PeopleTools 8.4: PeopleSoft Business Interlink Runtime Plug-in Programming Guide

PeopleSoft®

PeopleTools 8.4: PeopleSoft Business Interlink Runtime Plug-in Programming Guide

SKU Tr84BIS-B 0302

PeopleBooks Contributors: Teams from PeopleSoft Product Documentation and Development.

Contents

Chapter 8

Understanding The Business Interlink Methods

Chapter 9

Writing the Design-Time Functionality Using Dynamic Catalog Methods

PeopleSoft Business Interlink Runtime Plug-in Programming Guide Preface

This PeopleBook covers the concepts of Business Interlinks, which allows a PeopleSoft application to integrate with external systems. It discusses how programmers create Business Interlink Runtime Plug-ins. This PeopleBook is written for programmers who will be integrating external systems with PeopleSoft by writing Business Interlink Runtime Plug-ins.

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. For more information about effective dates, see **Understanding Effective Dates in *Using PeopleSoft Applications.***
EmplID (employee ID)	Unique identification code for an individual associated with your organization.

Field	*Definition*
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.
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.
	For more information about the Report List page, the Process List page, and the Process Scheduler, see Process Scheduler Basics in the PeopleTools documentation.
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 DETAIL_CALENDAR1.
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

Introduction to Business Interlinks

Business Interlinks enable you to perform component-based, realtime integration from PeopleSoft to external systems. Business Interlinks do this by creating synchronous transactions that allow PeopleSoft applications to pass data to and receive data from the external system in real time. You can use Business Interlinks to integrate PeopleSoft with external systems, with another PeopleSoft application, and systems on the internet web.

A transaction consists of a named command with optional named and typed inputs and outputs. The associated external system or the Business Interlink Plug-in understands this command.

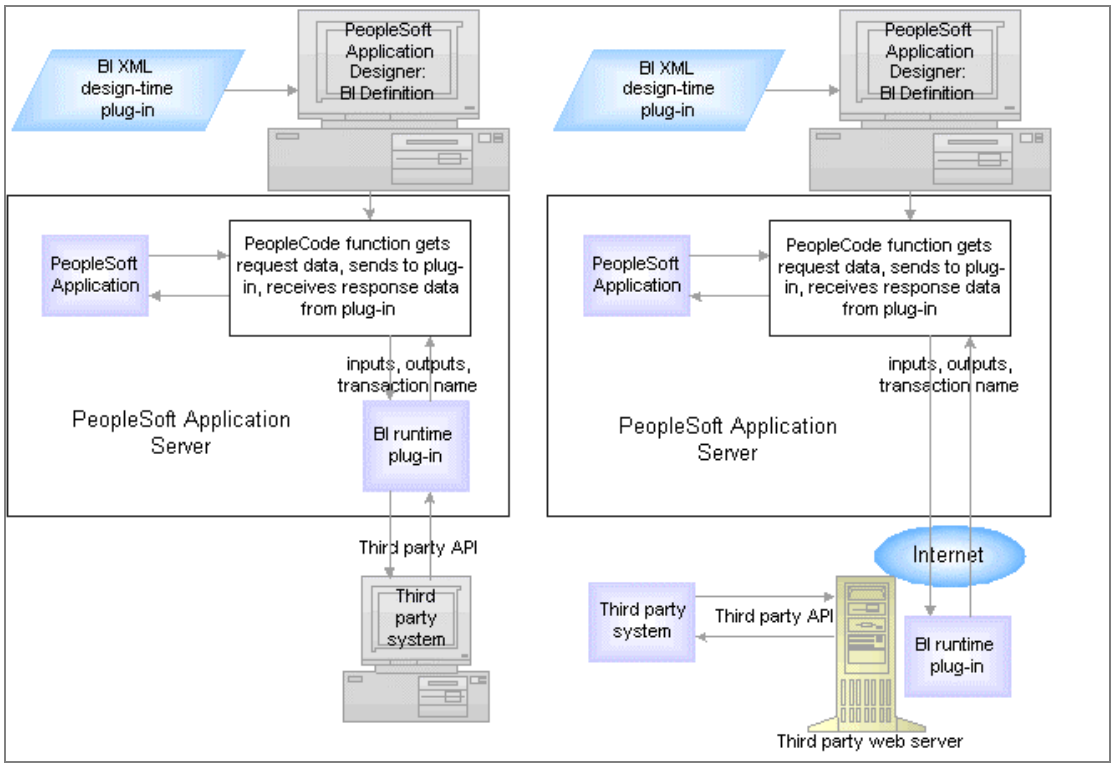
What is the Business Interlink Architecture

The Business Interlink Architecture provides a plug-in framework for PeopleSoft applications to invoke third party APIs over the Internet. Different vendors support different methods for exposing their APIs, including object technologies such as COM, COBRA, EJB; programming language specific interfaces for C or C++; or interfaces based on HTTP and XML. The Business Interlink framework provides a consistent framework for application developers to invoke external applications across this wide variety of technologies.

