Option 2.** Alternately, if you elect to use the following approach:

```
AND TL_EMPL_DATA1.EFFDT <= %datein(%P(1))
```

Then in PeopleCode you issue:

```
%SQL(MY_SQL, :1)
```

This assumes that you have referenced the literal as a bind variable.

Or in Application Engine SQL you issue:

```
%SQL(MY_SQL, %Bind(date_field, NOWRAP))
```

%ExecuteEdits

Enables PeopleSoft Application Engine to support data dictionary edits in batch. It is *Application Engine-only* meta-SQL—you cannot use it in COBOL, SQR, or PeopleCode—not even Application Engine PeopleCode.

```
%ExecuteEdits(<type>, recordname [alias][, field1, field2, ...])
```

where <type> consists of any combination of the following (added together):

```
%Edit_Required
%Edit_YesNo
%Edit_DateRange
%Edit_PromptTable
%Edit_TranslateTable
```

The *recordname* parameter specifies the record used to obtain the data dictionary edits, and the optional list of fields is used to restrict the edits to a subset of the record's fields.

For example, suppose you want to insert rows with missing or invalid values in three specific fields, selecting data from a temporary table but using edits defined on the original application table. Notice the use of an alias or "correlation name" inside the meta-SQL.

```
INSERT INTO PS_JRNL_LINE_ERROR (...)
SELECT ... FROM PS_JRNL_LINE_TMP A
WHERE A.PROCESS_INSTANCE = %BIND(PROCESS_INSTANCE)
      AND %EXECUTEEDITS(%Edit_Required + %Edit_PromptTable, JRNL_LINE A,
BUSINESS_UNIT, JOURNAL_ID, ACCOUNTING_DT)
```

If you need to update rows in a temporary table that have some kind of edit error, you can use custom edits defined on the temporary table record.

```
UPDATE PS_PENDITEM_TAO
SELECT ERROR_FLAG = 'Y'
WHERE PROCESS_INSTANCE = %BIND(PROCESS_INSTANCE)
      AND %EXECUTEEDITS(%Edit_Required + %Edit_YesNo + %Edit_DateRange +
%Edit_PromptTable + %Edit_TranslateTable,
                      PENDITEM_TAO)
```

Notes About %ExecuteEdits

- Consider performance carefully when using this construct. Prompt table and translate table edits have a significant impact, because they involve correlated sub-queries. Run a SQL trace at execution time so that you can view the generated SQL. Look for opportunities where it can be optimized.
- In general, %ExecuteEdits is best used on a temporary table. If you must run this against a "real" application table, you should provide WHERE clause conditions to limit the number of rows to include only those that the program is currently processing. PeopleSoft suggests that you process the rows in the current set all at once rather than processing them row-by-row.
- With %ExecuteEdits you can't use work records in a batch, set-based operation. So, all higher-order key fields used by prompt table edits must exist on the record that your code intends to edit, and the field names must match exactly. For example,

```
%ExecuteEdits(%Edit_PromptTable, MY_DATA_TMP)
```

The record MY_DATA_TMP contains the field STATE with prompt table edit against PS_REGION_VW, which has key fields COUNTRY and REGION. The REGION field corresponds to STATE, and COUNTRY is the higher-order key. For %ExecuteEdits to work correctly, the MY_DATA_TMP record must contain a field called COUNTRY. It's permissible for the edited field (STATE) to use a different name because PeopleSoft Application Engine always references the last key field (ignoring EFFDT).

- Restrict the number *and* type of edits to the minimum required. Don't perform edits on fields that are known to be valid, or that are defaulted later in the process. Also, consider using a separate record with edits defined specifically for batch, or provide a list of fields to be edited.

%Select

This function is required at the beginning of any and all SELECT statements. For example, you need one in the flow control actions and one in the SQL actions that contain a SELECT. The %SELECT function identifies the state record fields to hold the values returned by the corresponding SELECT statement. In other words, you use %SELECT to pass values to the state record buffers.

You use the %SELECT construct to pass variables to the state record, and you use the %BIND construct to retrieve or reference the variables.

```
%SELECT(statefield1[, statefield2]...[, statefieldN])
SELECT field1[, field2]...[, fieldN]
```

The *statefields* must be valid fields on the state record (may be fieldname or recordname.fieldname, as with %BIND) and *fields* must be either valid fields in the FROM tables or hard-coded values.

Consider the following sample statement:

```
%SELECT(BUSINESS_UNIT, CUST_ID)
SELECT BUSINESS_UNIT, CUST_ID
FROM PS_CUST_DATA
WHERE PROCESS_INSTANCE = %BIND(PROCESS_INSTANCE)
```

The following steps illustrate the execution of the previous statement:

1. Resolve Bind Variables. The string %BIND(PROCESS_INSTANCE) is replaced with the value of the state record field called PROCESS_INSTANCE.
2. Select. Execute the SQL SELECT statement.
3. Fetch. A SQL FETCH is performed. If a row is returned, the state record fields BUSINESS_UNIT and CUST_ID are updated with the results. If the Fetch does not return any rows, all fields in the %SELECT retain their prior values.

Note. All fields referenced by a %SELECT command must be defined in the associated state record. Also, aggregate functions always return a row so they always cause the state record to be updated. As such, for aggregate functions there is no difference whether you use %SelectInit or %Select.

%SelectInit

This meta-SQL construct is identical to %SELECT with the following exception—if the SELECT returns no rows, %SelectInit reinitializes the buffers. In the case of a %SELECT and no rows are returned, the state record fields retain their previous values.

Note. PeopleSoft added this construct primarily for backward compatibility to the &SELECT construct of PeopleTools 7.5. The conversion program substitutes %SelectInit for &SELECT so that after conversion the behavior does not change. However, for performance reasons, PeopleSoft recommends using %SELECT if the buffer initialization is not necessary.

Aggregate functions always return a row so they always cause the state record to be updated. As such, for aggregate functions there is no difference whether you use %SelectInit or %Select.

%SQL

When you use %SQL in a statement, PeopleSoft Application Engine replaces it with the specified SQL object. This enables commonly used SQL text to be shared among Application Engine and PeopleCode programs alike. In PeopleSoft Application Engine, you use %BIND to specify your bind variables. In PeopleCode SQL, you can use

```
:record.field
```

or

```
:1
```

If you create SQL objects that you plan to share between PeopleSoft Application Engine and PeopleCode programs, the %SQL construct enables you to pass parameters for resolving bind variables without being concerned with the difference in the bind syntax that exists between Application Engine and PeopleCode. However, keep in mind that the "base" SQL statement that uses %SQL to represent a shared object with binds needs to be tailored to PeopleSoft Application Engine or to PeopleCode.

For example, assume that your SQL is similar to the following:

```
UPDATE PS_TEMP_TBL SET ACTIVE = %BIND(MY_AET.ACTIVE)
WHERE PROCESS_INSTANCE = %ProcessInstance
```

That would not be valid if the SQL executed in PeopleCode. However, if you define your SQL as shown:

```
UPDATE PS_TEMP_TBL SET ACTIVE = %P(1)
WHERE PROCESS_INSTANCE = %ProcessInstance
```

You could use parameters in %SQL to insert the appropriate bind variable.

From PeopleSoft Application Engine, the "base SQL" or source statement might look like the following:

```
%SQL (SQL_ID, %BIND (MY_AET.ACTIVE))
```

The PeopleCode SQL may appear as the following:

```
%SQL (SQL_ID, :MY_AET.ACTIVE)
```

Note. You can use %SQL only to reference SQL Objects created directly in PeopleSoft Application Designer. For instance, you can not use %SQL to reference SQL that resides within a Section in an Application Library. Common SQL should be stored as a proper SQL Object.

%Table

This function returns the SQL table name for the record specified with *recname*. The basic syntax is:

```
%Table (recname)
```

For example,

```
%Table (ABSENCE_HIST)
```

Returns the record PS_ABSENCE_HIST.

If the record is a temporary table and the current process has a temporary table instance number specified, then %Table resolves to that instance of the temporary table PS_ABSENCE_HIST nm , where nm is the instance number.

This function can be used to specify temporary tables for running parallel Application Engine processes.

You can use the %Table function when you want to be able to run the same Application Engine program, in parallel, across different subsets of the data.

%TruncateTable

This construct is functionally identical to a DELETE SQL statement with no WHERE clause, but it is faster on databases that support bulk deletes. If you're familiar with COBOL this construct is an enhanced version of the COBOL meta-SQL construct with the same name.

Some database vendors have implemented bulk delete commands that decrease the time required to delete all the rows in a table by not logging rollback data in the transaction log. For the databases that support these commands, PeopleSoft Application Engine replaces %TruncateTable with "TRUNCATE TABLE" SQL. For the other database types, %TruncateTable is replaced with "DELETE FROM" SQL.

Unlike the COBOL version, PeopleSoft Application Engine determines if a commit is possible prior to making the substitution. If a commit is possible, PeopleSoft Application Engine makes the substitution and then forces a checkpoint and commit after the successful execution of the delete.

If a commit is not possible, PeopleSoft Application Engine replaces the meta-SQL with a DELETE FROM string. This ensures restart integrity when your program runs against a database where there is an implicit commit associated with TRUNCATE TABLE or where rollback data is not logged.

The basic syntax for %TruncateTable is

```
%TruncateTable(table name)
```

For example,

```
%TruncateTable(PS_PO_WRK1)
```

For databases that either execute an implicit commit for %TruncateTable or require a commit before and/or after this meta-SQL, replace %TruncateTable with an unconditional delete in the following circumstances:

- A commit is not allowed, as in within an Application Engine program called from PeopleCode.
- The program issues a non-select SQL statement since the last commit occurred. In such a situation, data is likely to have changed.
- You are deferring commits in a select/fetch loop within a restartable program.

Note. To use a record name as the argument for %TruncateTable (instead of an explicit table name), you must include a %Table () meta-SQL function to resolve the unspecified table name. For example, to specify the record PO_WEEK as the argument, use the following statement:

```
%TruncateTable(%Table(PO_WEEK))
```

%UpdateStats

PeopleSoft Application Engine replaces this meta-SQL construct with a platform-dependent SQL statement that updates the system catalog tables used by the database optimizer in choosing optimal query plans. Peoplesoft intends that you use this construct after your program has inserted large amounts of data into a temporary table that will be deleted before the end of the program run. This saves you from having to use "dummy" seed data for the temporary table and having to update statistics manually.

```
%UpdateStats(record name , [HIGH/LOW])
```

For example,

```
%UpdateStats(PO_WRK1)
```

The default is LOW.

Notes About %UpdateStats

For databases that either execute an implicit commit for %UpdateStats or require a commit before and/or after this meta-SQL, PeopleSoft Application Engine skips %UpdateStats in the following circumstances:

- A commit is not allowed, as in within an Application Engine program called from PeopleCode.
- The program issues a non-select SQL statement since the last commit occurred. In such a situation, data is likely to have changed.
- You are deferring commits in a select/fetch loop in a restartable program. PeopleSoft Application Engine skips %UpdateStats even if the previous condition is false.

The following table shows how the %UpdateStats construct is resolved by the supported database systems.

Database Function	Behavior
MSS %UpdateStats	LOW = UPDATE STATISTICS tablename HIGH => UPDATE STATISTICS tablename WITH FULLSCAN
Sybase %UpdateStats	LOW and HIGH = UPDATE ALL STATISTICS tablename
Oracle %UpdateStats	PeopleSoft now uses DDL templates (in PSDDLMODEL) to determine SQL statements for %UpdateStats. Use DDLDB2.DMS to change. If delivered LOW = ANALYZE TABLE [tablename ESTIMATE STATISTICS; Delivered High => ANALYZE TABLE [TBNAME] COMPUTE STATISTICS;
DB2UNIX %UpdateStats	LOW and HIGH => RUN STATISTICS FOR TABLE
DB2390 %UpdateStats	Now using DDL templates (in PSDDLMODEL) to determine SQL statements for %UpdateStats. Use DDLORA.DMS to change. Delivered LOW => RUNSTATS TABLESPACE [DBNAME].[TBSPCNAME] TABLE ALL SAMPLE 25 INDEX REPORT NO SHRLEVEL CHANGE; Delivered High => RUNSTATS TABLESPACE [DBNAME].[TBSPCNAME] TABLE ALL INDEX (ALL KEYCARD) REPORT NO SHRLEVEL CHANGE;
Informix %UpdateStats	LOW => UPDATE STATISTICS MEDIUM FOR TABLE tablename HIGH => UPDATE STATISTICS HIGH FOR TABLE tablename

Using %UpdateStats With COBOL

You can use the %UpdateStats construct from SQL embedded in your COBOL programs. Use this syntax:

```
%UpdateStats (<tablename>)
```

When you issue this construct from PeopleTools, the parameter is <record name>. Make note of this distinction.

%UpdateStats Considerations

The following table lists potential issues that you might encounter when using %UpdateStats.

Database	Consideration
Microsoft SQL Server Sybase UDB	PeopleSoft forces a commit before and after the %UpdateStats statement. Therefore, the system skips (null op) this meta-SQL if a commit is not allowed. For instance, a commit is not allowed in the following situations: The Application Engine program is running online, as in not in batch mode. You have issued non-Select/Fetch SQL (in which the data is likely to change) since the last Commit. You are deferring Commits in a SELECT/FETCH loop within a restartable program.
Oracle	Oracle has an implicit commit after the %UpdateStats statement executes. Same behavior as previous consideration.
DB2 for OS/390	For %UpdateStats to work correctly on DB2 for OS/390 you must complete a considerable amount of DBA work. For this reason, PeopleSoft recommends disabling %UpdateStats for OS/390.
Informix IBM UDB	%UpdateStats will lock the table being analyzed on UDB and Informix. Therefore, PeopleSoft recommends that this meta-SQL be used only on tables that are not likely to be concurrently accessed by other applications and users. A perfect use thereof would be to analyze AE dedicated temp tables.
All	%UpdateStats consumes an enormous amount of time and database resources if run against very large tables. Therefore, PeopleSoft recommends that permanent data tables be analyzed outside of application programs. Also, if temp tables are likely to grow very large during a batch run, PeopleSoft recommends running the batch program only with %UpdateStats enabled to seed the statistics data or when the data composition changes dramatically.

Disabling %UpdateStats

To disable %UpdateStats, you can do so in the following ways:

- Include the following parameter on the command line when running an Application Engine program:

```
-DBFLAGS 1
```

- Activate the Disable DB Stats check box in the Shared Flags group on the Process Scheduler tab in the Configuration Manager.
- Change the Dbflags=0 parameter in the Process Scheduler configuration file (or PSADMIN) to Dbflags=1.

Application Engine Macros

The section describes Application Engine macros.

%ClearCursor

Use the %CLEARCURSOR function to recompile a re-used statement and reset any STATIC %BINDs.

```
%CLEARCURSOR(program, section, step, action)
```

or

```
%CLEARCURSOR (ALL)
```

When you use the %CLEARCURSOR function keep the following items in mind:

- It must be located at the beginning of the statement.
- It can be the only function or command contained in the statement.
- The action parameter must specify an action that executes SQL.
- Using (ALL) clears all cursors.

For the action parameter, the following table contains the values.

Value	Action Type
D	Do Select
H	Do When
N	Do Until
S	SQL
W	Do While

%Execute

Enables you to execute database-specific commands from within your Application Engine program. Also, the %EXECUTE construct enables you to include multiple statements in a single Application Engine action without encountering database-specific differences. For instance, there are instances where you could code a single Application Engine action to contain multiple SQL statements and they may run successfully on one database platform. However, if you attempt to run the same code against a different database platform you may encounter errors or skipped SQL.

```
%EXECUTE ([/])
command1{; | /}
command2{; | /}...
commandN{; | /}
```

By default, PeopleSoft Application Engine expects a semicolon to be used to delimit multiple commands within an %EXECUTE function statement. You can instruct PeopleSoft Application Engine to use a forward slash (/) delimiter instead by placing a forward slash inside the function parentheses.

For example, the following code enables you to use an Oracle PL/SQL block in an %EXECUTE statement:

```
%EXECUTE (/)
DECLARE
  counter INTEGER;
BEGIN
  FOR counter := 1 TO 10
    UPDATE pslock SET version = version + 1;
  END FOR;
END;
/
```

Note. When you use the %EXECUTE function, it must be located at the beginning of the statement and can be the only function or command contained in the statement. The action type must be “SQL”.

%Next and %Previous

This is valid in any Application Engine SQL action, and should be used when performing sequence-numbering processing. Typically, you use it in place of a %Bind. These constructs use the current value of the number field as a Bind variable, and then increment (%Next) or decrement (%Previous) the value *after* the statement is executed successfully. By “number” field, we are referring to the numeric field on the state record that you have initially set to a particular value (as in ‘1’ to start).

If the statement is a SELECT and no rows are returned, the field value is not changed. The substitution rules are the same as for %Bind. For example, if re-use is enabled, then the field is a true bind (:n substituted). Otherwise, inline substitution occurs.

For example, you could use these constructs in an UPDATE statement within a Do Select.

Step 1

- Do Select

```
%SELECT(field1, field2, ...) SELECT key1, key2, ... FROM PS_TABLE WHERE ...
ORDER BY key1, key2, ..."
```

- SQL

```
UPDATE PS_TABLE SET SEQ_NBR = %NEXT(seq_field) WHERE key1 = %BIND(field1) AND
key2 = %BIND(field2) ...
```

With a Do Select the increment/decrement occurs once per execution, not once for every fetch. So unless your Do Select is of the "Reselect" type, the value is changed only on the *first* iteration of the loop. Alternatively, with Reselect or Do While/Until actions, every iteration re-executes the Select and then fetches one row, and with these types of loops, the value changes on *every* iteration.

%RoundCurrency

This meta-SQL is an enhanced version of the Application Engine &ROUND construct that appeared in previous releases. The %ROUNDCURRENCY function rounds an amount field to the currency precision specified by the field's **Currency Control Field** property—as defined in the PeopleSoft Application Designer Record Field Properties dialog box. For this function to work, you must have the **Multi-Currency** option selected in the PeopleTools Options panel. As with all Application Engine macros described in the current section, this construct is valid only in Application Engine SQL; it is not valid for SQLExecs or view text.

```
%RoundCurrency( <EXPRESSION>, [ALIAS.]<CURRENCY_FIELD>)
```

You can use this macro in the SET clause of an UPDATE statement or the SELECT list of an INSERT/SELECT statement. The first parameter is an arbitrary expression of numeric values and/or columns from the “source” tables that computes the monetary amount to be rounded. The second parameter is the control currency field from a particular “source” table (the UPDATE table, or a table in the FROM clause of an INSERT/SELECT statement). This field identifies the corresponding currency value for the monetary amount.

Note. Remember that the “As Of Date” of the Application Engine program is used for obtaining the currency-rounding factor. The currency-rounding factor is determined by the value of DECIMAL_POSITIONS on the corresponding row in PS_CURRENCY_CD_TBL, which is an effective-dated table.

If multi-currency is not in effect, then the result are rounded to the precision of the amount field (either 13.2 or 15.3 amount formats are possible).

For example,

```
UPDATE PS_PENDING_DST
  SET MONETARY_AMOUNT =
    %RoundCurrency( FOREIGN_AMOUNT * CUR_EXCHNG_RT, CURRENCY_CD)
  WHERE GROUP_BU = %Bind(GROUP_BU) AND GROUP_ID = %Bind(GROUP_ID)
```

Application Engine System (Meta) Variables

In addition to the meta-SQL and macros, there are some text-substitution variables—similar to meta-SQL—that are unique to PeopleSoft Application Engine.

%AeProgram Returns a quoted string containing the currently executing Application Engine program name.

%AeSection	Returns a quoted string containing the currently executing Application Engine Section name.
%AeStep	Returns a quoted string containing the currently executing Application Engine Step name.
%AsOfDate	Returns a quoted string containing the “AsOfDate” used for the current process.
%Comma	Returns a comma. This is useful in those cases where you must use a comma, but commas are not allowed due to the parsing rules. For example, you might use this if you wanted to pass a comma, as a parameter, to the %SQL meta-SQL function.
%JobInstance	Returns the numeric (unquoted) Process Scheduler Job Instance.
%LeftParen	Returns a left parenthesis. Usage is similar to %Comma.
%List	<p>Note. %List is not Application Engine specific, however, there are some restrictions when using %List in your Application Engine programs.</p> <p>The %List function expands into a list of field names, delimited by commas. Which fields are included in the expanded list depends on the parameters passed to the function.</p> <p>PeopleSoft has restricted the use of %List within your Application Engine programs: when using %List in an insert/select or insert/values or %Select statement, you must have matching pairs of %List (or %ListBind) in the target and source field lists, using the same list type argument and record name to ensure consistency.</p>
%ProcessInstance	Returns the numeric (unquoted) Process Instance.
%ReturnCode	Returns the numeric (unquoted) return code of the last SQL operation performed.
%RightParen	Returns a right parenthesis. Usage is similar to %Comma.
%RunControl	Returns a quoted string containing the current Run Control identifier. The Run Control ID is available to your program, when using %RunControl, regardless of whether there's a row in the AEREQUEST table.
%Space	Returns a single space. Usage is similar to %Comma.
%SQLRows	Can be used in any Application Engine SQL statement, but the underlying value is affected only by SQL Actions. It is not affected by Do When, Do Select, Do While, and Do Until. For SELECT statements, the value can only be 0 or 1: row not found or rows found, respectively. It does not reflect the actual number of rows that meet the WHERE criteria. In order to find the number of rows that

meet the WHERE criteria, you must code a SELECT COUNT(*).

Using PeopleCode in Programs

Inserting PeopleCode within Application Engine programs enables developers to reuse common function libraries and improve performance. In many cases, a small PeopleCode program used instead of Application Engine PeopleCode is an excellent way to build dynamic SQL, perform simple IF/ELSE edits, set defaults, and perform other tasks that don't require a trip to the database. There are appropriate and inappropriate uses of the PeopleCode within PeopleCode Actions.

See Also

PeopleTools 8.4 PeopleBook: PeopleCode Reference, “PeopleCode Built-in Functions,” Application Engine.

When to Use PeopleCode

Application Engine is *not* intended to execute programs that include nothing but PeopleCode actions— its primary purpose is still to run SQL against your data.

Because PeopleCode is an interpreted language, there is an inherent extra performance overhead when compared to a compiled language. For the most part, PeopleSoft recommends that you use PeopleCode for setting IF, THEN, ELSE logic constructs, performing data “preparation” tasks, and building dynamic portions of SQL, while still relying on SQL to complete the bulk of the actual program processing. We also expect that you will use PeopleCode as a way to reuse online logic that’s already developed. And, of course, PeopleCode is the tool for taking advantage of the new technologies, such as Component Interface and Business Interlinks.

Most programs need to check that a certain condition is TRUE prior to executing a particular Section. For example, if the hourly wage is less than or equal to ‘X’, do Step A; if not, fetch the next row. And, in certain instances, you need to modify variables that exist in a state record. PeopleCode enables you to set state record variables dynamically.

Environment Considerations

When writing or referencing PeopleCode in a PeopleCode action, you must consider the environment in which the Application Engine program runs. By environment, we are not referring to your database and workstation configurations but rather the differences between online and batch modes. It’s important to keep in mind that Application Engine programs usually run in batch mode, and, consequently, your PeopleCode cannot access pages or controls as it can while running in online mode. In short, any PeopleCode operations that manipulate pages will not run successfully. Even if you invoke your Application Engine program "online" from a record or a page using the CallAppEngine PeopleCode function, the Application Engine PeopleCode still does not have direct access to the page buffers.

Note. Application Engine programs cannot access page buffers.

Any RECORD.FIELD references that appear in a PeopleCode action can refer only to fields that exist on an Application Engine state record. Page buffers, controls, and so on are still inaccessible even if you define the page records as state records on the Program Properties dialog box. An Application Engine program can access only state records or other objects you create in PeopleCode.