When a business event triggers the execution of a Business Interlink, the Component Processor synchronously calls the Business Interlink Processor, which in turn invokes the appropriate Business Interlink plug-in. The plug-in provides a wrapper around the third party API and is designed to support any type of interface binding (COM, CORBA, EJB, XML) exposed by the third-party interface. The third-party software could be hosted on the same machine as the PeopleSoft Internet Application Server, or on a separate machine on the other side of the world, invoked over the Internet.

Running a Business Interlink

The following diagram shows the typical Business Interlink architecture, using an XML design-time plug-in and a runtime plug-in.



Business Interlink Architecture

When the Business Interlink Object is executed, the following actions take place:

1. A PeopleSoft Application triggers a PeopleCode event. For example, a user might, in a PeopleSoft page labeled “Shipping Time and Cost,” enter input values needed to calculate shipping time, and then press a button labeled “Calculate.”

2. The triggered event (pressing the “Calculate” button) passes the input values to a PeopleCode program, and then executes the PeopleCode program.

3. The PeopleCode program creates an Interlink Object, which contains the transaction name and inputs, and passes the Interlink Object to the Business Interlink runtime plug-in. The runtime plug-in can be located on:

   - A web server
   - The same PeopleSoft Application Server as the PeopleSoft Application.

4. The Business Interlink runtime plug-in provides the implementation of the transactions and/or data operations. It calls the external software system though its API, passing the input values from the Business Interlink Object. This external system can be located on:

   - An external server.
   - A web server.
   - The same PeopleSoft Application Server as the PeopleSoft Application.

5. The third party system performs operations based on those input values, and returns output values thought its API to the runtime plug-in. These operations could be, for example, executing functions in the external system, or performing operations on a database in the external system.

6. The Business Interlink runtime plug-in assigns output values to the Business Interlink Object (if there are outputs). If there are outputs, the PeopleCode program can assign the output values to PeopleSoft variables. For example, in a PeopleSoft page labeled "Shipping Time and Cost," the output values could then be displayed upon that page.

Creating a Business Interlink

To allow users to create and run Business Interlinks, developers must perform the following tasks. This manual describes the tasks that are in **bold**.

1. A developer designs the shape of a transaction(s) (inputs, outputs, name). For example, the action could be telling UPS to calculate the cost of shipping.

2. A developer writes a Business Interlink design-time plug-in that implements the shape of the transaction(s). This plug-in is written in the XML markup language.

3. A developer writes a Business Interlink runtime plug-in to implement the transaction: passing the inputs to an external system and receiving the outputs form the external system.

4. The design-time plug-in and the runtime plug-in are deployed for use by PeopleSoft applications.

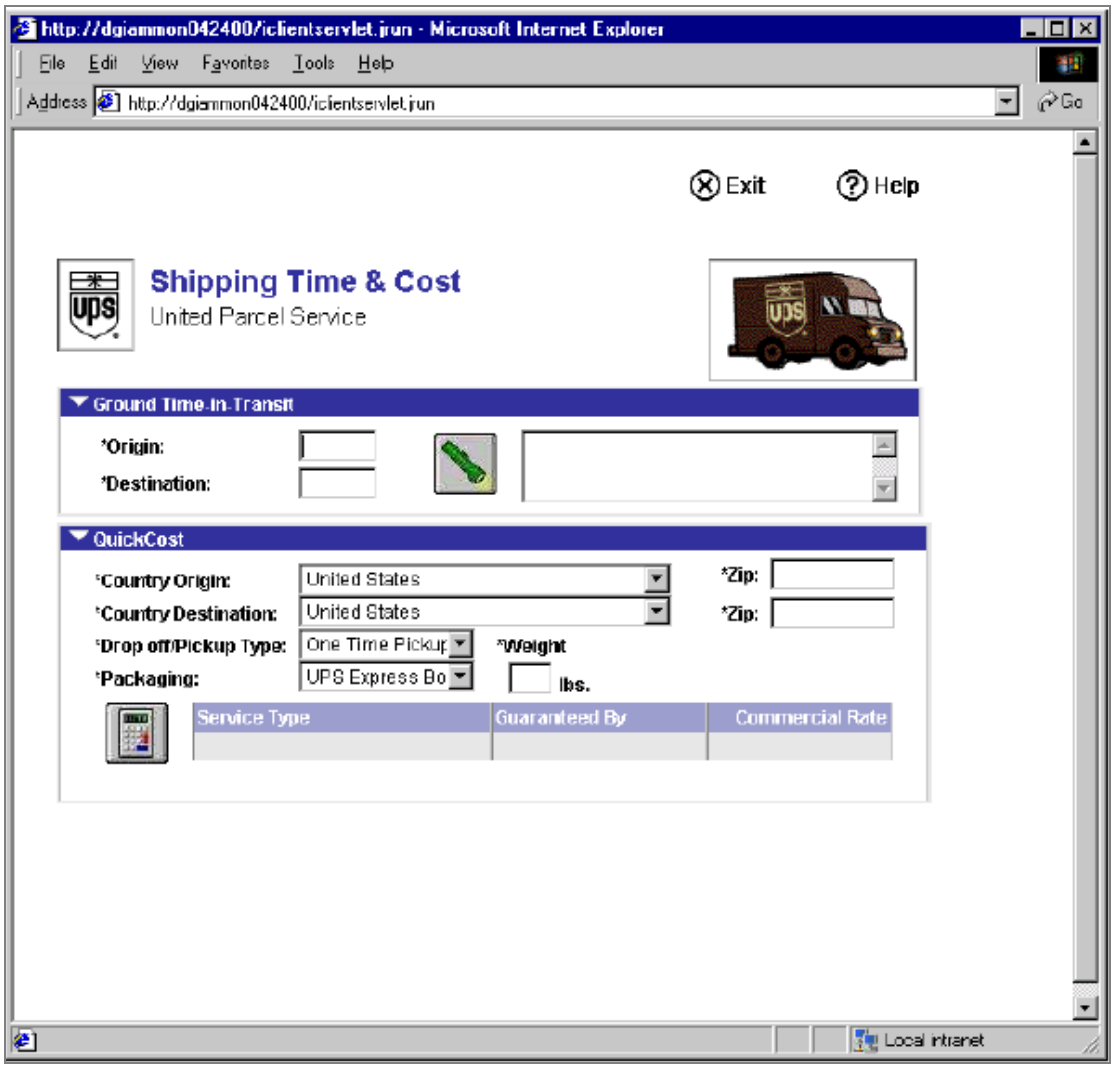
5. A PeopleSoft Application Developer creates a Business Interlink Definition, which creates a specific shape for a particular PeopleSoft application. For example, the application might be created, or modified, to be able to connect to UPS and calculate shipping costs.

6. A PeopleSoft Application Developer writes a PeopleCode program to call the Business Interlink object that is created for the Business Interlink definition.

7. A user can now use the PeopleSoft application to run the Business Interlink. For example, the use can now connect to UPS and calculate shipping costs.

Example 1: Calculating UPS Shipping Costs

The Shipping Time & Cost example shows how Business Interlinks can be used. Shipping Time & Cost use the Business Interlink plug-ins that were written for UPS.