However, PeopleSoft realizes that in many situations it is necessary to pass data from a page buffer to an Application Engine program. If your program requires this functionality, you have the following options.

Passing Parameters Through CallAppEngine

For individual page fields and simple PeopleCode variables, such as numbers and strings, you can use the CallAppEngine PeopleCode function to pass values as parameters. You can do this by performing the following actions:

- Declare a record object in PeopleCode, as in


```
Local Record &MyRecord;
```
- Assign the record objects to any state record that you want to pass to the Application Engine program. Record objects are parameters to the CallAppEngine function.
- Set the appropriate values on that state record.
- Include the record object in the function call.

After these values get set in the state record, all the actions in a particular program can use the values, not just the PeopleCode actions.

Defining Global Variables

You can also define global variables or objects in the page PeopleCode before calling an Application Engine program. Application Engine PeopleCode actions only are able to access the variables you define, however, the PeopleCode could set a state record field equal to a number/string variable for use by other Application Engine actions.

Also, an Application Engine PeopleCode program can read or update a page scroll using a global rowset object. When accessing a page scroll from Application Engine PeopleCode, the same rules apply and the same illegal operations are possible that you would see with accessing record or page PeopleCode.

The parameters submitted in a CallAppEngine will be "by value." These parameters "seed" the specified Application Engine state record field with the corresponding value. If that value is changed within PeopleSoft Application Engine by updating the state record field, the page data will not be affected. The only way to update page buffers or "external" PeopleCode variables from Application Engine is to use global PeopleCode variables and objects.

State Records

Executing PeopleCode from Application Engine steps enables you to complete some simple operations without having to use SQL. For example, to assign a literal value to an Application Engine state record field using SQL you may have issued a statement similar to the following:

```
%SELECT (MY_AET.MY_COLUMN)
SELECT 'BUSINESS_UNIT' FROM PS_INSTALLATION
```

You can use a PeopleCode assignment instead.

```
MY_AET.MY_COLUMN = "BUSINESS_UNIT";
```

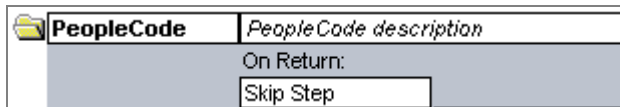
Similarly, you can use a PeopleCode IF statement instead of using a DO When to check the value of a state record field.

When accessing state records with PeopleCode, keep the following in mind:

- State records are records unique to an Application Engine program.
- Within Application Engine PeopleCode, state record values can be accessed and modified using the standard “record.field” notation.

IF, THEN Logic

From PeopleCode, you can trigger an error status, or false return, by using the EXIT(1) function. Use the **On Return** value on the PeopleCode action properties to specify how your Application Engine program behaves according to the return of your PeopleCode program.



On Return action property

By default, an abort will occur, similar to what happens when a SQL error occurs. But by changing this to *Skip Step* you can control the flow of your Application Engine program.

You can use EXIT() to add an IF condition to a Step or a Section Break. For example,

```
If StateRec.Field1 = 'N'
Exit (1);
Else
/* Do processing */
End-if;
```

You must specify a non-zero return value to trigger the On Return action. The concepts of “return 1” and “return True” are equivalent. So, if the return value is non-zero or True, then Application Engine will perform what you specify for On Return, as in Abort or Skip Step.

However, if the program returns zero or False, Application Engine ignores the selected On Return value.

Scope of Variables

The following table presents the different types of variables typically used in Application Engine programs and their scope.

<i>Type of variable</i>	<i>Scope</i>	<i>Comments</i>
State Record (Work record)	Transaction (Unit of Work)	Using a work record as your Application Engine state record means that the values in the Work record cannot be committed. Commits will happen as directed, but any values in Work records are not retained after a commit.
State Record (“Real” record)	Application Engine Program	Using a “real” record as your Application Engine state record will preserve the values in the state record on commit, and the committed values are available in the event of a restart.
Local PeopleCode variables	PeopleCode Program	Local PeopleCode variables are available only for the duration of the PeopleCode program that is using them.
Global PeopleCode variables	Application Engine Program	Global PeopleCode variables are available during the “life” of the program that is currently running. Any global PeopleCode variables are saved when an Application Engine program commits and checkpoints, and therefore they will be available in the event of a restart.
Component PeopleCode variables	Application Engine program.	Acts like global variables to PeopleSoft Application Engine.

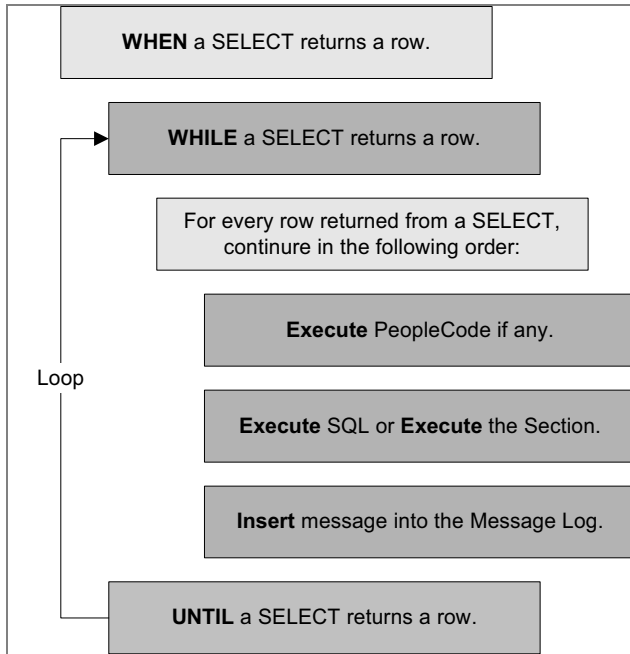
Action Execution Hierarchy

No other types of actions are required within a step in conjunction with a PeopleCode action (or program). So, you can have a step that contains nothing but one PeopleCode action. If you include other actions with your PeopleCode action within the same step, it’s important to keep in mind the execution hierarchy.

With PeopleCode actions, Application Engine executes the PeopleCode program *before* the SQL, Call Section, or Log Message actions, but a PeopleCode program executes *after* any program flow checks.

Because there are multiple action types, they must execute in agreement within a system, and therefore the order in which action’s execute is significant. At runtime, actions defined for a given step are evaluated based on their action type. All of the action types exist within a strict hierarchy of execution. For example, if both a Do When and PeopleCode action exist within a given step, the Do When is always executed first.

The following example shows the sequence and level of execution for each type of action:



Action execution hierarchy

Using PeopleCode in Loops

You can insert PeopleCode inside of a “Do” loop, but using PeopleCode inside of high volume “Do” Loops (While, Select, Until) should be made with care. PeopleSoft recommends keeping the number of distinct programs inside the loop to a minimum. As stated previously, you should avoid having PeopleCode performing the actual “work” of the program and instead use it primarily in your Application Engine programs to control the flow (IF, THEN logic), build dynamic SQL, or interact with external systems.

Using bind variables instead of literals to pass values to SQL statements is *essential* if the PeopleCode loops or if the PeopleCode gets called in a loop. If the PeopleCode loops, there is a good probability that PeopleSoft Application Engine will use a dedicated cursor, which saves the overhead of recompiling the SQL for all iterations. If the PeopleCode gets called from within a loop, PeopleSoft Application Engine does not reduce the number of compiles, but Application Engine avoids flooding the SQL cache (for those database servers that support SQL cache) when it uses bind variables. Do not use bind variables for values in a SELECT list or SQL identifiers such as table and column names, as some databases do not support this.

Note. NULL datetime/date/time Bind values are always resolved into literals.

On those database platforms for which PeopleSoft has implemented this feature, Setting BulkMode to TRUE often results in significant performance gains when inserting rows into a table in a Loop.

In general, avoid having PeopleCode calls within a loop. The “startup and teardown” costs of the PeopleCode interpreter will be multiplied. If you can call the PeopleCode outside of the loop, use that approach. Not only is it just “good” programming, but also it increases the overall performance with an interpreted language such as PeopleCode.

AESection Object

The AESection PeopleCode object enables you to change the properties of an Application Engine program Section dynamically without having to modify any of the Application Engine tables directly. This enables you to develop “rule-based” applications that conform dynamically to variables that an end user submits through a page, such as the Application Engine Request page.

The AESection object provides developers the following flexibility:

- Portions of SQL determined by checks prior to execution.
- The logic flow conforms as rules change, and the program adjusts to the rules.

When using the AESection object keep the following in mind:

- When you consider using the AESection object, you should first check to make sure that you primarily require dynamic capabilities with the SQL your program generates. Also, make sure that the rules to which your program conform are relatively static or at least defined well enough such that a standard template could easily accommodate them.
- Consider using the SQL Repository to create dynamic SQL for your programs to avoid the complexity of the AESection object using the StoreSQL function
- The AESection object is designed to dynamically update only SQL-based Actions, not PeopleCode, Call Section, and so on. You can *add* a PeopleCode action to your generated section, but you can not alter the PeopleCode.
- The AESection object is designed for use for online processing. Typically, the dynamic sections should be constructed in response to an end-user action.

Note. Do not call an AESection object from an Application Engine PeopleCode Action.

Online Application Engine Calls (CallAppEngine)

To make synchronous online calls to an Application Engine program, you must use the PeopleCode function CallAppEngine.

Note. Be aware, however, that this means the user can't perform another PeopleSoft task until the Application Engine program completes. Consequently, it's very important to consider the size and performance of the Application Engine program called by CallAppEngine. You need to be sure that the program will run to successful completion consistently within an acceptable amount of time.

If an Application Engine program called by CallAppEngine abends, the end user receives an error, similar to other save time errors, that forces the user to cancel the operation. The CallAppEngine function returns a value based on the result of the Application Engine call. If the program was successful it returns a zero, and if the program was unsuccessful, it returns a non-zero.

CallAppEngine Built-in Function

Application Engine supports the CallAppEngine built-in function, either directly through a PeopleCode action, or indirectly through a component interface. Use this function carefully, however, based on a clear understanding of the following rules and restrictions.

- There are no dedicated cursors supported inside a nested Application Engine instance (meaning an Application Engine program called by CallAppEngine from within another Application Engine program). If a nested Application Engine instance has any SQL actions with ReUse = Yes or Bulk Insert, those settings are ignored.
- As with CallAppEngine in any other type of PeopleCode event, no commits are performed within the called Application Engine program. This is an important consideration. If a batch Application Engine program called another program with CallAppEngine, and the child program updated many rows of data, the unit-of-work might become too large, resulting in contention with other processes. A batch Application Engine program should call other programs with a Call Section action, *not* CallAppEngine.
- Temp tables are not shared between a batch Application Engine program and child program called by CallAppEngine. Instead, the child program is assigned an "online" temporary table instance, which is used for all temp tables in that program. In addition, if that child program calls another program with CallAppEngine, the grandchild program *will* share the online temp instance with the calling program. In other words, only one online temp instance is allocated to a process at any one time, no matter how many nested programs there might be.
- Note that a batch Application Engine program that issues a CallAppEngine() function, will not be able to share any temp table data with the called process, and the called program must be function as an online Application Engine program even if it only runs in batch mode.
- The lock on an online temp instance persists until the next commit. If the processing time of the called program is significant (greater than a few seconds), this would be unacceptable.

Events to Trigger CallAppEngine

You need to include the CallAppEngine PeopleCode function within events that allow database updates. This includes the following PeopleCode events:

- SavePreChange (Record)
- SavePostChange (Page)
- Workflow
- Subscription (Message)
- FieldChange

Process Instance

The Process Instance value is always zero for programs initiated with CallAppEngine. This is because the program called with CallAppEngine runs “in process,” that is, it runs within the same unit of work as the component with which it is associated. There is no separate executable involved like when you call a COBOL or SQR program. As long as you delete from all the tables at the end of the Application Engine program, the Process Instance value can remain zero.

Note. There is no run control ID associated with a program called by CallAppEngine.

Process Instances cannot be assigned to Application Engine programs called using CallAppEngine. To assign a Process Instance, you must update a one-row table. The update won't be committed until the very end, so this means all users will be single-threaded through this task.

The only commit occurs after your Application Engine program completes and all the Save PeopleCode has finished. As a result, any other users running the same page simultaneously have to wait until the first user's save gets committed. After that, the temporary tables are empty and the next user's save starts.

Note. If you anticipate scalability problems due to temporary table contention, the solution is to use the multiple temporary table instances feature and reference them dynamically using the %Table meta-SQL construct.

Save Event Details

If you want to execute the Application Engine program based on an end user Save, then use the CallAppEngine function within a Save event. When you use CallAppEngine, you should keep the following items in mind:

- No commits will occur during the entire program run.
- During SavePreChange, any modified rows in the page have not been written to the database.

- During SavePostCange, the modified rows have been written to the database. The system issues one commit at the end of the Save cycle.

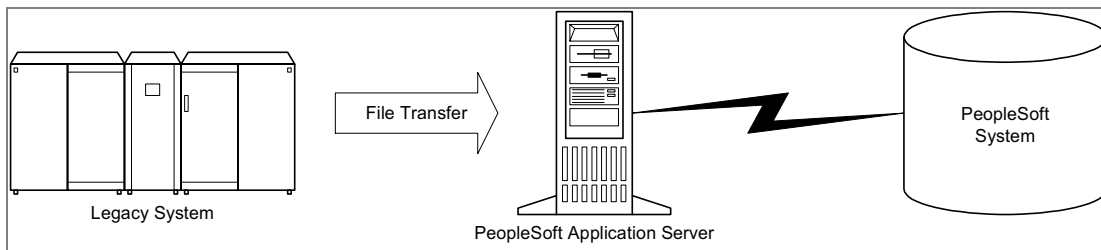
FieldChange Event Details

If you don't want the CallAppEngine call to depend on a Save event, you can also trigger CallAppEngine from a FieldChange event. When having a FieldChange event trigger CallAppEngine, keep the following items in mind:

- Since no commits occur within the CallAppEngine, the called program will remain a synchronous execution in the same unit of work.
- The normal FieldChange commit occurs, which frees any locks that the Application Engine program might have acquired.
- For performance reasons, PeopleSoft requires FieldChange PeopleCode that performs a CallAppEngine to be run on the server for three-tier clients.
- Do not include a DoSave or DoSaveNow function in the same FieldChange event. This is not allowed, and it indicates that you should be including the CallAppEngine within a Save event.

File Layout Object

The file layout object enables you to perform file input/output operations with PeopleSoft Application Engine using PeopleCode. With the File object, you can open a file (for reading or writing), read data from a file, or write data to it. Using the combination of the File Object and PeopleSoft Application Engine provides an effective method to integrate (or exchange) the data stored in a legacy system with your PeopleSoft system. The File Object facilitates the creation of a flat file that both your legacy system and Application Engine programs support.



System Integration using the File Object/Application Engine combination

In the previous example, the Application Engine program running on the application server uses the File object to read the file sent from the legacy system and “translate” it so that it can update the affected PeopleSoft application tables. For the PeopleSoft system and the legacy system to interoperate, you need to first construct a File Object that is compatible for both systems to insert and read data.

PeopleSoft recommends that you attain Rowset and Record access by way of a file using a File Layout definition. You create the File Layout Definition in PeopleSoft Application Designer, and it acts as a template for the file that both systems read from and write to. This

greatly simplifies reading, writing, and manipulating complex transaction data with PeopleCode.

Generally, PeopleSoft recommends using the File Object/Application Engine combination in situations where you can't implement the PeopleSoft Integration Broker solution.

Calling COBOL Modules (RemoteCall)

Using the PeopleCode RemoteCall function, you can call, or execute, COBOL modules from a PeopleCode action. Mainly, PeopleSoft intends this option to support existing Application Engine programs that call COBOL modules; and, to be used by Application Engine developers who are upgrading Application Engine programs from previous releases.

The following sections explain the options involved with Remote Calls from Application Engine programs.

PTPECOBL

This is an interface program is a PeopleSoft executable that enables you to invoke your called COBOL module and pass it required values. You code the RemoteCall to invoke PTPECOBL, which, in turn, calls the specified COBOL module.

If you use PTPECOBL, you don't have to write your own executable to process this task. However, be aware that PTPECOBL does not perform any SQL processing other than retrieve a list of state record values. Consequently, if your current logic requires prior SQL processing you may want to write your own EXE to call your COBOL module. In most situations, PTPECOBL saves you from having to write a custom EXE to handle each call to a GNT.

PTPECOBL performs the following tasks:

- Initializes the specified state record in memory.
- Invokes the COBOL module specified in your PeopleCode.
- Submits any required parameters to the called COBOL module.
- Updates the state record as necessary, issues a commit, and then disconnects from the database after your program completes.

Note. While your COBOL program runs, it can access and return values to the state record.

Sharing Values Between PeopleSoft Application Engine and COBOL

Note the following options for sharing values between the Application Engine program and your called COBOL program:

- **Use state records.** If you add field names Application Engine allows you to pass state record values to the called COBOL program and to get changes passed back to the calling PeopleCode program. If you pass the state record values in this manner use PTPECACH to retrieve and update values just as PTPEFCNV does.

- **Code custom SQL.** If you do not pass the initial values using state record fields you need to insert the appropriate SQL in your called COBOL module to retrieve the appropriate values. Then, to return any updated values to the calling Application Engine program, you must insert the appropriate SQL into a PeopleCode program.

If your COBOL program needs values that do not appear in a state record field, then you can pass PeopleCode variables and values. These variables and values would then be retrieved and/or updated by calling PTPNETRT from within your COBOL program.

- **Create custom EXE.** If you opt to include extra SQL processing and use non-state record values, for consistency purposes, it might be a better approach to create a custom EXE to handle your needs. This way you can call your program directly and have it perform all the PTPNETRT processing. Remember that a RemoteCall command can only call an executable program, not a GNT.

Syntax and Parameters

The following example shows a sample RemoteCall() issued from an Application Engine PeopleCode action to a COBOL module.

```
RemoteCall ("PSRCCBL", "PSCOBOLPROG", "PTPECOBL",
"AECOBOLPROG", "MY_GNT",
"STATERECORD", "MY_AET",
"PRCSINST", MY_AET.PROCESS_INSTANCE,
"RETCODE", &RC,
"ERRMSG", &ERR_MSG,
"FIELD1", MY_AET.FIELD1, "FIELD2", MY_AET.FIELD2);
```

PSRCCBL	This is the Remote Call “dispatcher.” It executes the specified COBOL program using the connect information of the current operator.
PSCOBOLPROG	Specify the name of the COBOL program to run. In this case it is PTPECOBL.
PTPECOBL	This is the parameter that makes the Remote Call from PeopleSoft Application Engine distinct from a normal Remote Call. When you enter this parameter, you, in effect, enable the following parameters, some of which are required.
AECOBOLPROG	Required. Specify the name of the COBOL module you’re calling.
STATERECORD	Required. Specify the appropriate state record that your Application Engine program will share with your COBOL module. PTPECOBL then reserves space in memory for all of the fields on the state record regardless of whether or not they will ultimately store values for processing.
PRCSINST	Required. Specify the state record and Process Instance field. This retrieves the current Process Instance value that

appears on the state record and submits it to your COBOL module using PTPECOBL.

RETCODE, ERRMSG

(Optional) Include RETCODE if you need to return information about any potential problems that the COBOL processing encountered, or use it if your Application Engine program must know whether it completed successfully.

Fieldnames and Values

This is where you specify any fields on the state record that contain initial values for your COBOL module. The quoted field names you specify *must* exist on the specified state record. The corresponding value can be a PeopleCode variable, a record.field reference, or a hardcoded value.

Commit and RemoteCall()

Note the following using RemoteCall() and an Application Engine program:

- The called COBOL module executes as a separate unit of work.
- PeopleSoft recommends that you execute a commit in the step immediately preceding the step containing the RemoteCall PeopleCode action and also in the step containing the Remote Call PeopleCode action. This enables the COBOL process to recognize the data changes made up to the point that it was called, and it also minimizes the time when the process might be in a non-restartable state.
- If you insert *any* SQL processing into your COBOL module, *you* must make sure to commit any updates made by your module. PTPECOBL does not issue any commits.
- If the intent of your COBOL process is to update the value of a passed state record field, then the calling Application Engine PeopleCode is responsible for ensuring that the state record field has been modified, and then the Application Engine program is responsible for committing the state record updates.
- Consider how your COBOL module will react in the event of a restart. Because the work in COBOL will have already been completed and committed will your module ignore a duplicate call or be able to undo/redo the work multiple times? This is similar to issues faced when you execute a Remote call from PeopleCode.
- Typically, when a COBOL program updates the database and then disconnects or terminates without having issuing an *explicit* commit or rollback, an *implicit* rollback occurs. So, without an explicit commit, the database will not retain any updates.

PeopleTools APIs

You can call all of the PeopleTools APIs from an Application Engine program. Keep the following items in mind when using APIs:

- All the PeopleTools APIs contain a Save() method. However, when you call an API from your Application Engine program, regardless of the API's Save() method, the data does not actually get saved until the Application Engine program issues a commit.
- If you've called a component interface from an Application Engine program, all the errors related to the API get logged in the PSMessages collection associated with the current session object.
- If you've published a message, the errors get written to the Message Log and the Application Engine message log.
- If an Application Engine program called from Message Subscription PeopleCode encounters errors and the program exits (with Exit (1)), the error is written to the Message Log and will be marked as "Error."

CommitWork

This function commits pending changes (inserts, updates, and deletes) to the database. Keep the following in mind when using CommitWork.

- This function applies only to a batch Application Engine program. If the program is invoked by CallAppEngine, the CommitWork is ignored. The same is true for commit settings at the section or step level.
- This function can be used only in an Application Engine program that has restart *disabled*. If you try to use this function in a program that doesn't have restart disabled, you receive a runtime error.
- The CommitWork function is useful only when you are doing row-at-a-time SQL processing in a single PeopleCode program, and you need to commit without exiting the program. In a typical Application Engine program, SQL commands are split between multiple Application Engine Actions that fetch, insert, update, or delete application data. Therefore, you would use the section or step level commit settings to manage the commits.

Notes on Various PeopleCode Objects and Functions

The following topics provide some examples of common ways that developers can utilize PeopleCode within Application Engine programs.

Do When

Instead of a Do When that checks a %BIND value you can use PeopleCode to perform the equivalent operation. For example, suppose the following SQL exists in your program:

```
%SELECT (EXISTS) SELECT 'Y' FROM PS_INSTALLATION WHERE %BIND (TYPE) = 'X' ,
```

Using PeopleCode, you could insert the following code:

```
If TYPE = 'X' Then
```

```

Exit(0);
Else
Exit(1);
End-if;

```

If you set the **On Return** parameter on the PeopleCode Action properties to *Skip Step*, this will behave the same as the Do When. The advantage of using PeopleCode is that there is no trip to the database

Dynamic SQL

If you have a select statement that populates a text field with dynamic SQL:

```

%SELECT (AE_WHERE1)
SELECT 'AND ACCOUNTING_DT <= %Bind(ASOF_DATE) '

```

you can use this PeopleCode:

```

AE_WHERE1 = "AND ACCOUNTING_DT <= %Bind(ASOF_DATE)";

```

Sequence Numbering

If you typically use SELECT statements to increment a sequence number inside of a Do Select, While, or Until loop, you can use the following PeopleCode instead:

```

SEQ_NBR = SEQ_NBR + 1;

```

Using PeopleCode rather than SQL is significant. Because the sequencing task occurs repeatedly inside a loop, the cost of using a SQL statement to increment the counter increases with the volume of transactions your program processes. When you are modifying a program to take advantage of PeopleCode, the areas of logic you should consider are those that start with Steps that are executed inside a loop.