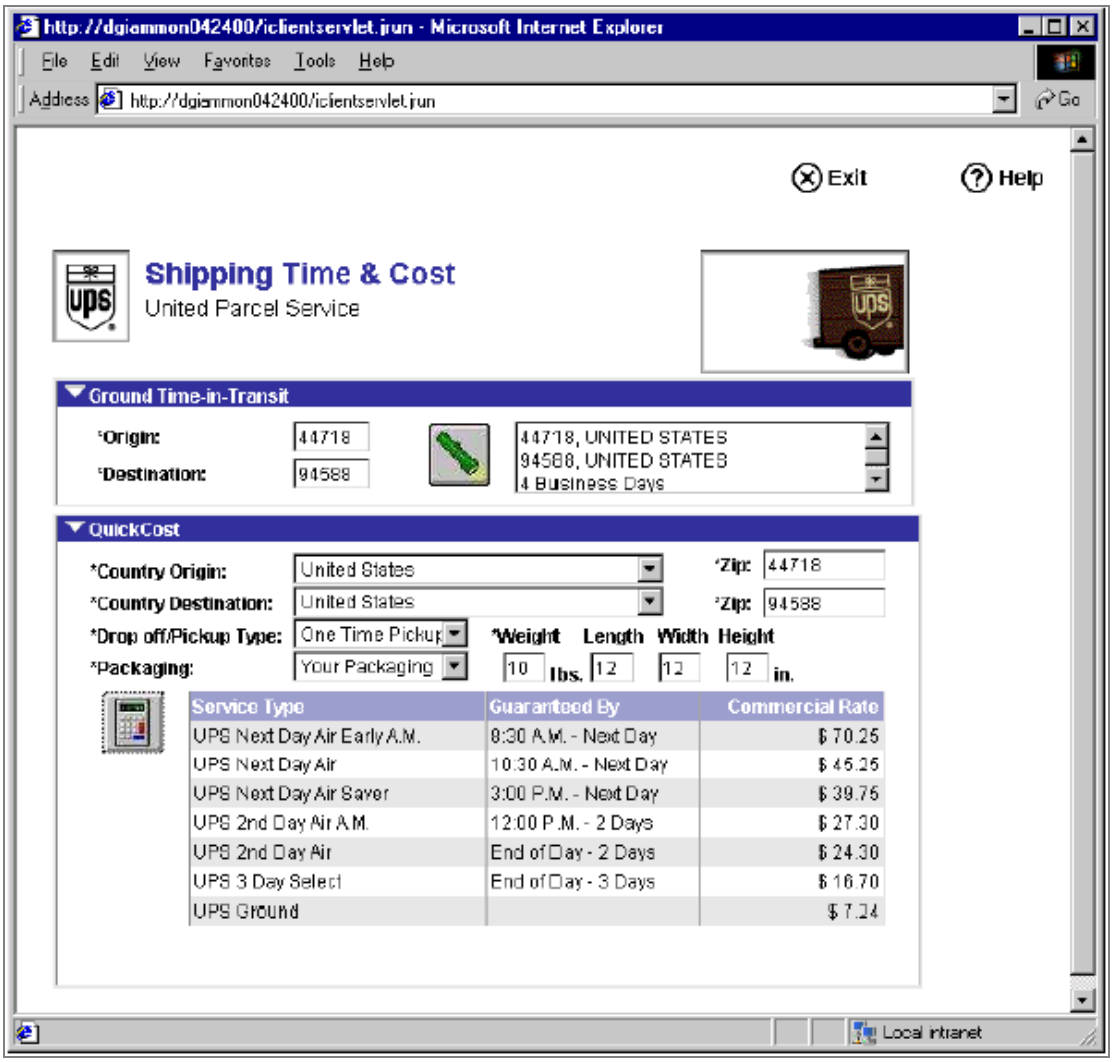


Shipping Time & Cost Page

In the Shipping and Time page, you can trigger two different Business Interlinks that call out to the UPS shipping and tracking Web site to calculate:

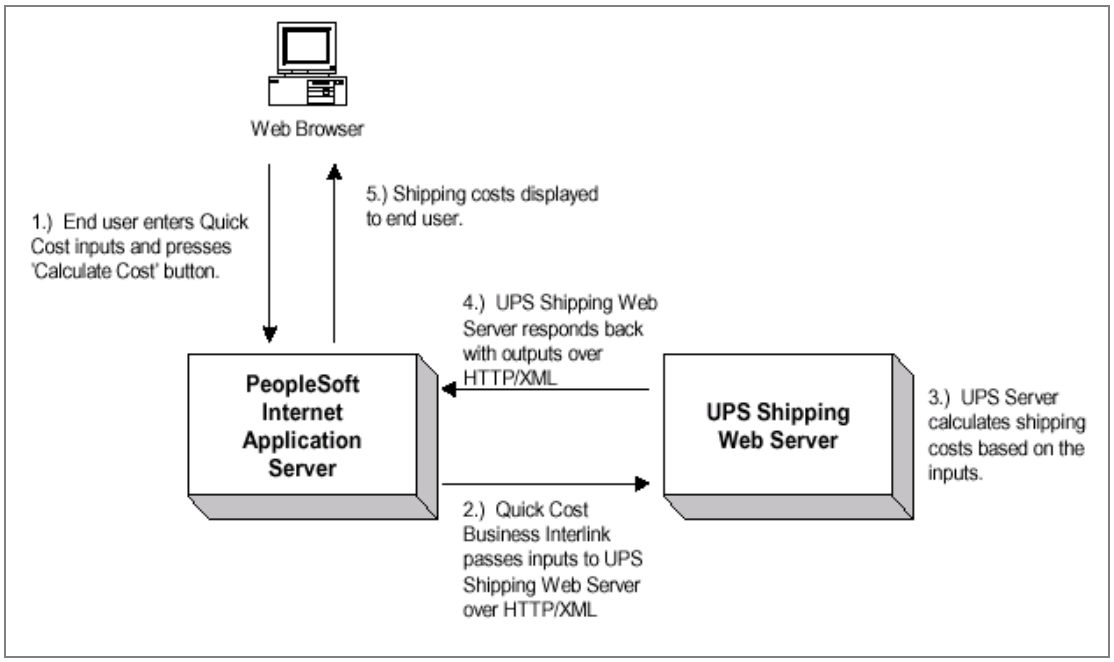
- Ground Time-in-Transit to determine how long it will take to deliver the package.
- Quick Cost to calculate how much it will cost to ship the package based on the shipping service type.