Note. You can also use the meta-SQL constructs %NEXT and %PREVIOUS when performing sequence numbering. Using these constructs may help performance in both PeopleCode and SQL calls.

Rowsets

You can use Rowsets in your Application Engine PeopleCode. However, keep in mind that using Rowsets means you'll be using PeopleCode to handle more complicated processing, which will degrade performance.

Math Functions

PeopleSoft recommends using the math functions that your database offers whenever possible.

Internally, PeopleCode assigns types to numeric values. Calculations for the decimal type are processed in arrays to ensure decimal point uniformity across hardware and operating system environments. This is much slower than calculations for type integer, which are processed at the hardware level.

When PeopleCode converts strings to numeric values, it does so using the internal decimal type. So, for performance reasons avoid doing calculations using these values.

A third type of numeric value is the Float type. It is not used as frequently for the following reasons:

- Constants are never stored as Float types in the compiled code. For example, 2.5 will always be Decimal.
- The only way to produce a Float value is by using built-in functions, such as Float() or the Financial math functions.

The Float type is used to produce a Float result only if all operands are also of the Float type. Float operations occur at the hardware level.

Be aware that PeopleCode does not offer optimum performance when processing non-Integer, non-Float math calculations. If you need to perform calculations with these numeric types, consider allowing the database to perform the calculations in COBOL.

SQL Objects

Instead of using SQL objects within PeopleCode, PeopleSoft recommends having Application Engine issue the SQL and using a Do Select that loops around sections containing PeopleCode actions.

It might *appear* easier to code all of the logic within a single PeopleCode program, but splitting the logic into smaller pieces is preferable because you will have better performance, and you get a finer granularity of commit control. You can't cause a commit to occur within a PeopleCode program, but you can issue a commit between Application Engine Steps.

Note. You can't keep global SQL Objects active during a commit, or "across" a checkpoint. PeopleCode behavior does not allow it due to a variety of potential restart issues. If you attempt to access a SQL Object that was open during a checkpoint/commit, you receive an error. In a complex program containing multiple units of work(multiple commits), you should be using Application Engine SQL Actions with ReUse enabled to manage persistent database cursors.

Arrays

PeopleSoft recommends that you avoid using arrays in your Application Engine PeopleCode. In situations where you are tempted to use an array, consider using temporary tables for storing pertinent/affected data. Using temporary tables also lends itself to set-based processing for which PeopleSoft Application Engine is most suited.

Dynamic SQL

Typically, developers include dynamic constructs in their Application Engine programs to change SQL based on various runtime factors or on user-defined input entered through a page.

There are a variety of ways to include dynamic SQL in your Application Engine programs. For example, you could use:

- Dynamic sections, using the AESection object.
- Changing SQL using the SQL object.
- References to SQL in your own tables because we support longs.

The AESection object is primarily designed for online section building, and therefore won't be the most frequently used solution.

PeopleSoft recommends that developers use the SQL Object to store their SQL in the repository. Then if you have a few SQL statements to execute, generate the SQLIDs based on some methodology, such as a timestamp, and then store these in a table.

When the program runs, your SQL could query this table based on "process" and extract the appropriate SQLIDs to be executed with a SQL Action in a DO SELECT loop.

```
%SQL(%BIND(MY_SQLID, NOQUOTES))
```

For a "dynamic" Do, the AE_APPLID and the AE_SECTION fields, must appear on the default state record.

CHAPTER 5

Managing Application Engine Programs

This chapter discusses how to:

- Run Application Engine programs.
- Debug Application Engine programs.
- Restart Application Engine programs.
- Cache Application Engine server.
- Manage abends.

Running Application Engine Programs

You can execute your Application Engine program in one of the following modes: batch using Process Scheduler, online using a PeopleCode function, and manually using the command line. The following table lists some differences between online and batch:

Online Execution	Batch Execution
Started by CallAppEngine from PeopleCode.	Started through PeopleSoft Process Scheduler.
Program runs quickly, synchronously, and at random times.	Programs run for longer amounts of time, asynchronously, and at scheduled times.
Potential for simultaneous execution.	Can be designed for parallel execution for performance.
Uses the <i>Online</i> Temporary Table pool.	Uses the <i>Batch/Dedicated</i> Temporary Table pool.

Batch Using PeopleSoft Process Scheduler

This is the most typical mode of execution. You invoke programs that run in this mode using PeopleSoft Process Scheduler or the Application Engine Process Request Page. Batch mode is also referred to as *asynchronous* execution meaning that it runs independently in the background. PeopleSoft Application Engine runs on any operating system that PeopleSoft supports as an application server. If your site uses an operating system that is not supported for PeopleSoft Application Engine, you must run your Application Engine programs on the application server. (The only exception is OS/390 (MVS)).

To run Application Engine programs on the batch server, you must install Tuxedo. This applies to both UNIX and Windows NT batch servers. If you run your batch server on the same server machine as your application server, then the application server and the batch server can share one Tuxedo install. If your batch server is separate from your application server, then you must install the Tuxedo CD-ROM to your batch server.

TOOLBINSRV Parameter The TOOLBINSRV parameter in the PeopleSoft Process Scheduler configuration file determines where Process Scheduler invokes an Application Engine program. For high-volume batch environments, PeopleSoft recommends specifying the PS_HOME\bin\server\winx86 directory that exists on the same machine where the Application Engine program runs.

Online Using PeopleCode

Application Engine programs that execute online, typically get executed from a page with the CallAppEngine PeopleCode function. Such online processes are *synchronous* meaning that subsequent processes wait on the results. For instance, a page may be "frozen" until the online process returns the necessary results. With CallAppEngine there are no COMMITs issued. There is an asynchronous online PeopleCode option, ProcessRequest. With ProcessRequest, COMMITs are allowed.

Manual Using the Command Line

Usually, you use this technique only during testing or if you need to manually restart the program.

In this section, we discuss how to:

- Use Process Scheduler to start Application Engine batch programs.
- Use the Application Engine Process Request page.
- Use PeopleCode to call an Application Engine program.
- Use the command line to invoke an Application Engine program

Using Process Scheduler

To use PeopleSoft Process Scheduler for starting your Application Engine batch programs you need to create a Process Definition for each program. Select PeopleTools, Process Scheduler, Process & Definition. You will enter options on two or three pages.

Process Type: Application Engine
 Name: TEST
 *Description: TEST API Aware
 Long Description:
 *Priority: Medium

Process Type Definition page: Process & Definition tab

Process Type: Application Engine
 Name: TEST
 Server Name: NT Server Agent
 Recurrence Name: Daily Purge
 Component: AE_APPL_TBL
 Process Groups: ARCHALL

Process Type Definition page: Process Definition Options tab

You can list parameters on the Override Options page. Here's the complete parameter list:

```
-ct MICROSOFT -cd %%DBNAME%% -co %%OPRID%% -cp %%OPRPSWD%% -r %%RUNCNTLID%% -i
%%INSTANCE%% -ai %%PRCSNAME%
```

Note. When creating a Process Definition based on the Application Engine Process Type Definition, the Process Name you assign must *exactly* match your Application Engine program name.

Running Application Engine programs is very similar to running any COBOL or SQR program that you typically invoke with PeopleSoft Process Scheduler. Use PeopleSoft Application Engine as the generic Process Type Definition, and each Application Engine program that you want to invoke using Process Scheduler requires a unique Process Definition derived from the generic Process Type Definition.

Application Engine Process Request Page

You can also start an Application Engine program by using the Application Engine Process Request page. Using this request page enables you to specify additional values and parameters than those that appear within the Process Scheduler Process Definitions.

Keep in mind that most end users will start Application Engine programs from an application-specific request page using PeopleSoft Process Scheduler. A systems expert or “power user” may, at times, need to create custom Process Requests that require multiple programs to perform parallel processing or that need to set specific, initial values in a State record. This is an example where PeopleSoft recommends using the Application Engine Process Request page.

Note. Generally, if “seed” data or other Application Engine request settings are required for a particular program, the application-specific request page will have SQL Execs that do the work transparent to the user. Typically, no end user should be invoking programs from the generic process request page discussed in this section. PeopleSoft uses this page for internal testing and to provide an example of how you can design you program-specific request pages.

Tables Used in Process Request Page

The Application Engine Process Request page inserts values into the following tables:

- **AEREQUESTTBL.** Contains all of the values that appear on the page except those in the Parameters group.
- **AEREQUESTPARM.** Includes only Initial State record values specified in the Parameters group, if needed.

Note. Inserting a row in either of the Application Engine request tables is *not* required to run an Application Engine program. This is a key difference from versions of PeopleSoft Application Engine *prior* to PeopleTools 8.x where a row in the Application Engine request tables is required to start a program *regardless* of how it is invoked. The Run Control ID is available to your program using %RunControl, whether there’s a row inserted into the AEREQUESTTBL.

You need to use the Application Engine Request page to invoke Application Engine and insert a row into the Application Engine Request records only if you need to perform any of the following tasks:

- Insert initial values into the State records associated with a particular program.
- Set an As Of Date for the Application Engine program to perform retroactive processing.
- Set a non-default market for the program.
- Set up a temporary table image to use if you are submitting a process request that will perform parallel processing.

Creating an Application Engine Request

Access the Application Engine Request page by selecting PeopleTools, Application Engine, Request AE..

Application Engine Request

User ID: QEDMO **Run Control ID:** PORTAL_MTUPG

Program Name: PORTAL_MTUPG

Last Run

Process Origin: Other	Process Instance:	Status: Pending
------------------------------	--------------------------	------------------------

Process Frequency: **Market:** **As Of Date:**

Parameters

State Record: ***Bind Variable Name:**

Value:

Date:

Application Engine Process Request page

To create an Application Engine process request page:

1. Select PeopleTools, Application Engine, Request AE.
2. Using the search page, enter a new value or find an existing value of the program name or run control ID for the request.
3. Enter appropriate values.

Last Run	This group of controls shows you the results from the last time a user submitted a particular Run Control. (Rather than using the Process Monitor).
Process Origin	Indicates from where the program was invoked: from the Process Scheduler, Command Line, and so on.
Process Instance	This is a display-only field for informational purposes. It shows the Process Instance assigned to the previous program run.
Status	Here you can check the status of the last program run, whether it is Successful, Pending, and so on.
Process Frequency	Specify how long a particular Process Request will remain active or valid: <i>Always:</i> Select if you want a Process Request to be run as needed.

	<i>Once</i> : Select if a Process Request is a one-time-only request.
	<i>Don't</i> : Select to disable a Process Request so that no one accidentally invokes it and potentially corrupts data.
Market	To run a program designed for a non-default Market, select the appropriate Market from the drop-down list box.
As Of Date	If you are requesting retroactive processing, specify the appropriate as of date.
Parameters	If you need to set any initial values in State records before a program runs, enter the appropriate values in this group of controls. To add additional rows, press F7.
State Record	Enter the name of the State Record associated with the program.
Bind Variable Name	Enter the appropriate field/bind variable for which you are inserting a value.
Value	Enter the initial value that you want to set for the specified field.

Using PeopleCode

To call a particular Application Engine program from a page using PeopleCode, you must use the `CallAppEngine()` function in your `SavePreChange` or `SavePostChange` PeopleCode. The basic syntax for `CallAppEngine()` is as follows:

```
CallAppEngine(applid [, statereclist ]);
```

Note. `RemoteCall` function is no longer valid for invoking Application Engine programs in PeopleCode. However, the `RemoteCall` function still applies to calling other COBOL functions. If you don't convert the `RemoteCall` PeopleCode that previously called an Application Engine program to use the new function, an error message appears.

When to Use CallAppEngine()

Use `CallAppEngine` if the program you are invoking is a “quick” process or will not require a user to wait for an unreasonable amount of time for the transaction to complete. Keep in mind, this is a synchronous process, which means the end user must wait for any process invoked by `CallAppEngine()` to complete before doing *anything* else. Developers must consider the end user. If the called program will cause an unreasonable delay, then PeopleSoft suggests using another alternative, such as the `ScheduleProcess` PeopleCode function.

Use `CallAppEngine` when you have a complex, SQL intensive business process that must run in batch *and* online or the process requires the use of dedicated temporary tables. If this is not the case, you are usually better off writing the entire program in native PeopleCode. Keep in mind that if you've written logic in PeopleCode, presumably for online, and you want to reuse it in batch, you may be forced into row-by-row processing. PeopleSoft recommends designing the batch logic *first*, and then you can decide whether to have a separate online version or just

reuse the batch code—using CallAppEngine. Developers must consider the tradeoff between code re-use and performance. It is inherently more difficult, but not impossible, to develop a common solution that performs adequately in both batch and online.

Do *not* use CallAppEngine within an Application Engine PeopleCode step. If you need to call an Application Engine program from another Application Engine program, you must use the Call Section Action.

Do *not* use CallAppEngine if you need to control the commit operation. Programs called with CallAppEngine are embedded within a larger unit of work defined by the page trigger, such as a page save.

Note. Online PeopleCode that calls CallAppEngine should be set to run on the application server. You will encounter performance issues if you attempt to run PeopleCode on the Client in a three-tier configuration, because every SQL statement that Application Engine issues must be serialized and then sent to the application server for execution.

Using the Command Line

In some cases, you may want to invoke an Application Engine program through the command line. For instance, this method is typically used in the following situations:

- **Restarting.** When a program abends, a system administrator might restart the program using the command line. If needed, you can locate all of the specific program/process information from the Process Monitor on the Process Request Detail dialog. Normally, end users (or system administrators) will perform a Restart from the Process Monitor.
- **Development/Testing.** Many developers include the command line in a batch file to launch a program they are developing or testing. This way, they can quickly execute the batch file as needed. This also enables separation of development of the Application program from its associated pages.
- **Debugging.** To debug a program with a Run Location of Server, you can log into the server (using Telnet, for example) and invoke the program from the command line.

To start an Application Engine program from the command line, you must specify the Application Engine executable (PSAE.EXE) followed by the required parameters, as shown in the following example:

```
psae -CT <dbtype>
      -CS <server>
      -CD <database name>
      -CO <oprid>
      -CP <oprpswd>
      -R <run control id>
      -AI <program id>
      -I <process instance>
      -DEBUG <Y|N>
```

```

-DR <Y|N>
-TRACE <value>
-DBFLAGS <value>
  -TOOLSTRACESQL <value>
-TOOLSTRACEPC <value>
  -OT <value>
-OF <value>
-FP<value>

```

Or, if your command line parameters are stored in a text file, you can enter:

```
psae <parmfile>
```

Note. For Windows NT and UNIX servers, you must set the PS_SERVER_CFG environment variable before you invoke an Application Engine program from the command line. PS_SERVER_CFG must contain the fully qualified name of a correctly configured Process Scheduler PSPRCS.CFG file. When PeopleSoft Application Engine runs from the command line, it resolves %PS_SERVDIR% to the value of the environment variable PS_SERVDIR instead of the parent directory of a Process Scheduler configuration.

Command Line Parameters

- CT <dbtype>** Required. Corresponds to the type of the database to which you are connecting. Values are MICROSOFT, ORACLE, SYBASE, INFORMIX, DB2UNIX, and DB2.
- CS <server>** Required for Sybase, Informix. For those platforms that require a server name as part of signon, enter the appropriate server name. This affects Sybase, Informix, and Microsoft SQL Server. However, for Microsoft SQL Server, this option is *valid* but not required.
- CD <database name>** Required. Enter the name of the database to which the program will connect.
- CO <oprid>** Required. Enter the Operator ID that is running the program.
- CP <oprpswd>** Required. Enter the password associated with the specified Operator ID/User ID.
- R <run control id>** Required. Enter the Run Control ID to use for this run of the program.
- AI <application id>** Required. Specify the Application Engine program to run.
- I <process_instance>** Required for Restart. Enter the Process Instance for the program run. The default is 0, which means PeopleSoft Application Engine uses the next available Process Instance.

- DEBUG <Y|N>** This parameter controls the Debug utility. Enter *Y* to indicate that you want the program to run in debugging mode, or enter *N* to indicate that you do not.
- DR <Y|N>** This parameter controls Restart. Enter *Y* to disable Restart, or enter *N* to enable Restart. (DR represents Disable Restart).
- TRACE <value>** This parameter turns on the Application Engine trace. To enable tracing from the command line, enter this parameter and a specific Trace value. The value you enter is the sum of the specific traces that you want to enable. The current traces and values are:
- 1—Initiates the Application Engine Step trace.
 - 2—Initiates the Application Engine SQL trace.
 - 128—Initiates the Application Engine timings file trace, which is similar to the COBOL timings trace.
 - 256—Includes the PeopleCode Detail Timings in the 128 trace.
 - 1024—Initiates the Application Engine timings table trace, that stores the results in database tables.
 - 2048—Initiates the database optimizer “explain”, writing the results to the trace file. This option is supported only on Oracle, Informix, and Microsoft SQL Server.
 - 4096—Initiates the database optimizer “explain”, storing the results in the Explain Plan Table of the current database. This option is supported only on Oracle, DB2, and Microsoft SQL Server.
- So, if you wanted to enable the 1, 2, and 128 traces, enter *131*—the sum of 1, 2, and 128. To indicate that you do not want any traces, enter *0*. If you don’t explicitly enter *0*, Application Engine will use the trace value set in PeopleSoft Configuration Manager.
- DBFLAGS** Using this parameter you can disable the %UpdateStats meta-SQL construct. To disable %UpdateStats enter:
- DBFLAGS 1**
- TOOLSTRACESQL <value>** Use this parameter to enable the SQL Trace.
- TOOLSTRACEPC <value>** Use this parameter to enable the PeopleCode Trace.
- psae <Parmfile>** If you submit a file to PeopleSoft Application Engine as the first parameter in the command line, Application Engine reads the contents of the file and interprets the contents as if it were parameters entered on the command line. This

option is intended mainly for the Windows NT or UNIX Process Scheduler Server environment. For example,

```
psae $temp/myparmfile.txt
```

Note. For security reasons, after Application Engine interprets the contents of the <parmfile>, it immediately deletes the <parmfile>.

- OT <value>** (Optional) Use to initialize the PeopleCode meta-variable %OutDestType (Numeric).
- ```
&ProcessRqst.OutDestType = %OutDestType ;
```
- OT** (Optional) Use to initialize the PeopleCode meta-variable %OutDestFormat (Numeric).
- PeopleCode example of %OutDestFormat:
- ```
Query.RunToFile(Record QryPromptRecord,
%OutDestFormat);
```
- FP** (Optional) Use to initialize the PeopleCode meta-variable %FilePath (String).
- PeopleCode example of %FilePath :
- ```
If All(%FilePath) Then
 &FILENAME = %FilePath | &FILENAME;
 &MYFILE = GetFile(&FILENAME, "E",
%FilePath_Absolute);
Else
 &MYFILE = GetFile(&FILENAME, "E",
%FilePath_Relative);
End-If;
```

## Debugging Application Engine Programs

This section discusses how to:

- Enable the Application Engine Debugger.
- Set debugger options.

---

### Enabling the Debugger

The following procedure describes the steps to have your program run in debug mode.



|                                                |
|------------------------------------------------|
| <b>To debug an Application Engine program:</b> |
|------------------------------------------------|

1. Set the Debug Option.

You can set the Debug option in the following locations:

- Start PeopleSoft Configuration Manager and select the Process Scheduler tab. In the Application Engine group, enable Debug by selecting the **Debug** check box. This is the method that applies to all methods of invocation.
- If you used the command line option to invoke your Application Engine program, then you can just include the `-DEBUG Y` parameter in the command line you submit to `PSAE.EXE`. If you already have the Debug check box selected in PeopleSoft Configuration Manager, then you do not need to include the `-DEBUG` parameter in your command line.
- If you have PeopleCode in your Application Engine program, you should also enable the PeopleCode Debugger. When you launch your program and the PeopleCode Action executes, you will enter the PeopleCode Debugger at that point.

---

**Note.** Setting the debug capabilities in either PeopleSoft Configuration Manager or the command line will turn debug mode on. The only situation where this is not true is when you have Debug enabled in Configuration Manager and you explicitly submit `-DEBUG N` on the command line. In this case, the PeopleSoft Configuration Manager setting defines your “default” command line value, and the command line can override the default.

---

2. Execute the Application Engine program that you want to debug.
3. At the Application Engine Debugger prompt, enter the appropriate command that enables the desired debugging option.

Each command is represented by a single letter such as ‘X’, ‘L’, or ‘M’. Enter the letter that corresponds to the current option you want to engage. To see a list of the available debugging options, enter ? at the prompt. For example,

Application Engine Debugger - enter command or type ? for help.

```
AETESTPROG.MAIN.STATS> ?
```

```
Debug Commands:
```

|             |                                                   |
|-------------|---------------------------------------------------|
| (Q)uit      | Rollback work and end program                     |
| E(X)it      | Commit work and end program (valid between steps) |
| (C)ommit    | Commit work (valid between steps)                 |
| (B)reak     | Set or remove a break point                       |
| (L)ook      | Examine state record fields                       |
| (M)odify    | Change a state record field                       |
| (W)atch     | Set or remove a watch field                       |
| (S)tep over | Execute current step or action and stop           |

|                 |                                                   |
|-----------------|---------------------------------------------------|
| Step (I)nto     | Go inside current step or called section and stop |
| Step (O)ut of   | Execute rest of step or called section and stop   |
| (G)o            | Resume execution                                  |
| (R)un to commit | Resume execution and stop after next commit       |

### To enable the PeopleCode Debugger for Application Engine:

1. Sign on to PeopleTools using the same User ID that you are going to use to invoke the Application Engine program.
2. Open PeopleSoft Application Designer.
3. Select Debug, PeopleCode Debugger Mode.

Your Application Engine program can be open on the desktop, but you do not need to open the Application Engine program or the PeopleCode action that you want to debug.

4. Select Debug, Break at Start.

This causes the Application Engine program to break prior to executing any PeopleCode programs within it.

---

## Debugging Options

Each debugger option is represented by a single letter that you must specify at the prompt. To engage the option you select, press ENTER.

### ***Before You Get Started***

Note the following tips about debugging programs:

- While using the debugger, you can always enter ? to get a list of all the available debugging options.
- In some cases, such as when setting breakpoints or Watch Fields, there are sub-menus that offer more options. After you are familiar with the commands, you can enter multiple items on the command line to combine commands to bypass the sub-menus. For example, to see a list of the breakpoints, you could enter

```
B L
```

Or, to set a field as a Watch Field, you could enter

```
W S MY_FIELD
```

Or, if it's on a different state record, enter

```
W S MY_AET.MY_FIELD
```

---

**Note.** The exception to this option is Modify, which always displays the current value and then prompts you to enter the new value. You can, however, enter M MY\_AET.MY\_FIELD to get directly to the new value prompt.

---

- The letter commands are not case sensitive. For example, 'Q' or 'q' is a valid command to submit at the prompt.

### Quit

Enter Q or q. This option performs a rollback on the current unit of work in the debugging run, and it ends the debugging session. It effectively acts as an abort of your Application Engine program.

Quit is useful for testing Restart. You want to have some of the work committed and some of it uncommitted. Then, abort at that point and roll back the pending work. You want to make sure the program restarts from the point of the last successful commit.

### Exit

Enter X or x. This option is valid only after one Step has completed and another has not already begun. It is not valid once you reach the Action level.

Use this option as an alternative to Quit. Exit will end the program run and the debugging session, but it will also commit the current unit of that the program has already completed. This option can be helpful when testing your Restart logic.

### Commit

Enter C or c. To commit the current unit of work in your program, use this option. It is valid only after a step has completed and before another has already begun. It is not valid once you reach the action level.

You can use this option, for example, to use your database query tool to check the data in your database.

### Break

Enter B or b. Sets a breakpoint. When the program reaches the breakpoint, it temporarily halts execution to enable you to observe the state of the current process.