For Quick Cost, you first enter Origin Country and Postal Code, Destination Country, Postal Code, Packaging information, and then you press the Calculate button. At this point, the Business Interlink is invoked on the Application Server, and it issues an HTTP request out to the UPS Web site, which calculates the shipping costs and returns the information back to the PeopleSoft application. Here are the results of a Quick Cost calculation.



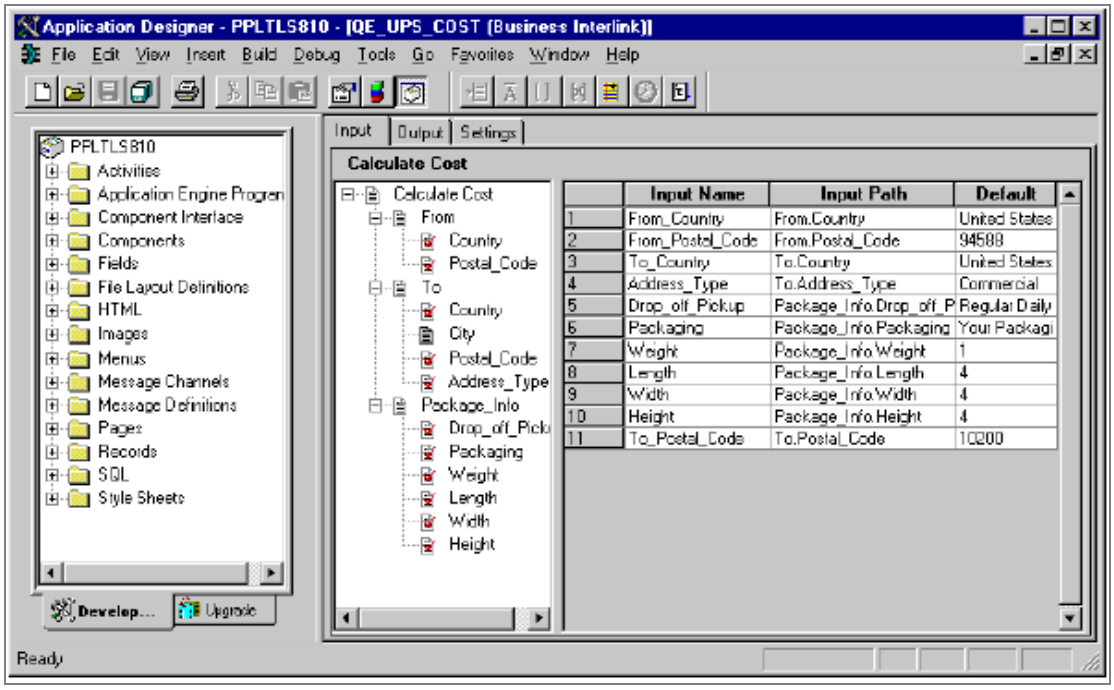
Shipping Time & Cost Page: Calculate Cost Business Interlink

This diagram explains the processing flow of this example in more detail.