Breakpoint options:

**Set.** Enter S or s to set a breakpoint location. For example:

```
AETESTPROG.MAIN.TESTNUM> b
(S)et, (U)nset, or (L)ist? s
Program [AETESTPROG] :
Section [MAIN] :
Step [TESTNUM] :
```

```
Breakpoint set at AETESTPROG.MAIN.TESTNUM
```

The breakpoint location defaults to the current location in the program, but you can specify other Sections or Steps,

by overriding the default values that appear in the brackets.

**Unset.** Enter U or u to remove any breakpoints previously set.

```
AETESTPROG.MAIN.TESTNUM> b
(S)et, (U)nset, or (L)ist? u
```

```
Active Breakpoints:
(1) AETESTPROG.MAIN.STATS
(2) AETESTPROG.MAIN.TESTNUM
```

```
Remove which breakpoint? 1
```

In the previous example, the break point set at AETESTPROG.MAIN.STATS will be removed.

```
List. Enter L or l to view the list of Active
Breakpoints. AETESTPROG.MAIN.TESTNUM> b
(S)et, (U)nset, or (L)ist? l
```

```
Active Breakpoints:
(1) AETESTPROG.MAIN.TESTNUM
```

When you want to enter this command, make sure that you have entered B or b first to specify the Break option. If you just enter L or l from the main command prompt, you engage the Look option.

## Look

Enter L or l. Enables you to observe the values currently in the state record associated with the program you are debugging. You must specify the state record at the Record Name prompt. By default, the default state record as specified in your Program Properties appears with the brackets.

You can also specify a specific field name on the state record in the Field Name prompt. To look at all the fields in the state record, leave the '\*' within the brackets unchanged.

## Modify

Enter M or m. Enables you to modify the value of a state record value for debugging purposes. Suppose the previous Steps did not set a value correctly. However, you may want to see how the rest of the program would perform *if* the appropriate value existed in the state record. This enables you to give your program some help in the debugging or testing phase.

As with the Look command, you must specify the appropriate state record (if you are using multiple state records), and you must specify one field. You can modify only one field at time.

|                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Watch</b>       | <p>Enter W or w. When you specify a field as a Watch Field, the program stops when the value of the field changes.</p> <p>Similar to the Break command you can Set, Unset, and List.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <b>Step Over</b>   | <p>Enter S or s. Executes the current step to completion and stop at the next step in the current section.</p> <p>The behavior depends on the current “level” or the program. You start at the step level, and then have the ability to “step into” the action level. If you are at the step level and use “step over”, you go to the next step in the current section, skipping over all actions (including any Call Sections). If you are at the action level, “step over” executes the current action and stops at the next action in the current step, or at the next step in the current section.</p>                                                                                                                                                                                                         |
| <b>Step Into</b>   | <p>Enter I or i. Use this option to observe a step or called section in a more granular level. For instance, you can check each SQL statement and stop. By using this option and checking the state record at each stop, you can easily isolate problem SQL or PeopleCode.</p> <p>As with Step Over, the behavior depends on the level. At the step level, you can “step into” the action level and stop before the first action in the step. At the action level, if the current action is a call section, Step Into takes you to the first step in the called section. For other action types, Step Into acts the same as Step Over, because there is no deeper level in which to step.</p>                                                                                                                      |
| <b>Step Out of</b> | <p>Enter O or o. After you’ve “Stepped Into” a Step or Called Section, use the “Step Out of” option to run the rest of the current Step or Called Section and stop. As with the previous “Step” options, the behavior of Step out of depends on the current level of the program.</p> <p>At the Step level, Step Out Of completes the remaining Steps in the current Section, returns to the calling Section or Step, and stops at the next Action in that Step. If the Section is MAIN and is not called by another Section or Step, then Step out of behaves the same as the Go option.</p> <p>At the Action level, Step Out Of completes the current Step and stops at the next Step in the current Section, or if the program is at the end of a Section, Step Out of returns to the calling Section/Step.</p> |
| <b>Go</b>          | <p>Enter G or g. After the program has stopped at a specific location, and you’ve examined its current state, you can use the Go command to resume the execution of the program. This is a helpful command when you have breakpoints set. With this command, the program won’t</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |

stop at a Step or Action; it will only stop at the next breakpoint, watched variable change (Watch Field), or until the program runs to completion.

### Run to commit

Enter R or r. Resumes execution of your program after it has stopped. This command forces the program to stop again after the next commit. This is a good option to use when observing your commit strategy and how it will affect a Restart.

### Example of Look Option

```
Record Name [AE_TESTAPPL_AET] :
Field Name [*]:
AE_TESTAPPL_AET:
 PROCESS_INSTANCE = 37
 AE_PRODUCT = 'AE'
 AE_APPL_ID = 'TESTAPPL'
 AE_STEP = ' '
 AE_SECTION = ' '
 AE_INT_15 = 0
 AE_INT_14 = 0
 AE_INT_13 = 0
 AE_INT_12 = 0
 AE_INT_11 = 0
 AE_INT_10 = 0
 AE_INT_9 = 0
 AE_INT_8 = 0
 AE_INT_7 = 8
 AE_INT_6 = 11
 AE_INT_5 = 3
 AE_INT_4 = 0
 AE_INT_3 = 0
 AE_INT_2 = 0
 AE_INT_1 = 0
```

If just want to view the value stored in a specific field of the state record after a particular Step or Action, just enter the appropriate field name at the Field Name prompt. For example,

```
AE TESTPROG.MAIN.TESTNUM> 1
Record Name [AE_TESTAPPL_AET] :
Field Name [*]: AE_INT_6
AE_TESTAPPL_AET:
 AE_INT_6 = 11
```

You can also use an asterisk (\*) as a wildcard to get a partial list. For example, if you enter

```
AE_INT*
```

at the Field Name prompt, you see only the fields that start with AE\_INT. This is also true for the Record Name prompt. This is useful both to list multiple fields across multiple records or as a shortcut. If you know there is only one state record that starts with XXX, you don't have to type the full name—just enter XXX.

### **Example of Modify Option**

If you wanted to set the AE\_INT\_15 field in the AETESTPROG to 10, you would enter the following:

```
AETESTPROG.MAIN.TESTNUM> m
Record Name [AE_TESTAPPL_AET]:
Field Name [none]: AE_INT_15

Current value: AE_TESTAPPL_AET.AE_INT_15 = 0

Enter new value (do not use quotes around text strings):
10
```

Using the Look command, you can check to see that the value you specified now exists in the state record. For example,

```
AETESTPROG.MAIN.TESTNUM> l
Record Name [AE_TESTAPPL_AET]:
Field Name [*]: AE_INT_15
AE_TESTAPPL_AET:
 AE_INT_15 = 10
```

### **Example of Watch Option**

- **Set.** Enter S or s to set a Watch field.

```
Set or remove a watch field
AETESTPROG.TESTNUM.INTRESLT> w
(S)et, (U)nset, or (L)ist? s
Record Name [AE_TESTAPPL_AET]:
Field Name [none]: AE_INT_7
```

- **Unset.** Enter U or u to “unset” or remove a Watch field from the list.

```
AETESTPROG.TESTNUM.INTMSG> w
(S)et, (U)nset, or (L)ist? u

Active Watch Fields:
(1) AE_TESTAPPL_AET.AE_INT_6 = 6
(2) AE_TESTAPPL_AET.AE_INT_7 = 7

Remove watch on which field? 2
```

To remove a field from the Watch Field list, enter the sequence number in the list that corresponds to the field that you want to remove. In the previous example, we will remove AE\_INT\_7 from the Watch Field list by entering 2.

- **List.** Enter L or l. After the completion of a Step or Action, you can view the values of all the fields that you have included in the Watch list. For example,

```

AETESTPROG.TESTNUM.INTMSG> w
(S)et, (U)nset, or (L)ist? 1

Active Watch Fields:
(1) AE_TESTAPPL_AET.AE_INT_6 = 6
(2) AE_TESTAPPL_AET.AE_INT_7 = 7

```

---

**Note.** You cannot set a Watch on a “long text” field.

---

## Restarting Application Engine Programs

One key feature of PeopleSoft Application Engine is its built-in checkpoint/restart capabilities. If there is an abnormal termination (abend) or failure on a step in the program, the end user can restart the request from the last successful checkpoint, or the step immediately preceding the step that failed. The end user restarts the program from the Process Request page.

PeopleSoft Application Engine can save a program state and restart where it stopped processing in the case of an abend (abnormal termination). This section discusses how to:

- Enable restart.
- Disable restart.

---

### How Restart Works

Application Engine programs save to the database (perform a commit) only when an entire program successfully completes. Program developers must set any individual commits where appropriate.

At the section level, you can set a commit after each step in that section. And, at the step level, you can require or defer commits for individual steps, or you can increase the commit frequency within a step to *n* iterations of a looping action within a step, such as a Do Select or Do While.

The commit level you select plays a major role in how restart works in a program. Each time that PeopleSoft Application Engine issues a commit with restart *enabled*, it records the current state of the program. The recording of the current state that PeopleSoft Application Engine performs is referred to as a *checkpoint*.



The program behavior changes depending on whether you have restart enabled or disabled. If you have restart enabled, whenever there is a commit, PeopleSoft Application Engine also performs a checkpoint beforehand. When you have restart disabled Application Engine performs a commit only.

Therefore, if a failure occurs at any point in the process, the end user can restart the program and expect the program to behave in the following manner:

- Ignore the steps that have already completed up to the last successful commit.
- Begin processing at the next step after the last successful commit.

The ability for Application Engine to “remember” what steps have already completed and which steps have not depends on an important record called AERUNCONTROL—keyed by Process Instance.

When a program runs, each time PeopleSoft Application Engine issues a commit it also saves all of the information required for a program restart in the AERUNCONTROL record.

### ***Design Effects of Restart***

Using the restart feature enables the developer to perform commits more often in a program. This reduces the overall impact on other users and processes while the background program is running. This is because it reduces the amount of rows that are locked by the program, allowing multiple instances of the program to run concurrently (parallel processing), which may be useful for high volume solutions.

---

## **When to Use Restart**

Usually, you want to develop programs to take advantage of the Application Engine restart capabilities. Programs that are good candidates for restart are those that do a lot of preparation work up front, like joining tables and loading data into temporary work tables. Also, programs that might put your data in an unstable state if they abend during a run should be considered to take advantage of restart. As a general rule, restart is essential for programs that primarily do set-based processing.

However, if your program has one the following characteristics you may want to *disable* restart:

- It’s mainly row-by-row processing.
- The overhead involved with PeopleSoft Application Engine performing a checkpoint during the program run is not desirable.
- The program commits after  $n$  iterations of a looping construct within a step, and the select statement driving the loop is composed in such a way that if the program abended and then started again, it would ignore transactions that were already processed in the previous program run. In this sense, the program processes the restart internally in that PeopleSoft Application Engine treats each start of a program as a “fresh” start instead of restarting a previous instance.

When developing for restart, PeopleSoft recommends that you consider the consequences if a program fails and you can't restart the program. Given the commit structure that you've defined for your Application Engine program, would your data remain in an usual state if a failure were to occur after any of the commits? Would it be easy to recover from such a case?

### ***At the Program Level***

PeopleSoft Application Engine automatically performs all state record updates. When an Application Engine program starts, it inserts a row in the state record for the assigned process instance. Then the system updates the state record whenever the program performs a commit to store changed values into the database. Finally, the state record row is deleted upon successful completion of the application.

However, if the state record the program uses is a work record, no database updates can be made to the record. Consequently, if you restart the program you might get unexpected results because the "memory" was lost when the program abended. In fact, the system re-initializes any state records that are work records at each commit, to ensure consistent behavior during a normal run and a restarted run. Therefore, you may need to make at least one of your state records a SQL table to contain values that must be retained across commits or in case of an abend.

Finally, the other consideration for programming for restart at the program level is to check *both* the Application Engine Program Properties dialog box and PeopleSoft Configuration Manager to make sure that Disable Restart check box is not selected.

### ***At the Section Level***

The section level property associated with restart is section type, which has the options Prepare Only and Critical Updates.

If a section is only preparing data, as in selecting it, populating temporary tables, or updating temporary tables, then set the section type to Prepare Only. However, if the section is actually updating *permanent* application tables in the database, set the option to Critical Updates.

During runtime, when the system arrives at the first section set to Critical Updates, it sets the AE\_CRITICAL\_PHASE value in the AERUNCONTROL record to *Y*. Once set, the value of AE\_CRITICAL\_PHASE will remain *Y* until the program completes successfully. When the program completes, the corresponding row in AERUNCONTROL is deleted. Therefore, a Prepare Only section following the Critical Updates section won't reset the AE\_CRITICAL\_PHASE value to *N*.

If your program does abend, the end user can check the AE\_CRITICAL\_PHASE value. If it's *Y*, they know that the section that failed is critical and that the program should be restarted to ensure data integrity. If AE\_CRITICAL\_PHASE = *N*, restarting may not be necessary; however, as a general rule, PeopleSoft recommends restarting even if AE\_CRITICAL\_PHASE is set to *N*.

### ***At the Step Level***

In your program's Where clause of a Do Select, you should include some conditions that reduce the answer set returned from the select.

For example,

```
SELECT RECNAME, FIELDNAME
 FROM PS_AERECFIELD
 ORDER BY RECNAME, FIELDNAME
```

If you ran this select statement as part of a Do Select action with *Restartable* selected as the Do Select type, the system might process some of the rows twice after a restart. Also, if you have specified *Reselect*, the program could execute in an infinite loop, because there's nothing to reduce the answer set. However, if you modified the select statement to look more like the following, you could make it *Restartable*.

```
SELECT RECNAME, FIELDNAME
 FROM PS_AE_RECFIELD
 WHERE RECNAME > %Bind(RECNAME)
 OR (RECNAME = %Bind(RECNAME) AND FIELDNAME > %Bind(FIELDNAME))
 ORDER BY RECNAME, FIELDNAME
```

A Do Select statement that has been coded for *Restartable* can be converted to *Select/Fetch*, but the opposite is not true.

The previous example shows the use of a key column to reduce the answer set. This can be convenient if your record has only one or two key fields. However, if your record has three or four keys, your SQL would become overly complex.

Instead of matching key fields, you could add a switch to the selected table, and then have the processing of the called section modify the switch as it processes the row. In this example, your select statement could look like the following:

```
SELECT COLUMN1, COLUMN2, . . .
 FROM PS_TABLE1
 WHERE PROCESSING_SWITCH='N' . . .
```

---

## Controlling Abends

Abends, or abnormal ends of your program, can be controlled or uncontrolled.

A controlled abend means that Application Engine exits "gracefully" because of a calculated error condition. Some examples of controlled abends are:

- SQL errors while you have set On Error = Abort
- PeopleCode return value of if On Return = Abort
- SQL statement affects no rows and you have set On No Rows = Abort

In these situations (when PeopleSoft Application Engine is in control) the Run Status in Process Monitor reads "Error".

An uncontrolled abend occurs in situations where there is a memory violation or a user terminates a process. In these cases, the Run Status in Process Monitor shows "Processing".

---

## Restarting an Application Engine Program

There are two ways to restart an Application Engine program:

- From the command line.
- From a Process Request page.

---

**Note.** The following procedures for restarting a failed Application Engine program assume that you have rectified the error that caused the program to fail in the first place. For instance, suppose the name of a referenced table has changed. Regardless of how many times you restart the program, it will continue to fail until you've modified references to the old table name.

---

### ***From the Command Line***

Normally, using the command line for restarting your Application Engine programs is reserved for developers and system administrators. End users, in most cases, should not be expected to use this method to restart programs that fail.

You can use the command line option to restart programs that have a run location of either Client or Server. PeopleSoft Application Engine references only the Process Instance of the failed process. Therefore, if you run a process on the client and it fails, you can restart it from the server using the server command line. Likewise, if you run a process from the server and it fails, you could restart it from the client using the command line.

**To restart an Application Engine program from the command line:**

1. Collect the command line values associated with the failed program.

These values include database type, database name, Operator ID and password, Run Control ID, program name, and the Process Instance. You can find these variables on the Process Details dialog box, the corresponding State Record, or the Application Engine Run Control table. Where the values reside depends on how you invoked the program. For instance, if you invoked the program using the command line, or outside of PeopleSoft Process Scheduler, then you will not be able to view any details associated with the program run on the Process Details dialog box.

2. Enter the following command line syntax at the command prompt substituting the values from the previous step.

```
PSAE.EXE -CT <DB_TYPE> -CD <DB_NAME> -CO <OPRID> -CP <PASSWORD> -R
<RUN_CONTROL> -AI <PROGRAM_NAME> -I <PROCESS_INSTANCE>
```

---

**Note.** Some database platforms, such as Sybase, also require that you include a server name in the argument list.

---

### ***From the Process Request Page***

You can restart programs from a Process Requests page only for those programs that have a server run location. Because most end users require assistance with the command line interface, PeopleSoft recommends that you limit the number of programs that run on the client.

#### **To restart an Application Engine program from the Process Requests page:**

1. Open Process Scheduler by selecting PeopleTools, Process Scheduler, System Process Requests.
2. Locate the Run Control ID Number of the program you need to restart.
3. To display the details of the failed process, click the **Process Detail** link.
4. On the Process Request Details page, select **Restart Request**, and click **OK**.

---

### **When to Restart or Start Again**

When an Application Engine program ends abnormally, you may have to decide whether you should restart the process or just start it from the beginning. Keep in mind that your Application Engine program ran at least part way through, so starting over may leave your data in an unknown state. Also, there may be some application logic that would need to be "undone" depending on what stage the program was at when it failed, what data the program had committed, and so on.

---

**Note.** By selecting the Disable Restart check box on the Program Properties dialog box you can specify the process to start each time as if it were "new."

---

If you decide that starting your program from the beginning is the best alternative, keep the following in mind. If you try to start a new process with the same Run Control ID and Operator ID as the process in a "restartable" or "No Success" state, you'll receive a "suspend" error. Consequently, to start the program over from the beginning, you must use SQL to delete the row that corresponds to the failed program from the Application Engine run control table and your state record.

#### **To start an Application Engine program from the beginning (again):**

1. Open your native SQL editor, and manually delete the row in PS\_AE\_RUN\_CONTROL that corresponds to the program you want to start from the beginning.

Use the following SQL to accomplish this step:

```
DELETE FROM PS_AE_RUN_CONTROL
WHERE OPRID=<OPRID>
AND RUN_CNTL_ID=<Run_Control_ID>
```

2. Delete from your state record the row that corresponds to the failed program run.

Use the following SQL to accomplish this step:

```
DELETE FROM PS_<MY_AET>
WHERE PROCESS_INSTANCE=<Process_Instance>
```

---

## Common Restart Errors

The section describes the most common errors associated with restarting Application Engine programs.

### ***Bad Restart Messages***

If you attempt to restart what Application Engine believes to be a process that completed successfully, you receive a “bad restart” message. You can also get this message if your Application Engine application is defined with restart disabled.

### ***Suspend Error***

A suspend error occurs when you attempt to start a new process that matches the Run Control ID and Operator ID for another process in a “restartable” state. Because the Process Instance for these two processes is different, the new request fails.

This usually occurs when an end user tries to run the program again after receiving an error on the previous attempt.

To resolve this issue you have the following options:

- Clear the process information from the Run Control table and state record, and start the process over from the beginning.
- Select Restart Request from the Process Request Details dialog box.

---

**Note.** By "restartable," we mean that Disable Restart is not selected on the Program Properties dialog box. Consequently, the program is restartable.

---

## Enabling or Disabling Restart

There are three places that you can enable or disable restart:

- Check box on the Program Properties dialog box
- Check box in the Configuration Manager Profile
- To disable, include the option `-DR Y` to the command line of `PSAE.EXE` (or delete to enable restart)

If in *any* of these three places you've disabled restart, then restart is disabled.

**To enable or disable restart in Program Properties dialog box:**

1. Select File, Definition Properties or click the **Properties** button.
2. Select the Advanced tab and select or clear the Disable Restart check box appropriately.

If you are developing programs and are ready to test but don't want a restart condition to get in the way, you could leave the Disable Restart check box clear in the Program Properties. Then, you could have the Disable Restart check box on the Configuration Manager selected, or if you are invoking your program from the command line, you could have the Restart flag enabled. This allows you to test new development and avoid errors about forcing you to restart during your testing. At the same time it prevents you from inadvertently leaving Disable Restart selected in the Program Properties when you moved the program into production.

**To enable or disable restart in PeopleSoft Configuration Manager:**

1. Start Configuration Manager. Select the Profile tab and click the **Edit** button.
2. Select the Process Scheduler tab and select or clear the Disable Restart check box appropriately.

## Caching the Application Engine Server

PeopleSoft Application Engine caches meta-data, just like the application server. This enhances performance because the program can refer to the local cache for any objects that it uses.

---

### Using CacheBaseDir

Application Engine programs that run on the NT or UNIX server each lock their own cache directory for the duration of the run. These directories are found under the master cache directory. The master directory is created under the directory specified by the CacheBaseDir variable in the PeopleSoft Process Scheduler configuration file. If all existing cache directories are locked, a new one is created. Cache sub-directories are named sequentially, starting at 1.

If you do not enter a fully qualified path for the CacheBaseDir variable, then PeopleSoft Application Engine creates the cache directory within the directory in which the program is set to run.

---

**Note.** Do not share the CacheBaseDir with application servers, and do not use environment variables when specifying CacheBaseDir, because the system does not resolve them. For example, do not have CacheBaseDir=\$PS\_HOME.

---

## Server Caching

In the PSPRCS.CFG (PS\_SERVER\_CFG), there are two additional cache parameters. They are:

- Enable Server Caching
- Server Cache Mode

PeopleSoft recommends that you *do not* alter these settings from the delivered defaults. These settings are reserved for future use.

## Managing Abends

If you are taking advantage of dedicated temporary tables for Application Engine programs, then you need to know how to free, or unlock, a temporary table in the event that the program running against it abends. Because most Application Engine programs run through PeopleSoft Process Scheduler, typically you just use the Process Monitor to unlock the temporary tables. Deleting or restarting a process using the Process Monitor automatically frees the locked temporary tables.

For the programs that you invoke outside of PeopleSoft Process Scheduler, PeopleTools provides the Manage Abends page. Programs running outside of Process Scheduler include those invoked from CallAppEngine PeopleCode and the command line.

| <a href="#">Process Monitor</a> Use Process Monitor for processes that were run via Process Scheduler. |                  |                |              |         |                      |
|--------------------------------------------------------------------------------------------------------|------------------|----------------|--------------|---------|----------------------|
| Temp Tables                                                                                            | Process Instance | Run Control ID | Program Name | User ID | Run Date and Time    |
| <a href="#">Temp Tables</a>                                                                            | 25               | DOCTEST        | BULKTEST     | PTDMO   | 05/17/2000 1:09:57PM |
| <a href="#">Temp Tables</a>                                                                            | 29               | DOCTEST        | AETESTPROG   | PTDMO   | 05/17/2000 1:56:27PM |

Managing Abends page

Use the following procedure to free temporary tables locked by a program abend.

### To free locked temporary tables using the Manage Abends page:

1. Select **PeopleTools, Application Engine, Manage Abends**.
2. Identify the program that has the particular temporary tables locked.

You can uniquely identify programs using the Process Instance, Run Control ID, Program Name, User ID, and Run Date and Time columns.

3. Click the **Temp Tables** link.
4. On the Temporary Tables sub-page, click the **Release** button to unlock the temporary tables associated with the program.



## CHAPTER 6

# Calling Application Engine Programs From COBOL

To facilitate the conversion of existing COBOL programs to Application Engine programs, you can now call Application Engine programs from existing COBOL code.

This chapter discusses how to:

- Add a copybook to your COBOL program.
- Assign copybook values.

## Adding Copybook

Include the copybook called PTCCBLAE.CBL to your COBOL program.

### *Listing of PTCCBLAE.CBL*

```
*01 CBLAE.
NOCLN 02 CBLAE-PRCSNAME PIC X(12) VALUE SPACE.
NOCLN 02 CBLAE-COMMIT-FLAG PIC X(1) VALUE SPACE.
 88 AE-COMMITS-SUCCESS VALUE 'B' .
 88 AE-COMMITS-ALL VALUE 'C' .
02 CBLAE-PARMS .
 03 CBLAE-PARM-CNT PIC 9(4) COMP.
 03 CBLAE-PARM-ENT OCCURS 500 TIMES.
 05 CBLAE-STATEREC PIC X(15) .
 05 CBLAE-FIELDNM PIC X(18) .
 05 CBLAE-DATA-PTR POINTER.
 05 CBLAE-LENGTH PIC 9999 COMP.
 05 CBLAE-SCALE PIC 99 COMP.
NOCLN 05 CBLAE-TYPE PIC X.
 88 CBLAE-TYPE-CHAR VALUE 'C' .
 88 CBLAE-TYPE-SMALLINT VALUE 'S' .
 88 CBLAE-TYPE-INT VALUE 'I' .
 88 CBLAE-TYPE-DEC VALUE 'P' .
 88 CBLAE-TYPE-DATE VALUE 'D' .
 88 CBLAE-TYPE-TIME VALUE 'T' .
```

```

88 CBLAE-TYPE-TIMEONLY VALUE 'V'.
88 CBLAE-TYPE-NUMERIC VALUE 'S' 'I' 'P'.

```

## Assigning Copybook Values

To assign values to the calling COBOL program's copybook to be passed as parameters into the state records of the called Application Engine program:

- Identify the fields in your COBOL program that contain the values you want to pass to the Application Engine program.
- Load the PTCCBLAE.CBL copybook with the state record name, field name, field length (this should be the size of the field not the size of the contents), the scale (decimal places if any), and set the field type.
- Call the PTPSETAD program to set the pointer in PTCCBLAE.CBL to the host programs variable.
- Set the variable AE-COMMIT-FLAG to either AE-COMMITS-ALL or AE-COMMITS-SUCCESS. AE-COMMITS-ALL means that the Application Engine program would commit as specified in the program. AE-COMMITS-SUCCESS means that the Application Engine program ignores all commits and performs one commit at the end of successful execution.

### *Example of Loading Values from PTPSTAE.CBL Sample Program*

```

MOVE 0 TO CBLAE-PARM-CNT OF CBLAE

ADD 1 TO CBLAE-PARM-CNT OF CBLAE
MOVE 'QE_CBLAETST_AET' TO CBLAE-STATEREC
 OF CBLAE (CBLAE-PARM-CNT OF CBLAE)
MOVE 'DESCR' TO CBLAE-FIELDNM
 OF CBLAE (CBLAE-PARM-CNT OF CBLAE)
MOVE 30 TO CBLAE-LENGTH
 OF CBLAE (CBLAE-PARM-CNT OF CBLAE)
MOVE 0 TO CBLAE-SCALE
 OF CBLAE (CBLAE-PARM-CNT OF CBLAE)
SET CBLAE-TYPE-CHAR OF CBLAE (CBLAE-PARM-CNT OF CBLAE)
 TO TRUE
CALL 'PTPSETAD' USING CBLAE-DATA-PTR
 OF CBLAE (CBLAE-PARM-CNT OF CBLAE)
 W-DESCR OF W-WORK

ADD 1 TO CBLAE-PARM-CNT OF CBLAE
MOVE 'QE_CBLAETST_AET' TO CBLAE-STATEREC
 OF CBLAE (CBLAE-PARM-CNT OF CBLAE)

```

```

MOVE 'QE_AE_INT_7' TO CBLAE-FIELDNM
 OF CBLAE (CBLAE-PARM-CNT OF CBLAE)
MOVE 2 TO CBLAE-LENGTH
 OF CBLAE (CBLAE-PARM-CNT OF CBLAE)
MOVE 0 TO CBLAE-SCALE
 OF CBLAE (CBLAE-PARM-CNT OF CBLAE)
SET CBLAE-TYPE-SMALLINT
OF CBLAE (CBLAE-PARM-CNT OF CBLAE)
 TO TRUE
CALL 'PTPSETAD' USING CBLAE-DATA-PTR
 OF CBLAE (CBLAE-PARM-CNT OF CBLAE)
 W-SMINT OF W-WORK

```

\*

```

DA000-CALL-AE SECTION.
DA000.

```

\*

```

MOVE 'QE_AETESTPRG' TO CBLAE-PRCSNAME OF CBLAE
SET AE-COMMITS-ALL TO TRUE

CALL 'PTPCBLAE' USING SQLRT CBLAE.
CALL-AE-EXIT.
EXIT.

```

---

**Note.** Make sure the calling COBOL program has successfully connected to the database before calling PTPCBLAE; and that the calling program is not running through a RemoteCall function.

---

### **Sample of the Communication Area of PTPBLAE.CBL**

```

* PTPCBLAE - Communication area for PTPCBLAE *
*01 CBLAE.
NOCLN 02 CBLAE-PRCSNAME PIC X(12) VALUE SPACE.
* Name of AE program to be called.
NOCLN 02 CBLAE-COMMIT-FLAG PIC X(1) VALUE SPACE.
* Flag to determine which of the following commits to make.
88 AE-COMMITS-SUCCESS VALUE 'B'.
* No in-process commit; if successful, then commit occurs.
88 AE-COMMITS-ALL VALUE 'C'.
* Commits occur when defined in the AE program.
02 CBLAE-PARMS.
03 CBLAE-PARM-CNT PIC 9(4)COMP.
* Counter of the number of state records passed.
03 CBLAE-PARM-ENT OCCURS 500 TIMES.
* Maximum value of state record entries.

```

```

 05 CBLAE-STATEREC PIC X(15) .
* State record name.
 05 CBLAE-FIELDNM PIC X(18) .
* Field name.
 05 CBLAE-DATA-PTR POINTER.
* Pointer to your own working storage area.
 05 CBLAE-LENGTH PIC 9999 COMP.
* Field length of defined state record.
 05 CBLAE-SCALE PIC 99 COMP.
* Number of decimal places.

NOCLN 05 CBLAE-TYPE PIC X.
* Field data type.
 88 CBLAE-TYPE-CHAR VALUE 'C' .
 88 CBLAE-TYPE-SMALLINT VALUE 'S' .
 88 CBLAE-TYPE-INT VALUE 'I' .
 88 CBLAE-TYPE-DEC VALUE 'P' .
 88 CBLAE-TYPE-DATE VALUE 'D' .
 88 CBLAE-TYPE-TIME VALUE 'T' .
 88 CBLAE-TYPE-TIMEONLY VALUE 'V' .
 88 CBLAE-TYPE-NUMERIC VALUE 'S' 'I' 'P' .

```

If the called Application Engine program made any updates to the state records or fields (that were passed by PTPCBLAE), they will be stored in the calling program's local variables as identified by PTPSETAD.

### **Developer Notes**

PeopleSoft created the PS\_AECOBOLPARM table to act as an interface between COBOL programs and Application Engine programs. The PTPCBLAE copybook is populated by the application program, then PTPCBLAE goes through the contents of the copybook and converts every variable to a character string representation.

If the character value is greater than 254 bytes, the value is split into 254 byte chunks and inserted into the PS\_AECOBOLPARM table with appropriate sequence numbering. Then the PTPCBLAE copybook calls the C program C\_STARTAEP, which in turn calls PSAE.

PSAE can determine if it was called by a COBOL program; if so, PSAE checks for values stored in the PS\_AECOBOLPARM table. Then the values are assigned to the appropriate state record field.

---

**Note.** For users that previously developed COBOL programs calling PeopleSoft Application Engine, ensure that you have updated the call to PTPCBLAE to pass the PTCLOGMS copybook as the second parameter.

---

### **Error Handling**

At this time, PeopleSoft suggests the following if you need error handling for your program:

- Add a field (return code) to your state record.
- Initialize the field to a negative value.
- Pass the value into the Application Engine program.
- Then, at the end (successful completion) of the Application Engine program change the field value to a positive value.
- Check for that value in your COBOL program.



## CHAPTER 7

# Tracing Application Engine Programs

This chapter provides an overview of tracing Application Engine programs and discusses how to enable, locate, and understand the results of Application Engine traces.

## Understanding Tracing Application Engine Programs

You can set the following traces to monitor the performance of your Application Engine programs:

- Application Engine step and SQL trace
- Application Engine statement timings trace
- PeopleCode detail timings trace
- Database optimizer trace

---

**Note.** The general PeopleTools SQL and PeopleCode traces also apply to Application Engine programs.

---

## Enabling Application Engine Traces

By default, all Application Engine traces are turned off. When you need to see a trace or a combination of traces, you must set any trace options prior to executing the program.

You enable traces by using any of the following methods:

- Setting command line options
- Server configuration files
- Setting options Configuration Manager

### **Setting Command Line Options**

The command line option is available on Windows NT and UNIX, but it is not available when calling Application Engine programs from PeopleCode.

To enable tracing from the command line, specify the `-TRACE` parameter within the command line that you submit to `PSAE.EXE`. For example,

```
n:\pt840\bin\client\winx86\psae.exe -CT MICROSOFT -CD PT800GES -CO PTDMO -CP
PTDMO -R PT8GES -AI AETESTPROG -I 45 -TRACE 2
```

The following is a list of the Trace values and what they do:

|             |                                                                                                                                                                                                                |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>0</b>    | Disables tracing.                                                                                                                                                                                              |
| <b>1</b>    | Initiates the Application Engine step trace.                                                                                                                                                                   |
| <b>2</b>    | Initiates the Application Engine SQL trace.                                                                                                                                                                    |
| <b>4</b>    | Initiates the trace for dedicated Temp Table Allocation to AET file.                                                                                                                                           |
| <b>128</b>  | Initiates the statement timings trace to file, which is similar to the COBOL timings trace to file.                                                                                                            |
| <b>256</b>  | Initiates the PeopleCode Detail to the file for the Timings trace.                                                                                                                                             |
| <b>1024</b> | Initiates the statement timings trace, but, instead of writing to the trace file, this trace stores the results in the following tables: <code>PS_BAT_TIMINGS_LOG</code> and <code>PS_BAT_TIMINGS_DTL</code> . |
| <b>2048</b> | Adding this value requests a database optimizer trace file.                                                                                                                                                    |
| <b>4096</b> | Requests a database optimizer to be inserted in the Explain Plan Table of the current database.                                                                                                                |
| <b>8192</b> | This value sets a trace for Integration Broker transform programs.                                                                                                                                             |

To specify traces on the command line, you enter the sum of the desired trace options. This is similar to adding the trace values using `PSADMIN`, such as the COBOL statement timings or the SQL statement trace value. To specify a combination of the traces, just enter the sum of the corresponding trace values. For example, if you wanted to enable the step (1), the SQL (2), and the statement timings (128) traces, you would enter 131—the sum of 1, 2, and 128.

If you want to completely disable tracing, you must explicitly specify `-TRACE 0`. If you don't include the `-TRACE` flag in the command line, PeopleSoft Application Engine uses the value specified in the Process Scheduler configuration file or the Configuration Manager. Otherwise, the command line parameters override any trace settings that may be set in the Configuration Manager.

### **Setting Parameters in Server Configuration Files**

You can also enable traces in the configuration files for both the application server and the Process Scheduler server.

- **Application Server.** For programs invoked by PeopleCode and run on the Application Server, set the `TraceAE` parameter in the Trace section to the desired value in the Application Server configuration file (`PSAPPSRV.CFG`). You can use `PSADMIN` to set this parameter.

```
Values for config section - Trace
TraceSql=0
```



```
TraceSqlMask=12319
TracePC=0
TracePCMask=4095
TracePPR=0
TracePPRMask=4095
TraceAE=<add trace value>
```

This option is available on Windows NT and UNIX.

- **Process Scheduler Server.** In the PeopleSoft Process Scheduler configuration file, you can set the TraceAE parameter in the Trace section to the desired level of tracing. You can use PSADMIN to set this parameter.

```
Values for config section - Trace
TraceFile=%PS_SERVDIR%\logs\PeopleTools.trc
TraceSQL=0
TracePC=0
TraceAE=<add trace value>
```

This option is available on Windows NT and UNIX, and applies *only* to Application Engine programs invoked in batch mode.

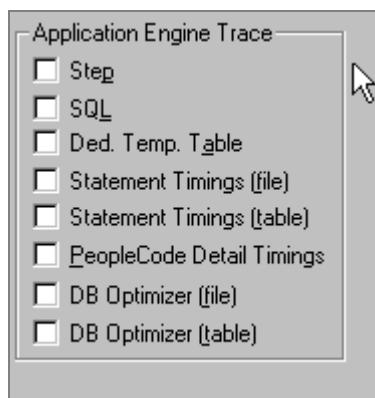
---

**Note.** The TraceFile parameter does not specify the location of the Application Engine trace file; it applies only to the generic PeopleTools SQL and PeopleCode traces.

---

### Setting Options in PeopleSoft Configuration Manager

For processes running on a Windows workstation, you can set your trace options on PeopleSoft Configuration Manager. This procedure is valid only if you are running Application Engine programs on a Windows workstation—the development environment.



Application Engine Trace check boxes

#### To set Application Engine traces:

1. Start the Configuration Manager, and select the Trace tab.

2. Select the trace options that apply to your current situation.

You can select any combination of the options.

3. After you have selected the appropriate options, click either the **Apply** or **OK** button to set your trace options.

---

## Locating the Trace File

Where you look for the generated trace file depends on how you invoked the program and the operating system on which the program runs, as follows:

| <b>Program started from:</b> | <b>Where to look for Trace file</b>                                                                                                                     |
|------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| Windows workstation          | Look for the trace file in %TEMP%\PS\ <db name&gt;<="" td=""> </db>                                                                                     |
| PeopleCode                   | Look for the trace file %TEMP%\PS\ <db \$ps_home="" \<db="" and="" log="" name&gt;="" name&gt;<="" nt="" on="" td=""> </db>                             |
| Command Line                 | Look for the trace file in the directory specified in the Log/Output field in the PS_SERVER_CFG file                                                    |
| Process Scheduler            | Look for the trace file in a sub-directory of the directory specified in the Log/Output field in the PS_SERVER_CFG file                                 |
| Without a Process Instance   | Application Engine names the file according the following convention.<br>AE_<Date/Time_Stamp>_<OS_PID>.AET<br>Where the date/time stamp is <mmddhhmmss> |
| With a Process Instance      | Application Engine names the file according to the following convention.<br>AE_<Program_name>_<Process_Instance>.AET                                    |

---

**Note.** For an Application Engine program running on a server, PeopleTools writes the generic PeopleTools trace for SQL and PeopleCode trace files to the same directories as the AET traces. The prefix of the trace file name is also the same, and the suffix is `trc`. On the Windows workstation, the trace is written to the “People Tools Trace File” specified in the Trace folder of PeopleSoft Configuration Manager.

---



---

## Understanding the Trace Results

At the top of each trace there is useful information that helps you to identify the PeopleTools version, the database name, and the database platform type.

### Step Trace Report

The step trace reports each step name that your program executes and in what order. Associated with each step is a timestamp, the Do level, and the Action type.

Also, the trace shows the steps that execute within a called section by the indented formatting. For example, a step that executes within a called section is preceded by two dots (..) while other steps are preceded by only one dot. The following is a sample trace file.

```

PeopleTools 8.40 -- Application Engine
Copyright (c) 2001-2002 PeopleSoft, Inc.
All Rights Reserved
Database: PT840GES(SqlServer)

16.42.29 2001-05-12 Tracing Application Engine program AETESTPROG Test
application
16.42.29 .(AETESTPROG.MAIN.STATS) (Do Select)
16.42.29 .(AETESTPROG.MAIN.STATS) (Call Section AETESTPROG.STATS)
16.42.30 ..(AETESTPROG.STATS.SECTIONS) (SQL)
16.42.30 ..(AETESTPROG.STATS.STEPS) (SQL)
16.42.30 ..(AETESTPROG.STATS.STMTS) (SQL)
16.42.30 ..(AETESTPROG.STATS.MSG) (Log Message)
16.42.31 .(AETESTPROG.MAIN.STATS) (Do Fetch)
16.42.31 .(AETESTPROG.MAIN.STATS) (Call Section AETESTPROG.STATS)
16.42.31 ..(AETESTPROG.STATS.SECTIONS) (SQL)
16.42.31 ..(AETESTPROG.STATS.STEPS) (SQL)
16.42.31 ..(AETESTPROG.STATS.STMTS) (SQL)
16.42.31 ..(AETESTPROG.STATS.MSG) (Log Message)
16.42.31 .(AETESTPROG.MAIN.STATS) (Do Fetch)
16.42.31 .(AETESTPROG.MAIN.TESTNUM) (Call Section AETESTPROG.TESTNUM)
16.42.31 ..(AETESTPROG.TESTNUM.INTEGER) (SQL)
16.42.31 ..(AETESTPROG.TESTNUM.INTRESLT) (SQL)
16.42.31 ..(AETESTPROG.TESTNUM.INTMSG) (Log Message)
16.42.31 .(AETESTPROG.MAIN.DBSTATS) (Call Section AETESTPROG.DBSTATS)
16.42.32 ..(AETESTPROG.DBSTATS.UPDSTATS) (SQL)
16.42.32 .(AETESTPROG.MAIN.PCODE) (Call Section AETESTPROG.PCODE)
16.42.32 ..(AETESTPROG.PCODE.PCode01) (PeopleCode)
16.42.32 ..(AETESTPROG.PCODE.PCode02) (PeopleCode)
16.42.33 Application Engine program AETESTPROG ended normally
16.42.33 Application Engine ended normally

```

### SQL Trace Report

The SQL trace shows the formatted SQL processes including commits, rollbacks, and restarts. You can also view the buffers associated with each SQL statement. Use the SQL trace to spot errors in your SQL and to view your commit strategy.

The following is a sample of the SQL trace file.

```

PeopleTools 8.40 -- Application Engine
Copyright (c) 2001-2002 PeopleSoft, Inc.
All Rights Reserved
Database: PT840GES(SqlServer)

```

```

%Select(AE_APPL_ID, AE_PRODUCT) SELECT AE_APPL_ID, AE_PRODUCT FROM
PS_AE_APPL_TBL ORDER BY AE_APPL_ID, AE_PRODUCT
/
-- Buffers:
-- 1) EC_AUDIT
-- 2) PO

%Select(AE_INT_5) SELECT COUNT(*) FROM PS_AE_SECTION_TBL WHERE AE_PRODUCT =
:1 AND AE_APPL_ID = :2
/
-- Buffers:
-- 1) 1

COMMIT
/
%Select(AE_INT_6) SELECT COUNT(*) FROM PS_AE_STEP_TBL WHERE AE_PRODUCT =
'PO' AND AE_APPL_ID = 'EC_AUDIT'
/
-- Buffers:
-- 1) 7

COMMIT
/
%Select(AE_INT_7) SELECT COUNT(*) FROM PS_AE_STMT_TBL WHERE AE_PRODUCT =
'PO' AND AE_APPL_ID = 'EC_AUDIT'
/
-- Buffers:
-- 1) 7

COMMIT
/

```

### ***Application Engine Statement Timing Trace***

The Application Engine statement timing trace is very similar to a COBOL timings trace in which you monitor the execution of your COBOL programs for performance evaluations. This trace enables you to gather performance information to determine program bottlenecks. Once bottlenecks have been identified, it may be possible to modify your program to run more efficiently, or you may want to change the database schema and configuration to optimize the execution of your program.

The statement timings trace is invaluable for tuning an Application Engine program. It may also be useful as a default trace level for all production runs to provide a metric for long term performance trends.

By examining all of the figures in this trace, you can easily identify areas of your program that are not running as efficiently as possible. For instance, if compile counts are high, you can reduce the numbers by using Application Engine's reuse feature. If inserts appear to be running slow and you have many of them, you can increase the performance by using Application Engine's bulk insert feature.

You can opt to write this trace to a file, or you can write the results to tables. Either way, timings trace overhead is minimal. Internal testing reveals that the Application Engine trace has an overhead between 2% and 5% of total runtime.

Each value in the trace, including cumulative totals, appears in a form rounded to the nearest tenth of a second, but totals are calculated using non-rounded timings.

---

**Note.** PeopleSoft Application Engine does not write the timings trace to table for programs invoked by CallAppEngine. To write to the table a Process Instance is required, and programs invoked by CallAppEngine are not assigned a Process Instance.

---



---

## Trace File Sections

### SQL Counts and Timings (Section 1)

The first section of the trace file is the SQL section. It records the performance of application specific SQL. The trace values appear within a series of columns and sections as shown in the following example.

| SQL Statement             | Compile |      | Execute |      | Fetch |      | Time  |
|---------------------------|---------|------|---------|------|-------|------|-------|
|                           | Count   | Time | Count   | Time | Count | Time |       |
| -----                     |         |      |         |      |       |      |       |
| <b>PeopleCode</b>         |         |      |         |      |       |      |       |
| BI_RECFIELD_SEL           |         |      | 1       | 0.0  | 153   | 0.0  | 0.0   |
| INSERT PS_INTFC_BI_NTMP1  |         |      | 1       | 0.0  | 0     | 0.0  | 0.0   |
| <snip>                    |         |      |         |      |       |      |       |
| UPDATE PS_INTFC_BI_HTMP1  |         |      | 7       | 0.0  | 0     | 0.0  | 0.0   |
|                           |         |      |         |      |       |      | ----- |
|                           |         |      |         |      |       |      | 0.2   |
| <b>Application Engine</b> |         |      |         |      |       |      |       |
| COMMIT                    | 0       | 0.0  | 76      | 0.2  | 0     | 0.0  | 0.2   |
| LOGMSG                    | 0       | 0.0  | 4       | 0.1  | 0     | 0.0  | 0.1   |
|                           |         |      |         |      |       |      | ----- |
|                           |         |      |         |      |       |      | 0.3   |

**AE Program: BIIF000**

|                                    |   |     |    |     |   |     |     |
|------------------------------------|---|-----|----|-----|---|-----|-----|
| AE.UPD_AE.S                        | 1 | 0.0 | 16 | 0.0 | 0 | 0.0 | 0.0 |
| ASGNIVC1.NTMP.D                    | 1 | 0.0 | 1  | 0.0 | 2 | 0.0 | 0.0 |
| ASGNIVC1.REGISTER.D                | 1 | 0.0 | 1  | 0.0 | 2 | 0.0 | 0.0 |
| ASGNIVC2.UPDIVC.D                  | 1 | 0.0 | 1  | 0.0 | 8 | 0.0 | 0.0 |
| ASGNIVC5.CK_BIHDR.D                | 1 | 0.0 | 1  | 0.0 | 1 | 0.0 | 0.0 |
| ASGNIVC5.INSHTMP.S<br>(BulkInsert) | 1 | 0.1 | 17 | 0.0 | 1 | 0.0 | 0.1 |
| ASGNIVC6.Step01.S                  | 1 | 0.0 | 16 | 0.0 | 0 | 0.0 | 0.0 |
| <snip>                             |   |     |    |     |   |     |     |
| UPIVCAMT.UPCHDRAM.D                | 7 | 0.4 | 7  | 0.0 | 7 | 0.0 | 0.4 |
| UPIVCAMT.UPHDRAMT.S                | 7 | 0.8 | 7  | 0.0 | 0 | 0.0 | 0.8 |

-----  
41.6

The following table contains more information about each column within the first section of the trace file.