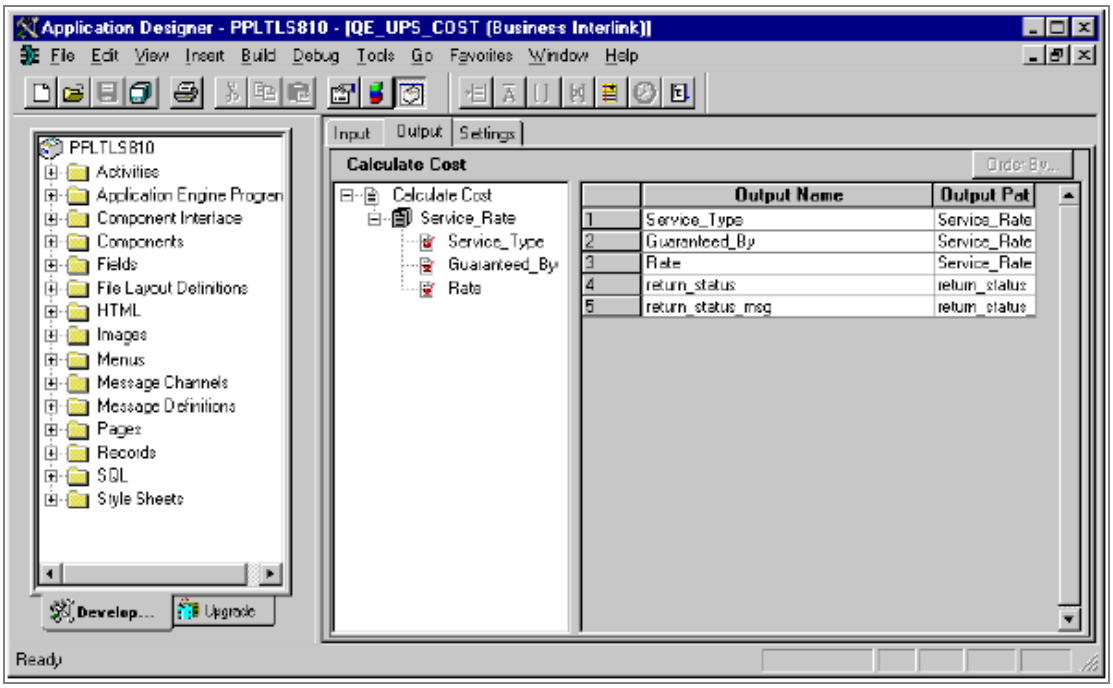


Calculate Cost Business Interlink Processing Flow

In the PeopleTools Application Designer, you can view the Business Interlink Definition for the Quick Cost application. A Business Interlink consists of a set of inputs and outputs. The first screen shows the inputs, and the second shows the outputs.



Calculate Cost Business Interlink Inputs



Calculate Cost Business Interlink Outputs

Example 2: Creating a Calculate Cost Business Interlink

This section contains code segments that overview a Business Interlink runtime plug-in that calculates shipping costs.

Runtime Plug-in Example

The runtime plug-in can be written in C++, Visual Basic, or Java. It uses the Business Interlink Object as its input and output. the Business Interlink Object containing the inputs, outputs, and Business Interlink methods. Since the runtime plug-in encalsulates the external system, PeopleSoft applications can treat all Business Interlinks the same: as a PeopleSoft object.

The runtime plug-in performs the following tasks:

- Connects to the external system.
- Passes inputs to the external system.
- Receives outputs from the external system.

A runtime plug-in must be written specifically for each external system's interface architecture.

Following is a simplified runtime plug-in pseudo-code example for a transaction named Calculate Cost. This transaction gets inputs from the Business Interlink Object IntObj, passes

them to a function provided by the third party that calculates the cost, or ServiceRateValue, and then receives the cost and passes it to the Business Interlink Object.

```
ExecuteTransaction(InterfaceObject IntObj)
{
   docsOut = IntObj->GetOutputDocs();
   docsIn = IntObj->GetInputDocs();

   if(IntObj->GetObjName() == "Calculate Cost") {
      FromValue = docsIn.GetValue("From");
      ToValue = docsIn.GetValue("To");
      PackageInfoValue = docsIn.GetValue("Package_Info");

      ExternalCallToCalcCost(FromValue, ToValue, PackageInfoValue,
         ServiceRateValue)

      docsOut.AddValue("Service_Rate", ServiceRateValue)
   }
}
```

List of Common Business Interlink Terms

Application Designer: The integrated development environment used to develop PeopleSoft applications.

Basic Type: A data type such as an integer or string.

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, Pages, etc.

XML 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. This functionality can also be written using a set of C++, Visual Basic, or other high-level language methods.

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.

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.

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 catalog: lists transactions used to interface to the external system. See Transaction.
- 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.
- Operator catalog: lists query operators used to query the external system. These query operators are used to query the classes in the class catalog.
- 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.

External System: Any system that is not directly compiled with the PeopleTools servers.

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.

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.

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.

Shape: For a transaction, the set of inputs and outputs for that transaction. For a class, the data members of that class.

CHAPTER 2

Setting Up A Business Interlink Runtime Plug-in

This section shows how to set up your development environment for writing a Business Interlink runtime plug-in. It gives examples for Visual C++ and Visual Basic.

Setting up the Development Environment in C++

This section contains directions for setting up the development environment in C++ under UNIX and under Windows NT. It also contains a template you can use for a C++ header file.