- SQL Statement**                      Application Engine SQL Actions and stored SQL objects always have a statement ID. The SQL Statement column shows the statement ID so that you can attribute the trace values to individual SQL statements. In the case of SQLExec SQL, a portion of the SQL statement appears in the first column to help you identify it. For SQL objects, PeopleSoft recommends that you use the TraceName property in the CreateSQL so that you can uniquely identify it in the traces.
- Compile Column**                      This column of values shows how many times the system compiled a SQL statement and how long it took. "Compiled" refers to the SQL statement being sent to the database to be parsed and optimized, and it also includes the time required for the first resolution of any PeopleSoft Meta-SQL.
- Execute Column**                      This column shows how many times the system executed the SQL statement and the time consumed doing so. "Executed" refers to the system sending the compiled SQL to the database server to be run against the database.
- Fetch Column**                         This column applies to SELECT statements. It shows how many rows your program fetched from the database and how much time this consumed. Keep in mind that the system must first Execute a SELECT statement against the database to find the relevant rows and generate an "active" set. After the set exists, the program must still fetch the rows. Some database API's have buffered fetches which means that the fetch may include more than one row. This makes subsequent fetches "free" until the buffer becomes empty.
- Total Column**                         This column shows the sum of the Compile, Execute, and Fetch times of the SQL statement. Some database API's may defer a Compile to the Execute and/or defer an Execute to the first Fetch.

|                                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|---------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>PeopleCode SQL</b>                       | This sub-section is for SQL executed from PeopleCode Actions. Compile counts and times for such SQL is included in Execute Count and Times because the programmer does not explicitly control ReUse. To determine whether ReUse is occurring, you must do a program run after enabling the generic PeopleTools trace for SQL statements, API calls, and so on. As a starting point, PeopleSoft suggests a trace value of 31.                                                                                                                                                                                  |
| <b>Application Engine SQL</b>               | This sub-section reveals the time attributed to Application Engine overhead that is not directly related to the SQL within your program. For example, the values in this section represent the SQL generated for checkpoints, commits, and so on. If there are COMMITs without checkpoints, it indicates that restart has been disabled, or a restartable program has called a non-restartable program.<br><br>If the time consumed performing a checkpoint or committing seems more than expected, you should try to reduce it if possible by setting the Commit Frequency of the Steps containing DO loops. |
| <b>AE Program:<br/>&lt;Program_Name&gt;</b> | This sub-section shows all the SQL Actions for a particular program. The Action properties that impact performance are flagged. For example, BulkInsert. ReUse is not flagged because it is self-evident when the Execute count is higher than the compile count.                                                                                                                                                                                                                                                                                                                                             |

---

**Note.** Keep in mind that when you run a SQL trace at the Application Engine level and the PeopleTools level simultaneously, you may see misleading results. In short, extra overhead gets added to the overall SQL timings by the PeopleTools trace. Tracing SQL at the Application Engine level (-TRACE) adds to the non-SQL times because PeopleTools writes the trace data after timing the SQL.

---

### ***PeopleCode Actions (Section 2)***

The second section, or PeopleCode section, records the performance associated with all the PeopleCode Actions in your program.

|                   |                                                                                                                                                                                                     |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>PeopleCode</b> | This column contains all the names of the PeopleCode Actions in your program.                                                                                                                       |
| <b>Call</b>       | This column shows how many times each PeopleCode Action gets called during the program run.                                                                                                         |
| <b>Non-SQL</b>    | This column shows the time consumed by your PeopleCode Actions that does not involve any SQL.                                                                                                       |
| <b>SQL</b>        | This column shows the time consumed by your PeopleCode Actions executing SQL. The SQL time total should be similar to that of the PeopleCode SQL subsection in the first section of the trace file. |

**Total** The total indicates the cumulative amount of time spent in the Action.

The following example shows a sample PeopleCode Actions section.

| PeopleCode           | Call Count | Non-SQL Time | SQL Time | Total Time |
|----------------------|------------|--------------|----------|------------|
| -----                |            |              |          |            |
| AE Program: BIIF0001 |            |              |          |            |
| AE.UPD_AE            | 16         | 0.0          | 0.0      | 0.0        |
| ASGNIVC1.NTMP        | 1          | 0.0          | 0.0      | 0.0        |
| ASGNIVC1.REGISTER    | 1          | 0.0          | 0.0      | 0.0        |
| <snip>               |            |              |          |            |
| NOTE.NEXTSEQ         | 7          | 0.0          | 0.0      | 0.0        |
| OPTEDIT.UPD2NEW      | 1          | 0.0          | 0.0      | 0.0        |
| UPDDSLVL.INIT        | 9          | 0.0          | 0.0      | 0.0        |
| UPIVCAMT.UPCHDRAM    | 7          | 0.0          | 0.0      | 0.0        |
|                      |            | -----        | -----    | -----      |
|                      |            | 2.5          | 0.2      | 2.7        |

---

**Note.** The columns containing all zeros but totaling 2.5, for example, is somewhat misleading. Keep in mind that the system rounds to the first decimal place (tenths) but only after calculating the sum of each Action time. You can assume that for the previous example, each Action required .09 seconds or less to complete.

---

**PeopleCode Built-ins and Methods (Section 3)**

The third section contains either a list of all of the PeopleCode built-ins and methods used or a summary thereof. To get a list of all of the built-ins and methods, you must enable the PeopleCode Detail Timings in addition to the Statement Timings trace.

If a method or built-in function consumes a large amount of time, you may want to consider alternatives. For example, if array processing dominates your run time, consider inserting the data into temporary tables and performing the processing on tables in the database.

The following example shows a trace with the PeopleCode Detail Timings trace turned on.

| PEOPLECODE Builtin/Method             | Count | Time |
|---------------------------------------|-------|------|
| -----                                 |       |      |
| Decimal(Type 0) Builtin Len           | 9     | 0.0  |
| Decimal(Type 0) Builtin RoundCurrency | 52    | 0.0  |
| String(Type 1) Builtin LTrim          | 16    | 0.0  |



```

<snip>
Array(Type 263) BuiltIn CreateArrayRept 6 0.0
Array(Type 275) Method Push 6 0.0

```

The following example shows a trace with the PeopleCode Detail Timings trace turned off.

```

 E x e c u t e
PEOPLECODE Builtin/Method Count Time

Any(Type 4) BuiltIns 8 0.0
SQL(Type 130) Methods 8 0.1
SQL(Type 130) BuiltIns 2 0.0
Record(Type 131) BuiltIns 1 0.2

```

#### **Summary Data (Section 4)**

The fourth section contains summary data. The values in this section reveal an overview of the program run without drilling down too far into the details.

```

Total run time : 50.8

Total time in application SQL : 42.1 Percent time in application SQL :
83.0%

Total time in PeopleCode : 2.5 Percent time in PeopleCode :
5.0%

Total time in cache : 2.5 Number of calls to cache :
257

```

The following table describes the values that appear in this section.

|                                        |                                                                                                                                                                                       |
|----------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Total run time</b>                  | This value presents the overall amount of time a program required to complete from start to finish.                                                                                   |
| <b>Time in application SQL</b>         | This value represents all of the time that your program spent executing SQL. The value includes SQL executed by both PeopleCode and SQL Actions.                                      |
| <b>Percent time in application SQL</b> | This value represents the percentage of time spent executing SQL compared to the entire program run.                                                                                  |
| <b>Time in PeopleCode</b>              | This value represents all of the time that your program spent executing PeopleCode. Time in PeopleCode excludes SQL executed from within PeopleCode.                                  |
| <b>Percent time in PeopleCode</b>      | This value represents the percentage of time spent executing PeopleCode compared to the entire program run.                                                                           |
| <b>Total time in Cache</b>             | This value represents the amount of time your program spent retrieving objects from cache or refreshing cache. Time in cache includes all memory cache access, file cache access, and |

SQL executed to load managed objects, such as Application Engine program components, meta-data, and so on. Time will vary according to where Application Engine finds an object. For instance, retrieving an object that the system cached during a previous run will be faster than retrieving it from the database.

**Number of calls to Cache** This value represents the actual number of calls your program made to cache. The number of calls to cache will remain constant for the same Application Engine program processing the same data.

### ***Environment Information (Section 5)***

The fifth section contains environment information specific to PeopleSoft Application Engine. If your programs appear to be performing poorly, you should always check the trace value that you have set.

Each trace produces an unavoidable degree of overhead. As such, the more traces you have enabled the more likely you are to see degraded performance. Make sure that you are running only the traces you need. This section shows you the following:

- PeopleTools SQL Trace
- PeopleTools PeopleCode Trace
- Application Engine Trace
- Application Engine DbFlags (%UpdateStats)

### ***Table***

You also have the option of writing the Statement Timings traces to a table. By storing timings information in a table, you can store historical data in the database, which enables you to produce reports that aid in trend analysis, allow ad hoc SQL queries for longest running statements, and so on. With the timings data stored in the database, you can manipulate and customize reports to show only the metrics in which you are most interested.

You can use third-party tools to query and present the data in such ways as detailed graphical representations of your program's performance. You can also implement alarms if the performance of a program reaches a specified maximum value in a particular area such as SQL Compile time.

The Statement Timings trace populates the following tables. For convenience, we've included the columns and schema for the tables associated with this trace.

### ***PS\_BAT\_TIMINGS\_LOG (Parent)***

This table stores general information for a program run.

| BAT_TIMINGS_LOG (Record) |                  |             |                 |                           |  |
|--------------------------|------------------|-------------|-----------------|---------------------------|--|
| Record Fields            |                  | Record Type |                 |                           |  |
| Num                      | Field Name       | Type        | Short Name      | Long Name                 |  |
| 1                        | PROCESS_INSTANCE | Nbr         | Instance        | Process Instance          |  |
| 2                        | PROCESS_NAME     | Char        | Proc Name       | Process Name              |  |
| 3                        | OPRID            | Char        | User            | User ID                   |  |
| 4                        | RUN_CNTL_ID      | Char        | Run Cntl        | Run Control ID            |  |
| 5                        | BEGINDTTM        | DtTm        | Begin Date/Time | Begin Date/Time           |  |
| 6                        | ENDDTTM          | DtTm        | End Date/Time   | End Date/Time             |  |
| 7                        | TIME_ELAPSED     | Nbr         | Elapsed         | Elapsed Time (in millisec |  |
| 8                        | TIME_IN_PC       | Nbr         | In PeopleCode   | PeopleCode Time (in milli |  |
| 9                        | TIME_IN_SQL      | Nbr         | In SQL          | SQL Time (in millisecond  |  |
| 10                       | TRACE_LEVEL      | Nbr         | AE Trace level  | Trace level (AE)          |  |
| 11                       | TRACE_LEVEL_SAM  | Nbr         | SAM Trace Level | SAM Trace Level           |  |

PS\_BAT\_TIMINGS\_LOG table

**PS\_BAT\_TIMINGS\_DTL (Child)**

This table stores the more granular details associated with a program run, such as Execute Count, Fetch Time, and so on.

| BAT_TIMINGS_DTL (Record) |                   |             |     |        |                 |                          |
|--------------------------|-------------------|-------------|-----|--------|-----------------|--------------------------|
| Record Fields            |                   | Record Type |     |        |                 |                          |
| Num                      | Field Name        | Type        | Len | Format | Short Name      | Long Name                |
| 1                        | PROCESS_INSTANCE  | Nbr         | 10  |        | Instance        | Process Instance         |
| 2                        | BAT_PROCESS_TYPE  | Char        | 1   | Upper  | Process type    | Process type (cobol, AE, |
| 3                        | BAT_PROGRAM_NAME  | Char        | 12  | Upper  | Program         | Program or module name   |
| 4                        | BAT_DTL_TYPE      | Char        | 1   | Upper  | Detail type     | Detail type (SQL, People |
| 5                        | DETAIL_ID         | Char        | 100 | Mixed  | Detail ID       | Detail line identifier   |
| 6                        | COMPILE_COUNT     | Nbr         | 10  | Raw B  | Compile Count   | Count                    |
| 7                        | COMPILE_TIME      | Nbr         | 15  |        | Compile Time    | Time (in milliseconds)   |
| 8                        | EXECUTE_COUNT     | Nbr         | 10  | Raw B  | Count           | Count                    |
| 9                        | EXECUTE_TIME      | Nbr         | 15  |        | Time            | Time (in milliseconds)   |
| 10                       | FETCH_COUNT       | Nbr         | 10  | Raw B  | Fetch Count     | SQL Fetch Count          |
| 11                       | FETCH_TIME        | Nbr         | 15  |        | Fetch Time      | SQL Fetch Time           |
| 12                       | RETRIEVE_COUNT    | Nbr         | 10  | Raw B  | Count           | Count                    |
| 13                       | RETRIEVE_TIME     | Nbr         | 15  |        | Time            | Time (in milliseconds)   |
| 14                       | BULK_INSERT       | Char        | 1   | Upper  | Bulk Insert fla | Bulk Insert flag         |
| 15                       | PEOPLECODECOUNT   | Nbr         | 10  |        | PC Count        | PeopleCodeCount          |
| 16                       | PEOPLECODESQLTIME | Nbr         | 15  |        | Time            | PeopleCodeSQLTime        |
| 17                       | PEOPLECODETIME    | Nbr         | 15  |        | PC Time         | PeopleCodeTime           |
| 18                       | CURRENCY_ROUND    | Char        | 1   | Upper  | Currency roundi | Currency rounding flag   |
| 19                       | EXECUTE_EDITS     | Char        | 1   | Upper  | Execute edits f | Execute edits flag       |

PS\_BAT\_TIMINGS\_DTL table

### PS\_BAT\_TIMINGS\_FN

This table stores the PeopleCode Detail Timings information.

| Num | Field Name       | Type | Len | Format | Short Name | Long Name              |
|-----|------------------|------|-----|--------|------------|------------------------|
| 1   | PROCESS_INSTANCE | Nbr  | 10  |        | Instance   | Process Instance       |
| 2   | DETAIL_ID        | Char | 100 | Mixed  | Detail ID  | Detail line identifier |
| 3   | EXECUTE_COUNT    | Nbr  | 10  | Raw B  | Count      | Count                  |
| 4   | EXECUTE_TIME     | Nbr  | 15  |        | Time       | Time (in milliseconds) |

### PS\_BAT\_TIMINGS\_FN

### BATTIMES.SQR

PeopleSoft provides BATTIMES.SQR as an example of the type of reports you can generate to reflect the information stored in the previous BAT\_TIMINGS tables.

You can produce a summary report for all the programs for a specific Run Control ID that looks similar to the following:

```

PeopleTools 8.1
Copyright (c) 1988-2000 PeopleSoft, Inc.
All Rights Reserved
All timings in seconds

 PeopleSoft Stored Batch Timings Summary Report

Process Name : AETESTPR0G Process Instance : 2
Run Control ID : greg Operator ID : PTDM0

Total run time : 5.5 Run Start : 04/24/2000 10:42:04
Total time in application SQL : 1.9 Run Complete : 04/24/2000 10:42:40
Total time in PeopleCode : 0.3 Percent time in application SQL : 33.7
AE Trace Setting : 1024 Percent time in PeopleCode : 4.7
PeopleTools SQL Trace Setting : 31

Process Name : AETESTPR0G Process Instance : 14
Run Control ID : greg Operator ID : PTDM0

Total run time : 3.1 Run Start : 04/24/2000 11:03:16
Total time in application SQL : 0.8 Run Complete : 04/24/2000 11:03:19
Total time in PeopleCode : 0.0 Percent time in application SQL : 25.2
AE Trace Setting : 1024 Percent time in PeopleCode : 1.0
PeopleTools SQL Trace Setting : 31

Process Name : AETESTPR0G Process Instance : 15
Run Control ID : greg Operator ID : PTDM0

Total run time : 3.6 Run Start : 04/24/2000 11:03:27
Total time in application SQL : 0.8 Run Complete : 04/24/2000 11:03:30
Total time in PeopleCode : 0.3 Percent time in application SQL : 22.2
AE Trace Setting : 1024 Percent time in PeopleCode : 9.0
PeopleTools SQL Trace Setting : 31

```

Summary Report for all program runs from a Run Control

Or, you can get detail data for a specific Process instance that looks similar to the following:

PeopleTools 8.1  
Copyright (c) 1988-2000 PeopleSoft, Inc.  
All Rights Reserved  
All timings in seconds

PeopleSoft Stored Batch Timings Detail Report

Process Name : AETESTPROG Process Instance : 28  
Run Control ID : greg Operator ID : PTDMO

Total run time : 4.3 Run Start : 04/24/2000 12:10:37  
Run Complete : 04/24/2000 12:10:41

Total time in application SQL : 0.9 Percent time in application SQL : 20.7  
Total time in PeopleCode : 0.1 Percent time in PeopleCode : 1.4  
AE Trace Setting : 1024 PeopleTools SQL Trace Setting : 31

| SQL Statement      | C o m p i l e |      | E x e c u t e |      | F e t c h |      | Total Time |
|--------------------|---------------|------|---------------|------|-----------|------|------------|
|                    | Count         | Time | Count         | Time | Count     | Time |            |
| -----              |               |      |               |      |           |      |            |
| AE Internal        |               |      |               |      |           |      |            |
| -----              |               |      |               |      |           |      |            |
| COMMIT             | 0             | 0.0  | 15            | 0.1  | 0         | 0.0  | 0.1        |
| LOGMSG             | 0             | 0.0  | 3             | 0.3  | 0         | 0.0  | 0.3        |
|                    |               |      |               |      |           |      | -----      |
|                    |               |      |               |      |           |      | 0.4        |
| -----              |               |      |               |      |           |      |            |
| AETESTPROG         |               |      |               |      |           |      |            |
| -----              |               |      |               |      |           |      |            |
| DESTATS.UPDSTATS.S | 1             | 0.0  | 1             | 0.2  | 0         | 0.0  | 0.2        |
| MAIN.STATS.D       | 1             | 0.0  | 1             | 0.0  | 3         | 0.0  | 0.0        |
| STATS.SECTIONS.S   | 1             | 0.0  | 2             | 0.0  | 2         | 0.0  | 0.1        |
| STATS.STEPS.S      | 2             | 0.0  | 2             | 0.0  | 2         | 0.0  | 0.0        |
| STATS.STMTS.S      | 2             | 0.0  | 2             | 0.0  | 2         | 0.0  | 0.0        |
| TESTNUM.INTEGER.S  | 1             | 0.0  | 1             | 0.0  | 1         | 0.0  | 0.0        |
| TESTNUM.INTRESLT.S | 1             | 0.0  | 1             | 0.0  | 1         | 0.0  | 0.0        |

For Help, press F1 NUM 1/1

### Detail Timings Report

You invoke the BATTIMES.SQR through PeopleSoft Process Scheduler using the following procedure.

#### To invoke batch timings:

1. Select **PeopleTools, Process Scheduler, Batch Timings**.  
The Batch Timings page appears.
2. From the **Report Type** drop-down list box, select *Detail* or *Summary*.
3. In the **Batch Timings For** group box, enter the Run Control ID for Summary reports, and enter the specific Process Instance for Detail reports.
4. When you have made the appropriate selections, click the **Run** button.

#### To view the batch timings using Process Monitor:

1. Select **PeopleTools, Process Scheduler, Process Monitor**.
2. Locate the program run associated with the current trace.
3. Click the **Job Details** button.

4. On the Process Detail dialog box, click the **Batch Timings** link.

On the Batch Timings dialog box, the PeopleCode Detail Timings do not appear. They appear only in the file format.

### **DB Optimizer Trace**

With the DB Optimizer trace you can write the trace to a file or a table. The DB Optimizer trace reveals the execution/query plan for the SQL that your Application Engine program generates. Each SQL statement gets traced only once.

How you view the results of this trace depends on the type of RDBMS you're currently using. For instance, on some platforms only the trace to file option is available, whereas on others only the trace to table option is available. The following table shows the options available for each of the platforms PeopleSoft supports.

| <b>RDBMS</b>                   | <b>File</b> | <b>Table</b> |
|--------------------------------|-------------|--------------|
| Microsoft SQL Server           | X           | X            |
| DB2 for OS/390                 |             | X            |
| DB2 for UDB (AIX, Solaris, NT) |             | X            |
| Oracle                         | X           | X            |
| Informix                       | X           |              |
| Sybase                         | N/A         | N/A          |

The following information offers some details related to the implementation of this method to facilitate analysis of the trace results. Examine the information that pertains to your RDBMS.

---

**Note.** PeopleTools does not collect optimizer data for SQL originating from PeopleCode Actions, except in the following circumstances. If you run Oracle and Informix and run an optimizer trace to file, the system traces *all* SQL that executes after the first SQL Action executes.

---

### **Microsoft SQL Server**

On Microsoft SQL Server both the file and the table options are available for the DB Optimizer trace.

For the file option, PeopleSoft Application Engine writes the DB Optimizer traces to the following location: %TEMP%\psms<queueid><spid>.trc. To read the trace, you must use the SQL Server Profiler utility.

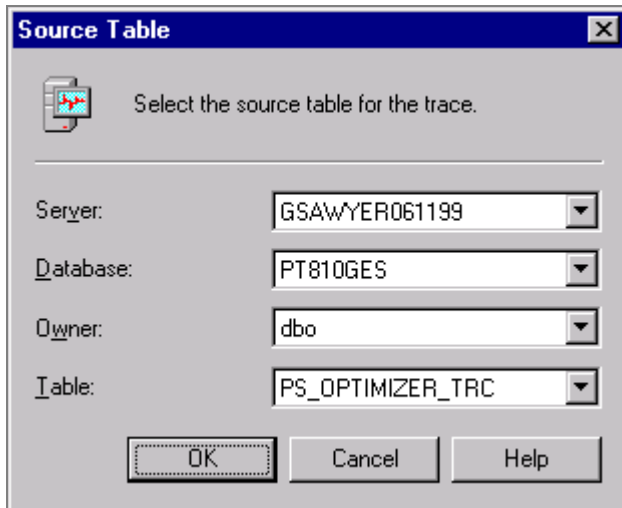
---

**Note.** The trace file will be written to the server directory when you've specified the trace on the client. If the client has %Temp% set to a drive or directory that does not exist on the server, PeopleSoft Application Engine does not generate a trace file.

---

For the table option, PeopleSoft Application Engine writes the trace data to the following table: `dbo.PS_OPTIMIZER_TRC`. PeopleTools creates the table automatically when you run the trace for the first time. The trace data written to the table is identical to the data appearing in the optimizer trace file.

You also need to use the SQL Server Profiler utility to view the optimizer results. To view the populated trace table you need to specify the current server and database on the Source Table dialog box, and the Owner and Table values must be `dbo` and `PS_OPTIMIZER_TRC` respectively. For example,



Source Table dialog box

In the trace, you will find information regarding Text, Duration, and StartTime for the following:

- Execution Plans
- Remote Procedure Calls
- INSERTs (UPDATEs, DELETEs, and SELECTs)
- PeopleSoft generated user events associating trace data with a PeopleSoft SQL identifier

If the Application Engine program aborts while you're using this trace option, you must check that PeopleSoft Application Engine was not tracing a SQL statement at the moment the program aborted. If it was tracing a SQL statement at that time, you must manually kill the trace. Otherwise the trace thread on the server will continue to run and lock the trace file, and each time that SPID gets reused by the server, new information will be appended to the locked trace file.

To stop the trace manually, submit the following command from the Query Analyzer:

```
xp_trace_destroyqueue <queueid>
```

The `<queueid>` in the file name `%TEMP%\psms<queueid><spid>.trc` is the ID corresponding to the queue used for the first SQL statement that the system profiled. Because this trace is

only designed to trace Application Engine SQL (not PeopleTools SQL), we close the queue after every statement profiled. Therefore the queue that must be destroyed may not be the Queue ID used in the trace file.

---

**Note.** If the %TEMP% variable is set to a location that does not exist, PeopleSoft Application Engine does not generate a trace file.

---

### **Oracle**

With the trace file option, Application Engine writes the trace file to the default Oracle trace directory specified on the database server. To read the trace file, use the TKPROF utility.

To use the DB Optimizer(Table) option on Oracle, a PLAN\_TABLE must exist and the statement\_id must be varchar2(254) instead of varchar2(30).

With the table option, PeopleSoft updates the trace rows as follows:

- **EXPLAIN PLAN SET STATEMENT\_ID.** PeopleSoft updates the STATEMENT ID column:

```
EXPLAIN PLAN SET STATEMENT_ID = ApplId.Section.Step.Type FOR <sqlstmt>
```

- **PLAN\_TABLE's REMARKS column.** PeopleSoft updates the REMARKS column:

```
PLAN_TABLE's REMARKS column = 'ProcessInstance-RunControlId(QueryNo)'
```

Where QueryNo is a count of how many SQL statements have been traced up to a particular point.

---

**Note.** When tracing to a table with Oracle, PeopleSoft does not perform optimizer traces on %UpdateStats and %TruncateTable unless the latter resolves into a DELETE. Alternatively, Oracle's TKPROF to file, handles both the ANALYZE and TRUNCATE commands.

---

### **Informix**

For Informix, Peoplesoft supports only the trace file option. Where the trace file gets written depends on the operating system on which your database server runs.

- **UNIX.** For UNIX, PeopleSoft Application Engine writes the plan to the sqexplain.out file. If the client program runs on the same machine as the database server, the sqexplain.out file appears in the current directory. When the current database is on another computer, the sqexplain.out file gets written to the PeopleSoft owner's directory on the remote host.
- **Windows NT.** For Windows NT, PeopleSoft Application Engine writes the plan to the following file: INFORMIXDIR%\sqexpln\username.out.



**DB2 for OS/390**

For DB2 for OS/390, PeopleSoft supports only the table option. PeopleSoft has implemented the following to facilitate this trace:

- PeopleSoft selects the max QueryNo from the PLAN\_TABLE, increments it by 1000 to avoid clashing with other processes and then increments it by 1 for every SQL statement traced.
- PeopleSoft sets the following parameter: SET REMARKS = ApplId.Section.Step.Type-RunControlId(ProcessInstance)

**DB2/UDB**

For DB2 for UNIX, PeopleSoft only supports the table option. To facilitate this trace for DB2/UNIX, PeopleSoft has implemented EXPLAIN ALL SET QUERYNO =ProcessInstance SET QUERYTAG = 'Section.Step' FOR <sql stmt>.

**Disabling the DB Optimizer Trace**

While the DB Optimizer Trace is enabled, performance may be affected. Typically, you turn on this trace only during periods in which you are collecting detailed performance metrics. When you are not tuning your performance, the DB Optimizer trace should be turned off.

To prevent an administrator, or perhaps a user, from unwittingly turning the optimizer trace on or leaving it on after doing performance tuning, you can disable the DB Optimizer trace for an entire database.

For example, suppose you have a production and a development database, you might want to enable the optimizer trace for the development database while disabling the optimizer trace for the production database.

On the PeopleTools Options page, the **Allow DB Optimizer Trace** option enables you to control whether or not a user can initiate the Application Engine DB Optimizer Trace on the database.

If **Allow DB Optimizer Trace** is selected, then the DB Optimizer Trace is a valid tracing option on the database. On the other hand, if **Allow DB Optimizer Trace** is clear then when any user attempts to initiate the optimizer trace, the following error message appears:

```
PeopleTools 8.10 - Application Engine
Copyright (c) 1988-2000 PeopleSoft, Inc.
All Rights Reserved
```

```
Optimizer trace request ignored, because disallowed in PSOPTIONS.
```

---

**Tracing Base Temporary Table Usage**

If you take advantage of temporary tables to run parallel Application Engine programs, you can trace how the system allocates, locks, and releases temporary tables during program runs.

---

**Note.** Use trace option “4” in the Trace section of the PSADMIN utility for both Application Server and Process Scheduler Server for temporary table tracing.

---

For example, if you want to trace the steps (1), SQL statements (2), and temporary table allocation (4), you would enter 7 for the TraceAE parameter. The following example shows how such a trace appears in the Application Engine Trace file (AET).

```

PeopleTools 8.40-D1 -- Application Engine
Copyright (c) 2001 PeopleSoft, Inc.
All Rights Reserved
Database: PT813GES(SqlServer)

16.21.09 Dedicating temporary tables for AETESTPROG:
16.21.11 Releasing temporary tables for AETESTPROG:
16.21.11 (1) PS_AETEST_TAO1 locked
16.21.11 2001-02-08 Tracing Application Engine program AETESTPROG Test
application
.
<snip>
.

16.21.11 Releasing temporary tables for AETESTPROG:
COMMIT
/

```

## CHAPTER 8

# Using Temporary Tables

This chapter provides an overview of temporary tables and discusses how to:

- Create temporary table instances.
- Manage temporary table instances.
- Make external calls.
- Understand temporary table performance considerations.
- View temporary table usage.

## Understanding Temporary Tables

Because Application Engine programs run in batch mode, multiple instances of the same program often execute in parallel. When this happens, there is a significant risk of data contention and deadlocks on tables. To avoid this PeopleSoft PeopleTools has a feature that enables you to dedicate specific instances of *temporary* tables for each program run.

You can also use temporary tables to improve performance. For example, if you find that multiple times during a run the program accesses a small subset of rows from a much larger table, you can insert the necessary rows into a temporary table as an initialization task. Then the program accesses the data residing in the smaller temporary table rather than the large application table. This technique is similar to reading the data into an array in memory, except that the data never leaves the database, which is an important consideration when the program employs a set-based processing algorithm.

Any number of programs, not just Application Engine programs, can use the temporary table definitions. When you specify a temporary table on the Temp Tables tab, PeopleSoft Application Engine automatically manages the assignment of temporary table instances. When Application Engine "manages" a dedicated temporary table instance, it controls the locking of the table before use and the unlocking of the table after use.

### ***Parallel Processing***

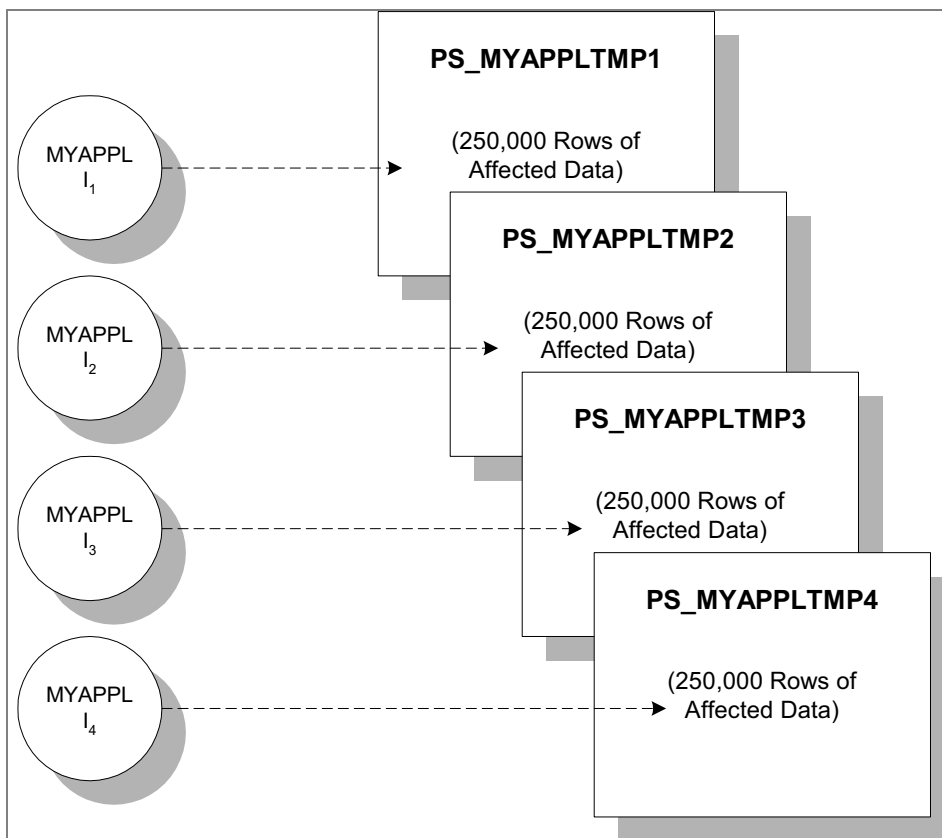
Parallel processing is used when considerable amounts of data must be updated or processed within a limited amount of time, or "batch window." In most cases, parallel processing is more efficient in environments containing multiple CPU's and partitioned data.

To use parallel processing, partition the data between multiple concurrent runs of a program, each with its own dedicated version of a temporary table (for example PS\_MYAPPLTMP). If

you have a payroll batch process, you could divide the employee data by last name. For example, employees with last names beginning with A through M get inserted into PS\_MYAPPLTMP1; employees with last names beginning with N-Z get inserted into PS\_MYAPPLTMP2.

To use two instances of the temporary table, you would define your program (say, MYAPPL) to access to one of two *dedicated* temporary tables. One execution would use A-M and the other N-Z.

The Application Engine runtime program invokes logic to pick one of the available instances. After each program instance gets matched with an available temporary table instance, the %Table meta-SQL construct uses the corresponding temporary table instance. Run control parameters passed to each instance of the MYAPPL program enable it to identify which input rows "belong" to it, and each program instance inserts the rows from the source table into its assigned temporary table instance using %Table.



Multiple program instances running against multiple temporary table instances

There is no simple switch or check box that enables you to turn parallel processing on and off. To implement parallel processing, you must complete a set of tasks in the order that they appear in the following list. With each task you must consider important details regarding your specific implementation.

- PeopleSoft Application Designer. Define and save all of the temporary table records. You don't need to run the SQL Build process at this point.

- PeopleSoft Application Engine Designer. Assign temporary tables to Application Engine programs, and set the Instance Counts dedicated for each program. Assign each temporary table to the Application Engine programs using it, set the appropriate number of Instance Counts, and employ the %Table meta-SQL construct so that Application Engine can resolve table references to the assigned temporary table instance dynamically at runtime.
- PeopleSoft PeopleTools Options. Set the global instance counts for online programs in PeopleTools Options. Set the number of temporary table instances on the PeopleTools Options page.
- PeopleSoft Application Designer. Build all the temporary table records in Application Designer (SQL Build).

## Creating Temporary Table Instances

To take advantage of multiple instances of a temporary table, you must specify Temporary Table as the record type.

PeopleSoft recommends that you insert the PROCESS\_INSTANCE field as a key on any temporary tables you intend to use with Application Engine. PeopleSoft Application Engine expects Temporary Table records to contain the PROCESS\_INSTANCE field.

---

**Note.** When all instances of a temporary table are in use and the Temp Table runtime options are set to "Continue," PeopleTools will insert rows into the base table using PROCESS\_INSTANCE as a key. If you opt *not* to include PROCESS\_INSTANCE as a key field in a temporary table, you should change the Temp Table runtime options to "Abort" in the appropriate Application Engine programs.

---

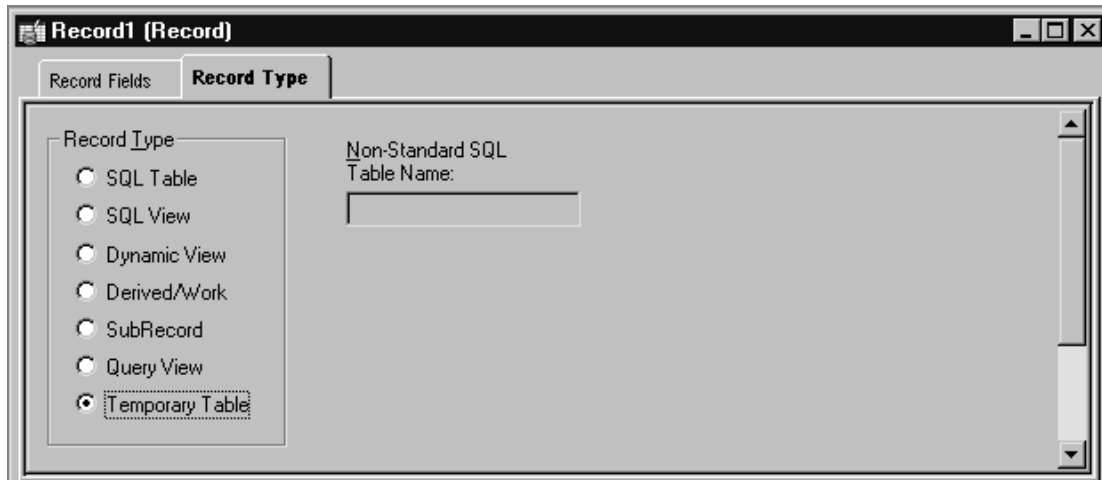
In this section, we discuss how to:

- Define temporary tables.
- Build table instances.

---

## Defining Temporary Tables

Access the Record dialog box: Record Type tab.



Record dialog box: Record Type tab

#### To define a temporary table:

1. Select File, New.  
The New Definition dialog box appears.
2. Select Record.
3. Select the Record Type tab and select the Temporary Table option.
4. Insert the PROCESS\_INSTANCE field.  
Select Insert from the toolbar. Scroll to the PROCESS\_INSTANCE field and click Insert.

## Building Table Instances

The system builds the temporary table instances at the same time it builds the base table for the record definition. When the system builds a table (as in, Build, Current Object) and the Record Type is Temporary Table, it determines the total number of instances of the temporary table based on the sum of the following two items.

- The Temporary Table Instance (Total) specified on the PeopleTools Options page.
- The total number of instances of that table dedicated within your Application Engine programs.

The system creates a maximum of 99 temporary table instances only, even if the sum exceeds 99 for a particular temporary table.

The naming convention for the temporary table instances is as follows.

<base table name>nn

Where *nn* is a number between 1 and 99, as in PS\_TEST\_TMP23.

### Temp Table Instances (Online)

Setting the appropriate number of temporary table instances is important for Application Engine processes started online from the PeopleCode CallAppEngine function. In general, the number you select should be relatively small (less than 10), so that extra instances do not affect performance.

PeopleSoft Application Engine uses this value to identify a range of temporary tables devoted to programs called by PeopleCode CallAppEngine. A randomizing algorithm balances the load for the online process that gets assigned to a temporary table devoted to online program execution.

This value provides backward compatibility for developers who took advantage of the %Table(<record\_name>, <instance\_number>) approach for *manually* managing Temporary Table locking.

### Temp Table Instances (Total)

There is also a check box for Total temporary table instances. The difference between the Total and Online will be your EMP Managed tables. If you are not using PeopleSoft EPM the total and online should be the same. These two values should always be the same except for EPM.

| PeopleTools Options                                              |                                                                                 |
|------------------------------------------------------------------|---------------------------------------------------------------------------------|
| <b>Language Settings</b>                                         |                                                                                 |
| Language Code:                                                   | English <input type="checkbox"/> Translations Change Last Updat                 |
| *Sort Order Option:                                              | Binary Sorting                                                                  |
| <b>General Options</b>                                           |                                                                                 |
| Disconnect Cursors After:                                        | 30 Background Disconnect Temp Table Instances (Total): <input type="checkbox"/> |
| <input type="checkbox"/> Multi-Company Organization              | Temp Table Instances (Online): <input type="checkbox"/>                         |
| <input checked="" type="checkbox"/> Multi-Currency               | *Maximum App Message Size: 10,000,000                                           |
| <input checked="" type="checkbox"/> Use Business Unit in nVision | Base Time Zone: PST                                                             |
| <input type="checkbox"/> Multiple Jobs Allowed                   | Last Help Context # Used: 100222                                                |
| <input checked="" type="checkbox"/> Allow DB Optimizer Trace     | *Data Field Length Checking: Others                                             |
| <input checked="" type="checkbox"/> Grant Access                 | *Maximum Attachment Chunk Size: 28,000                                          |
| <input checked="" type="checkbox"/> Platform Compatibility Mode  | Upgrade Project Commit Limit: 50                                                |
| <input checked="" type="checkbox"/> Case Insensitive Searching   |                                                                                 |
| <input type="checkbox"/> Allow NT batch when CCSID<=>37          |                                                                                 |
| Style Sheet Name:                                                | PSSTYLEDEF                                                                      |

PeopleTools Options page

#### To set temporary table instances for online processing:

1. Select PeopleTools, Utilities, Administration, PeopleTools Options.

2. Enter the number of Temp Table Instances (Total) and Temp Table Instances (Online).
3. Save the page.

---

**Note.** You can take advantage of database-specific features such as table-spaces and segmentation. For instance, you may want to use the Build process to generate a DDL script, then fine-tune the script prior to execution; or, you could place different sets of temporary tables on different table-spaces according to instance number.

---

## Managing Temporary Table Instances

Regardless of the value in the Instance Counts on either the Program Properties dialog box or the PeopleTools Options page, make sure that you have the appropriate records assigned to the appropriate programs. You also need to ensure that the SQL inside your Application Engine program contains the correct usage of the %Table construct.

The Temp Tables tab in Program Properties dialog box enables you to manage the number of different batch or dedicated temporary tables required for each program definition and the number of instances of each. You select *all* the necessary temporary table records to meet the needs of your program's logic.

---

**Note.** You must decide the instance count on the Temp Tables tab prior to building the tables in PeopleSoft Application Designer.

---

The number of temporary table instances built for a specific Temporary Table record during the SQL Build process is calculated by: value of the Temp Table Instances (Total) added to the sum of all the Instance Count values specified in the Temp Table tab for all the Application Engine programs that use that Temporary Table.

For example, assume that we have defined APPLTMPA as a Temporary Record type. Suppose that Temp Table Instances (Total) is set to 10 and APPLTMPA appears in the Temp Tables tab in the Program Properties for two Application Engine programs. In one program the Instance Count is set to 3, and in the other the Instance Count is set to 2. When you run SQL Build, PeopleTools builds a total of 15 temporary table instances for APPLTMPA.

In this section, we discuss how to:

- Assign temporary tables to programs.
- Adjust meta-SQL.

---

### Assigning Temporary Tables to Programs

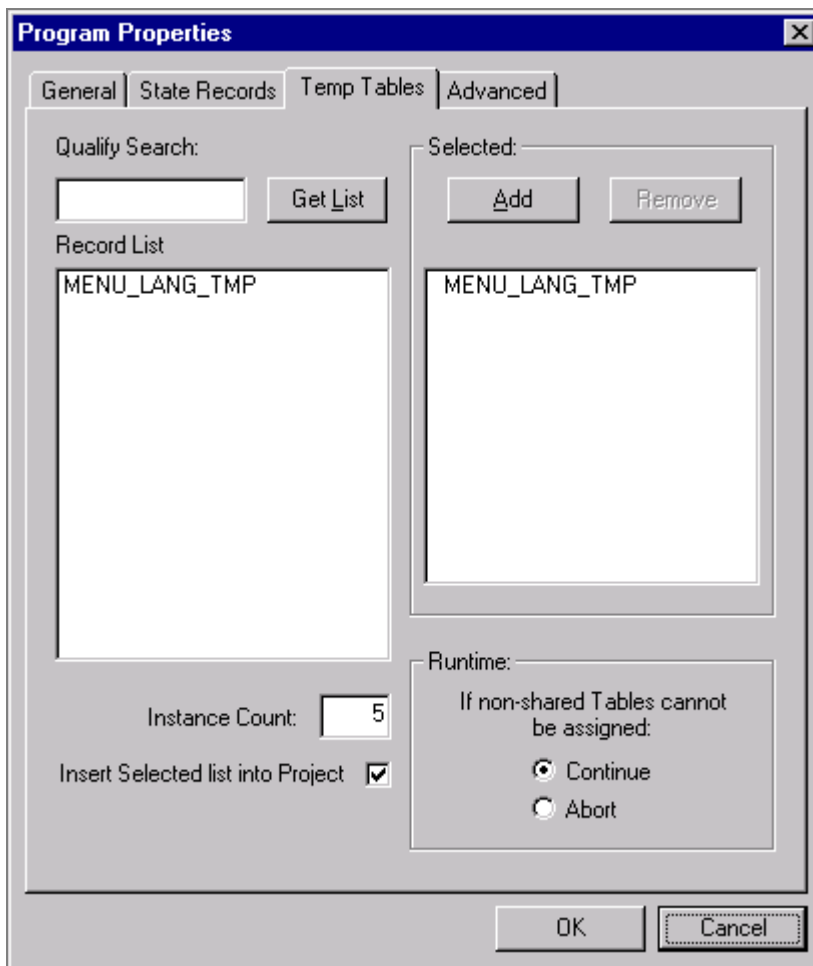
You manage the number of temporary tables assigned to a program by using the Temp Tables tab in the Program Properties dialog box.



It is the program developer's responsibility to ensure that all the necessary temporary table records are included.

The Instance Count edit box enables you to specify the number of copies of temporary tables for a program. When you change the Instance Count value and click OK, a dialog box appears to remind you that the records must be rebuilt. Anytime you change the instance counts, you need to rebuild the temporary tables to ensure that the right number of instances get created and are available for your programs.

Access Program Properties by opening an Application Engine program in PeopleSoft Application Designer. Select File, Definition Properties. Select the Temp Tables tab.



Program Properties dialog box: Temp Tables tab

---

**Note.** The concept of dedicated temporary tables is isolated to the Application Engine program run. The locking, truncate/delete from, and unlocking are designed to occur within the bounds of an Application Engine program run. Therefore, the system does not keep a temporary table instance available after the Application Engine program run completes.

---

### Runtime Allocation of Temporary Tables

Online processes have their own set of dedicated temporary tables, defined globally on the PeopleTools Options page.

When you invoke a process online, PeopleTools randomly allocates a single temporary table instance number to programX for *all* its dedicated temp table needs. While programX runs, no other program can use that instance number assigned to programX until programX has run to completion. Any other online process that happens to get the same instance value as programX will wait for the lock to be released when programX completes and the instance number is unlocked.

Batch processes are allocated temporary table instances on a record-by-record basis. The system begins with the lowest instance number available for each temporary table until all of the temporary table instances are in use. If there are not any temporary tables available and you indicated *Continue* for the *If non-shared Tables cannot be assigned*, then the base table will be used with the process instance number as a key.

When a program ends normally or is cancelled by way of Process Monitor, the system automatically releases the assigned instances.

| Condition                          | Online                                                                                                                                                                                  | Batch                                                                                                                                                                                                                                                                           |
|------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Meta-SQL for allocation            | %Table(temp-tbl)                                                                                                                                                                        | %Table(temp-tbl)                                                                                                                                                                                                                                                                |
| Runtime temporary table allocation | Psae.exe randomly assigns an instance number from the number range on your Temp Tables (Online) setting in PS Options; uses that number for all tables for that program run.            | Individually allocates instance number based on availability on a record-by-record basis. Psae.exe begins with the lowest instance number available for each temporary table until all of the instances are in use.                                                             |
| No temp tables are free            | For a particular record, if the instance is currently in use, and the program is set to Continue, then the psae.exe queues the program until the assigned instance number becomes free. | If the program is set to <i>Continue</i> , the system uses a shared base table.<br><br>If the program is set to <i>Abort</i> , then the system terminates the program.<br><br>Never queues for a table.                                                                         |
| Clear temp table (initially)       | Yes, when program instance becomes available.                                                                                                                                           | Yes, when assigned.                                                                                                                                                                                                                                                             |
| Locked                             | Lock is on when the program is loading into memory.                                                                                                                                     | Lock is on when the program is loading into memory. For restartable programs, the temp tables remain locked <i>across</i> restarts until the program has completed successfully or until the temp tables are manually released using Process Monitor or the Manage Abends page. |

| <b>Condition</b> | <b>Online</b>                                                                                                                                                                                        | <b>Batch</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Un-locked        | Temp tables unlocked on completion of program.<br><br>In the event of a "kill" or a "crash," the tables remain locked, and the tables must be freed using Process Monitor or the Manage Abends page. | If restart is <i>disabled</i> the temp tables are unassigned automatically in the event of a controlled abend.<br><br>If you cancel a process using Process Monitor, PeopleTools frees the temporary tables automatically.<br><br>When you use the Manage Abends page, you must click the Temp Tables button corresponding to the correct Process Instance, and then click the Release button on the Temporary Tables dialog box.<br><br>Cancel from Process Monitor frees tables. |

---

**Note.** When you have manually released the temporary tables from their locked state, you lose any option to restart the program run.