Setting up the Development Environment in C++ Under Windows NT

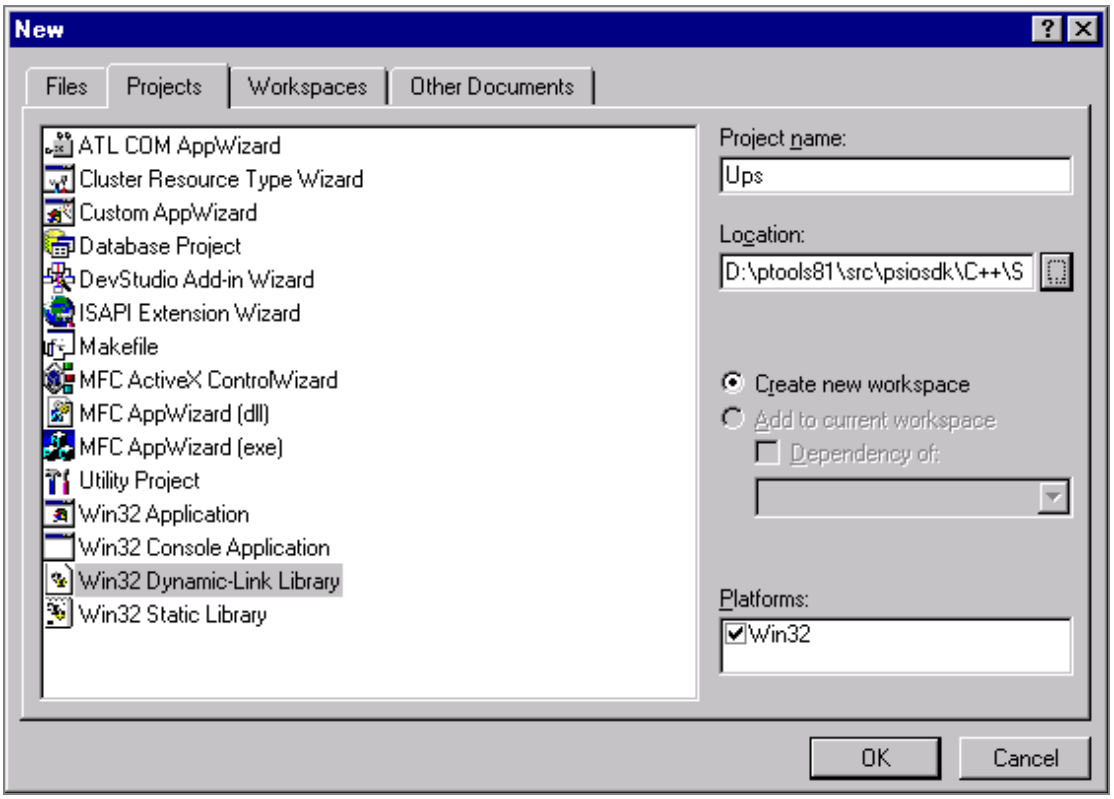
To create the C++ project that you use to write your Business Interlink Plug-in under Windows NT

1. Within Visual C++, select File:New, or press Ctrl N, and then click the Projects tab. The New dialog box, Projects tab appears.
2. Fill out the New Project dialog box, Projects tab as follows, and then click OK.

   In the list view control on the left side, select the Win32 Dynamic-Link Library type.

   In the project name type the name of the project. In this example, the name of the project is Ups.

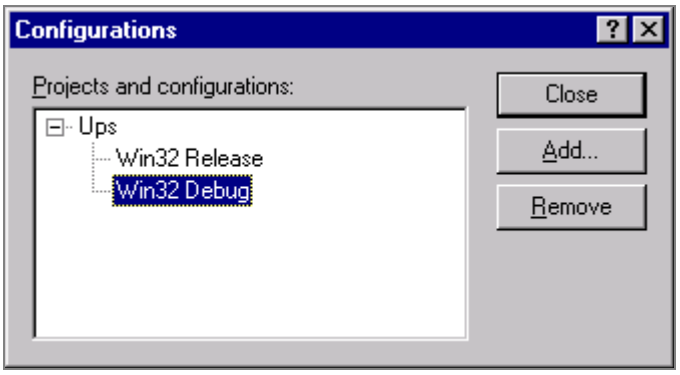
   In the location edit boxes, type the location where you want to store your project's files.



New Project Dialog Box, Projects Tab

3. Select Build:Configurations. The Configurations dialog box appears.

4. Press the Add button. The Add Project Configuration dialog box appears.



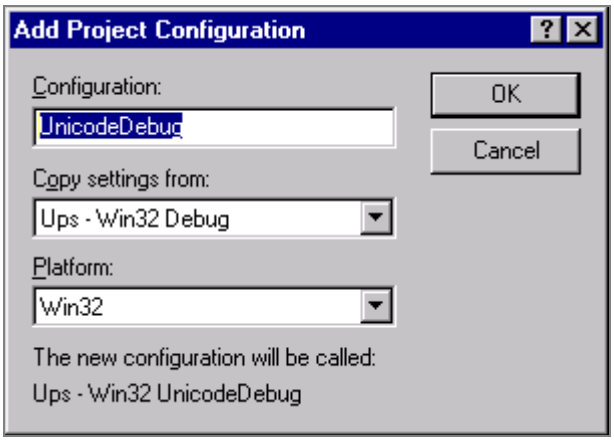
Configurations Dialog Box

The Configurations dialog box shows all of the different configurations that have been defined for your project. By default, only the Win32 Release and Win32 Debug configurations are defined.

5. Fill out the Add Project Configuration dialog box as follows and then click OK:

In the Configuration edit box, change the text from “Debug” to “UnicodeDebug”. This is set because, by default, the project is not Unicode enabled, so Unicode must be enabled here.