---

### ***Sharing Temporary Table Data***

Dedicated temporary tables do not remain locked across Process Instances. If sequential Application Engine programs need to share data by way of temporary tables, a parent Application Engine program should call the programs that share data.

---

## **Adjusting Meta-SQL**

A critical step in implementing parallel processing is to make sure that you've included all of the appropriate meta-SQL within the code that your Application Engine program executes. This section contains the specific meta-SQL constructs that you should address.

### ***Referencing Temporary Tables (%Table)***

To reference a dedicated temp table, you must use

```
%Table (record)
```

You can reference any table with %Table, but only those records defined as Temporary Tables get replaced with a dedicated instance table by PeopleSoft Application Engine. When you are developing programs that take advantage of %Table, keep the following items in mind:

- Your choice of indexes on temporary tables is an important consideration. Depending on the use of the temporary table in your program and your data profile, the system indexes may be sufficient. On the other hand, a custom index may be needed instead, or perhaps no indexes are necessary at all. It is worth considering these issues when designing your application. You want to define indexes and SQL that will perform well in most situations, but individual programs or environments may require additional performance tuning during implementation.

---

**Note.** The default table name refers to PS\_recname, where PS\_recname1,2,... refer to the dedicated temporary tables.

---

As PeopleSoft Application Engine resolves any %Table, it checks an internal array to see if a temporary table instance has already been chosen for the current record. If so, then PeopleSoft Application Engine substitutes the chosen table name. If not, as in when a record does not appear in the temp table list for the program, then PeopleSoft Application Engine uses the base table instance (PS\_recname) by default. Regardless of whether %Table is in PeopleCode SQL or in an Application Engine SQL Action the program uses the same physical SQL table.

### ***Populating Temporary Table Process Instance With the Process Instance***

All temporary tables should be keyed by Process Instance as a general rule. Also, if you have opted to use the "Continue" option when batch/dedicated tables can't be assigned, Process Instance is *required* as a key field. The current Process Instance is automatically put into the state record, but when you Insert rows into your temporary tables you must supply that Process Instance.

```
%ProcessInstance OR %Bind(PROCESS_INSTANCE)
```

This meta-SQL returns the numeric (unquoted) Process Instance.

The Process Instance value is always zero for programs initiated with CallAppEngine. This is because the program called with CallAppEngine runs "in process", that is, it runs within the same unit of work as the component with which it is associated.

If you are using dedicated tables, and have elected to continue if dedicated tables can't be assigned, then all of your SQL references to dedicated temporary tables must include PROCESS\_INSTANCE in the WHERE clause.

### ***Clearing Temporary Tables (%TruncateTable)***

You do not need to delete data from a temporary table manually. The temporary tables are truncated automatically when they are assigned to your program. If the shared base table has been allocated because no dedicated instances were available, then Application Engine performs a delete by process instance instead of performing a truncate. In such a case, the PROCESS\_INSTANCE is required as a high-level key.

You can perform additional deletes of temporary table results during the run, but you must include your own SQL Action that does a %TruncateTable. If the shared base table has been allocated because no dedicated instances were available, then %TruncateTable is replaced with a delete by process instance instead of a truncate.

---

**Note.** You should always use %TruncateTable to perform a mass delete on dedicated temporary tables, especially if the Continue option is in effect.

---

Even if you have elected to abort if a dedicated table cannot be allocated, you may still use %TruncateTable meta-SQL with dedicated temporary tables. %TruncateTable resolves to either a TRUNCATE or a DELETE by process instance, as needed.

Keep in mind that the argument of %TruncateTable is a table name instead of a record name. As a result, you must code your SQL as shown in the following example.

```
%TruncateTable(%Table(<rename literal or bind>))
```

---

**Note.** You should avoid hard-coded table names inside %TruncateTable since they preclude the possibility of concurrent processing.

---

## Making External Calls

When you call one Application Engine program from another, the assignment of dedicated tables for the called, or "child," program, occurs only if the calling, or parent, program is in a state where a commit can occur immediately.

PeopleTools enables you to commit immediately so that PeopleSoft Application Engine can commit the update it performs to lock the temporary table instance. Otherwise, no other parallel process could perform any assignments. In general, this means that you should issue a commit just prior to the Call Section Action.

While making external program calls, note the following:

- If the situation is suitable for a commit then the temporary table assignment and the appropriate truncates occur.
- If the situation is not suitable for a commit *and* the called program is set to *continue* if dedicated tables cannot be allocated, then the base tables are used instead, and a delete by process instance is performed.
- If the situation is not suitable for a commit *and* the called program is set to *abort* if dedicated tables cannot be allocated, then program execution terminates. This would actually reflect an implementation flaw that you would need to correct.
- If the called Application Engine program shares temporary tables with the calling program, this is allowed. Common temporary tables are the way you share data between the calling and called programs. PeopleSoft Application Engine locks only instances of temporary tables that have not already been used during the current program run. Temporary tables that already have an assigned instance will continue to use that same instance.

Also when you make calls to external programs, there are considerations for batch and online program runs to keep in mind.

### **External Calls in Batch Mode**

For batch runs, in the Program Properties of the root program list all of the temporary tables that any called programs or sections use. This ensures that the tables get locked sooner and as a single unit. This approach can improve performance, and it ensures that all the tables required by the program are ready before execution starts.

### **External Calls in Online Mode**

If the online program run is designed to use any temporary tables at *any* point during the CallAppEngine unit of work, then the root program must have at least 1 temp table specified in the Program Properties dialog box. This is true even if the root program doesn't use any temporary tables. This is required so that the system locks the instance number early on to avoid an instance assignment failure after the process has already started processing.

All temporary tables used by a specific program, library, or external section must be specified in that program to ensure that the system issues Truncates (deletes) for the tables being utilized.

If no temporary tables appear in the root program properties, and PeopleSoft Application Engine encounters a %Table reference for a temporary table record, the following error appears:

```
Online AE Process - Invalid attempt to process Temporary Table <record name>
```

---

## **Sample Implementation**

The following scenario describes the runtime behavior of PeopleSoft Application Engine and temporary tables.

Assume you have Program A and Program B and three temporary table definitions: PS\_TMPA, PS\_TMPB, PS\_TMPC. Values on the Temporary Tables tab in Program Properties dialog box for each program are as follows:

- *Program A.* PS\_TMPA and PS\_TMPB are specified as the dedicated temporary tables, and the Instance Count is 4.
- *Program B.* PS\_TMPB and PS\_TMPC are specified as the dedicated temporary tables, and the Instance Count is 3.

After you run SQL Build in PeopleSoft Application Designer, the following inventory of temporary tables appears in the database.

| <b>PS_TMPA</b> | <b>PS_TMPB</b> | <b>PS_TMPC</b> |
|----------------|----------------|----------------|
| PS_TMPA1       | PS_TMPB1       | PS_TMPC1       |
| PS_TMPA2       | PS_TMPB2       | PS_TMPC2       |
| PS_TMPA3       | PS_TMPB3       | PS_TMPC3       |
| PS_TMPA4       | PS_TMPB4       |                |
|                | PS_TMPB5       |                |

| <i>PS_TMPA</i> | <i>PS_TMPB</i> | <i>PS_TMPC</i> |
|----------------|----------------|----------------|
|                | PS_TMPB6       |                |
|                | PS_TMPB7       |                |

Because the Instance Count for Program A is 4, the system builds four instances of PS\_TMPA and PS\_TMPB for Program A to use. Because the Instance Count for Program B is 3, the system builds an additional three instances of PS\_TMPB and three instances of PS\_TMPC for Program B to use.

Notice that because Program A and Program B are "sharing" PS\_TMPB there are seven instances. The system derives this total by adding the Instance Count value from all the programs that share a particular temporary table instance. In this case, the four from Program A and the three from Program B combine to require a total of seven instances of PS\_TMPB to be built.

Given that this collection of temporary tables exists in your database, let's say that you start Program A. At runtime, PeopleSoft Application Engine examines the list of temporary tables dedicated to Program A, and assigns the first available instances to Program A. Then, assuming that no other programs are running, PeopleSoft Application Engine assigns PS\_TMPA1 and PS\_TMPB1 to Program A.

Now suppose that shortly after you started Program A another user starts Program B. Again, PeopleSoft Application Engine examines the list of temporary tables dedicated to Program B and assigns the first available instances. In this scenario, PeopleSoft Application Engine assigns PS\_TMPB2 and PS\_TMPC1 to Program B. Because Program A is already using PS\_TMPB1, the system assigns PS\_TMPB2 to Program B.

The system assigns records, such as TMPA, to programs. The base tables, such as PS\_TMPA, are also built, by default, in addition to the dedicated temporary instances. If the Program Properties dialog box setting for Temp Tables is set to *continue* when no instances are available, the system uses the base table instead of the dedicated instance.

## Temporary Table Performance Considerations

Before implementing dedicated temporary tables for parallel processing, note these performance considerations.

---

### Initial Estimates

Keep in mind that if you find that you need more temporary table instances after you've entered production, you must rebuild all of your temporary tables so that the database reflects the proper inventory of instances. While the build process runs, users can't access the database. Because of this, it is important to spend time deriving adequate estimates as to the number of temporary tables required.

## Online Temporary Table Allocation

There is a physical table within the database, named PS\_AEONLINEINST, designed to store online temporary table instance usage.

If you notice performance issues related to online Application Engine program runs, enable the Application Engine SQL and Timings trace.

If the following SQL command requires more time than normal to complete, this is a good indication that there aren't enough online temporary instances defined on PeopleTools Options.

```
UPDATE PS_AEONLINEINST ...
```

## Viewing Temporary Table Usage

In this section, we discuss how to:

- View temporary table usage.
- View online instance usage.
- View the temp table usage warning messages.

If you have implemented temporary tables for parallel Application Engine program runs, you use the Temporary Table Usage page to find out how the system allocates temporary tables to your programs. To access this page select PeopleTools, Application Engine, Review Temporary Table Usage.

Parallel processing is designed to be a performance enhancing option. However, if the demand for temporary table instances consistently exceeds the current supply, your performance suffers. Also, in other situations your inventory of temporary table instances may far outnumber demand. Here, you may consider reducing the number of instances provided to conserve system resources.

| Temp Table Usage by Record |                      | Temp Table Settings by Program |                  |                      |                                        |
|----------------------------|----------------------|--------------------------------|------------------|----------------------|----------------------------------------|
| Temp Table Usage by Record |                      |                                |                  |                      |                                        |
| Filter List by             |                      |                                |                  |                      |                                        |
| Record (Table) Name:       | <input type="text"/> | <input type="text"/>           | Program Name:    | <input type="text"/> | <input type="button" value="Refresh"/> |
| Lock Details               |                      |                                |                  |                      |                                        |
| Record (Table) Name        | Program Use Count    | Total Instances                | Locked Instances | Unused Instances     | View Programs                          |
| AEXT_TAO                   | 1                    | 5                              | 0                | 5                    | <a href="#">View Programs</a>          |
| QE_AEXT_TAO                | 1                    | 5                              | 0                | 5                    | <a href="#">View Programs</a>          |
| QE_AETEST_TAO              | 1                    | 5                              | 0                | 5                    | <a href="#">View Programs</a>          |

Temp Table Usage by Record page



You can view usage by the Record (Table) Name or you can view by Program Name. You use the lookup buttons to select a particular record or program. To refresh the contents of the page, click the **Refresh** button.

This page shows you the following metrics for evaluating your inventory and allocation of temporary tables:

|                          |                                                                                                    |
|--------------------------|----------------------------------------------------------------------------------------------------|
| <b>Program Use Count</b> | Shows the instance count of listed program.                                                        |
| <b>Total Instances</b>   | Shows the total number of instances of a temporary table that exist.                               |
| <b>Locked Instances</b>  | This value indicates the current number of instances that they system has locked for program runs. |
| <b>Unused Instances</b>  | This value indicates the current number of instances that are available for use.                   |

Temp Table Usage by Record
Temp Table Settings by Program

Temp Table Settings by Program

Filter List by

Record (Table) Name:  
Program Name:

Settings Details

| Program Name | Total Instances | Batch Only | Abort Flag | Disable Restart | View Records                 |
|--------------|-----------------|------------|------------|-----------------|------------------------------|
| AETESTEXT    | 5               | N          | N          | N               | <a href="#">View Records</a> |
| QE_AETESTEXT | 5               | N          | N          | N               | <a href="#">View Records</a> |
| QE_AETESTPRG | 5               | N          | N          | Y               | <a href="#">View Records</a> |

Temp Table Setting by Program page

From this page you can access the Manage Abends page and then to the Process Monitor if the Application Engine process was started in Process Monitor.

---

## Viewing Online Instance Usage

To view the online temporary table usage, select PeopleTools, Application Engine, Review Online Instance Usage.

| <b>Online Instance Usage</b>    |                               |
|---------------------------------|-------------------------------|
| <b>Locks Issued by Instance</b> |                               |
| <b>Temp Table Instance</b>      | <b>Number of Locks Issued</b> |
| 0                               | 0                             |
| 1                               | 0                             |
| 2                               | 0                             |
| 3                               | 0                             |
| 4                               | 0                             |
| 5                               | 0                             |
| 6                               | 0                             |
| 7                               | 0                             |
| 8                               | 0                             |
| 9                               | 0                             |
| 10                              | 0                             |
| 11                              | 0                             |
| 12                              | 0                             |
| 13                              | 0                             |

Online Instance Usage page

## Viewing the Temp Table Usage Warning Message

If an Application Engine batch program is unable to get a dedicated temporary table because all instances are locked, but it can use the base table, the system issues a warning. However, if the program has been set to abort when a dedicated instance is not available, then the program aborts even if the base table can be used.

You could see the warning message in two ways:

- A warning message appears in the standard output of the process. When running from the command prompt, the message appears in that window. When the program is running on a server through PeopleSoft Process Scheduler, the output is sent to the standard status file, which you can access using Process Monitor.
- A warning message appears in the AET trace file if a dedicated temporary table instance can't be locked because none are available. This message appears in the trace file regardless of the trace settings you've selected.

### Resolving the Warning Message

If you see the warning regarding base temporary table usage, this means either there aren't enough temporary table instances defined or there are some locked instances that must be released.

When a restartable process abends, the temporary tables stay locked to enable a smooth restart. However, if you don't want to restart the process, then the locked temporary tables must be released. When you cancel the process using Process Monitor the release of locked

temporary tables occurs automatically. If the process wasn't launched through PeopleSoft Process Scheduler, Process Monitor does not track the process. Because of this, you must use the Manage Abends page to release temporary tables used by processes invoked outside of PeopleSoft Process Scheduler.



# 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 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 an 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'Entreprise).

**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) uncancelled 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 Claeys 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 MgrGetObject, 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 rehire.



**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 `mksyd` 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 differred. 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.

***VEETS-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

## %

- %Bind 4-1
- %ClearCursor 4-12
- %Execute 4-12
- %ExecuteEdits 4-4
- %Next 4-13
- %Previous 4-13
- %RoundCurrency 4-14
- %Select 4-6
- %SelectInit 4-7
- %SQL 4-7
- %Table 4-8, 8-9
- %TruncateTable 4-8, 8-10
- %UpdateStats 4-9

## A

- abends 5-21, 5-26
- actions 2-25
  - call section 2-31
  - DO Select 2-29
  - DO Until 2-30
  - Do When 2-28
  - Do While 2-29
  - execution hierarchy 2-34
  - inserting 2-36
  - log Message 2-33
  - No Rows 2-27
  - PeopleCode 2-31, 2-34, 4-19
  - program flow 2-28
  - ReUse 2-26
  - SQL 2-26
- adding COBOL copybook 6-1
- AESection Object 4-21
- AI parameter 5-8
- API
  - PeopleTools 4-27
- Application Engine
  - abends 5-26
  - action types 2-37
  - actions 2-25
  - Actions *See* actions
  - advanced properties 2-18
  - commit *See* commit
  - CommitWork 4-28
  - daemon program 1-4
  - debugging 5-10
  - debugging options 5-12

- defined i
- dynamic SQL 4-30
- execution precedence 2-22
- file layout object 4-24
- general properties 2-14
- IF, THEN logic 4-18
- inserting actions 2-36
- introduction 1-1
- locating sections 2-20
- macros 4-12
- managing programs 5-1
- meta-SQL 1-1, 4-1
- no rows 2-27
- PeopleCode
  - action execution hierarchy 4-19
  - in loops 4-20
  - math functions 4-29
  - SQL objects 4-30

- PeopleCode notes 4-28
- PeopleCode variables 4-19
- program flow 2-28
- program properties 2-14
- RemoteCall function 4-25
- restarting programs 5-18
- reusing SQL 3-5
- reusing statements 2-26
- running programs 5-1
- section object 4-21
- section properties 2-21
- set processing 3-7
- sharing state records 2-33
- SQL actions 2-26
- state records 2-14
- statement timings trace 7-6
- system variables *See* system variables, Application Engine
- terminology 1-2
- testing a program 2-12
- tracing 7-1
  - enabling 7-1
- transform program 1-6
- Tuxedo requirements 5-2
- viewing definitions 2-2
- viewing program flow 2-4

Application Engine programs

- abnormal ends 5-21
- inserting sections 2-19

- inserting steps 2-23
- online calls to 4-21
- opening 2-11
- renaming 2-11
- restarting from command line 5-22
- set processing examples 3-12
- step properties 2-24
- using Process Scheduler 5-2

## B

- batch server
  - Tuxedo requirements 5-2
- batch timings
  - table 7-12
- BATTIMES.SQR 7-14
  - invoking 7-16
- bulk insert 3-6

## C

- caching
  - Application Engine 5-25
  - server 5-25
- call section
  - program properties 2-33
- call section actions 2-31
- call section references
  - finding 2-20
- CallAppEngine 4-21
  - defining global variables 4-17
  - events 4-23
  - fieldchange event 4-24
  - passing parameters 4-17
  - process instance 4-23
  - save event 4-23
- CallAppEngine function 5-6
- calling Application Engine programs from COBOL 6-1
- command line
  - Application Engine 5-7
  - tracing Application Engine programs 7-1
- commit 3-4
- CommitWork 4-28
- copybook values
  - assigning 6-2
- CP parameter 5-8
- creating Application Engine programs 2-11

## D

- daemon program type 1-4
  - starting from Process Scheduler 1-5
  - starting from the command line 1-5
- DB Optimizer trace 7-17

- DB2 UDB 7-20
  - Disabling 7-20
  - Informix 7-19
  - Microsoft SQL Server 7-17
  - Oracle 7-19
  - OS/390 7-20
- DBFLAGS parameter 5-9
- DEBUG parameter 5-9
- debugging
  - Application Engine 5-10
- disabling restart 5-24
- Do Select 2-29
- DO Select 2-29
  - considerations 2-30
- DO Until 2-30
- Do When 4-28
- DO When 2-28
- DO While 2-29
- DR parameter 5-9
- dynamic SQL 4-30
- Dynamic SQL 4-29

## F

- file layout object 4-24
- FP (filepath) parameter 5-10

## I

- I parameter 5-8

## L

- log message actions 2-33

## M

- macros
  - Application Engine 4-12
    - %ClearCursor 4-12
    - %Execute 4-12
    - %Next 4-13
    - %RoundCurrency 4-14
- managing Application Engine programs 5-1
- managing temporary tables 8-6
- meta-SQL
  - adjusting for temporary tables 8-9
    - Application Engine 4-1
      - %Bind 4-1
      - %ExecuteEdits 4-4

- %Select 4-6
- %SelectInit 4-7
- %SQL 4-7
- %Table 4-8
- %TruncateTable 4-8
- %UpdateStats 4-9
- referencing temporary tables 8-9

## N

- No Rows 2-27

## O

- OT parameter 5-10

## P

- parallel processing
  - temporary tables 8-1
- PeopleBooks
  - printed, ordering ii
- PeopleCode
  - AESection object 4-21
  - APIs 4-27
  - CallAppEngine 5-6
  - loops 4-20
  - sequence numbering 4-29
  - state records 4-18
- PeopleCode actions
  - tracing 7-9
- performance
  - temporary tables 8-13
- process request
  - application engine 5-5
  - pages for Application Engine 5-4
- Process Scheduler
  - using for Application Engine 5-2
- program flow actions 2-28
- program flow view 2-5
- programs
  - call section 2-31
  - finding sections within 2-20
  - procedures 2-11
  - properties 2-14
  - sections 2-19
  - steps 2-23
- properties
  - Application Engine actions 2-36
  - temp tables 2-16

- PS\_BAT\_TIMINGS\_DTL 7-2, 7-13
- PS\_BAT\_TIMINGS\_LOG 7-2, 7-13
- psae <parmfile> parameter 5-9
- PTCCBLAE.CBL copybook 6-1
- PTPECOBL 4-25

## R

- R parameter 5-8
- RemoteCall 4-25
  - commit 4-27
  - PTPECOBL 4-25
- restart
  - Application Engine programs 5-18
  - deciding when to 5-19
  - disabling 5-24
  - how it works 5-18
  - program level 5-20
  - section level 5-20
  - step level 5-20
- restartable 2-30
- restarting Application Engine programs
  - from a process request page 5-23
- ReUse 2-26
- Rowsets 4-29

## S

- section filtering 2-7
- section properties 2-21
- sections
  - Application Engine 2-19
  - execution precedence 2-22
  - finding within current program 2-20
  - inserting 2-19
  - list of references 2-20
  - locating 2-20
  - navigating to 2-20
- set processing 3-7
  - examples 3-12
  - planning 3-8
  - platform issues 3-15
  - tips 3-10
- sharing state records 3-2
- SQL
  - reusing 3-5
  - set processing 3-7
- SQL actions
  - bulk insert 3-6
  - ReUse 3-5
- SQL counts 7-7
- SQL objects
  - Application Engine 4-30
- starting
  - Application Engine programs 5-1
- state records 2-14

- PeopleCode 4-18
- sharing 2-33, 3-2
- using 3-1
- statement timings 7-6
- table 7-12
- Step
  - traces 7-5
- steps 2-23
  - defining 2-23
  - inserting 2-23
  - properties 2-24
- system variables
  - Application Engine 4-14

## T

- temporary tables 2-16
  - adjusting meta-SQL for 8-9
  - assigning 8-6
  - batch mode 8-12
  - building 8-3
  - calling other programs 8-11
  - clearing 8-10
  - instances 8-3
  - managing 8-6
  - online mode 8-12
  - performance considerations 8-13
  - set processing 3-8

- using 8-1
- testing an Application Engine program 2-12
- TOOLBINSRV 5-2
- TOOLSTRACEPC parameter 5-9
- TOOLSTRACESQL parameter 5-9
- trace
  - batch timings 7-6
  - DB Optimizer 7-17
  - locating Application Engine trace 7-4
  - PeopleCode actions 7-9
  - PeopleCode built-ins and methods 7-10
  - understanding results 7-4
- TRACE parameter 5-9
- TraceAE 7-2, 7-3
- tracing
  - Application Engine program steps 7-5
  - Application Engine programs 7-1
  - configuration files 7-2
  - configuration options 7-3
  - enabling Application Engine traces 7-1
- transformation program 1-6
- triggering CallAppEngine 4-23

## V

- views
  - switching between definition and program flow 2-5