In the “Copy settings from” drop down box, select the Win32 Debug configuration as the template.

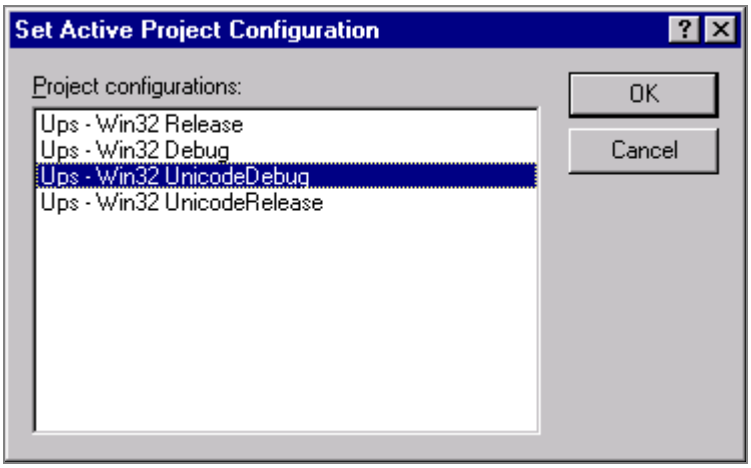


Add Project Configuration dialog box

6. Repeat steps 4 and 5, except in step 5:

   - Within the “Copy settings from” drop down box in the Add Project Configuration dialog box, select the Win32 Release configuration as the template.
   - Within the Configuration edit box in the Add Project Configuration Dialog Box, change the text to “Unicode Release”.
   - Click the Close button.

7. Select Build:Set Active Configuration. The Set Active Project Configuration dialog box appears. Select Win32 UnicodeDebug and click OK.



Set Active Project Configuration

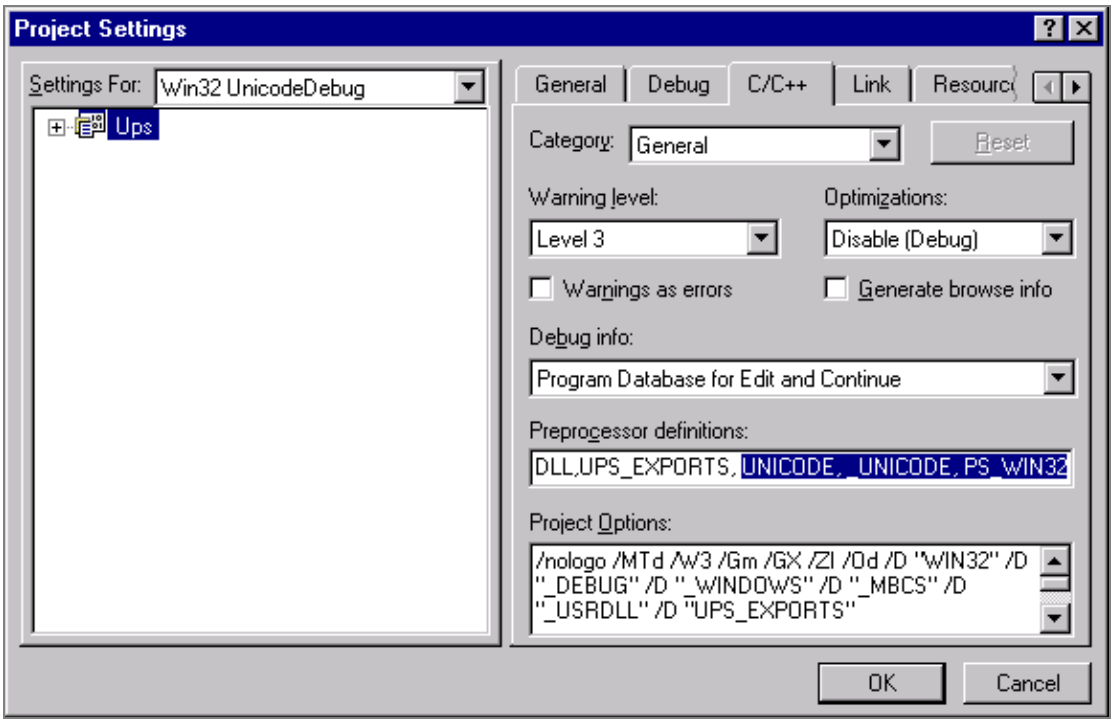
8. Select Project:Settings. The Project Setting dialog box appears.

   The Settings For drop down box contains a list of each of the configurations that you have defined for your project. This example uses the Win32 UnicodeDebug configuration; the tasks are similar for the other configurations.

9. Click the C/C++ tab. With the Category drop down box showing General, within the Preprocessor definitions edit box, check to see that the following text exists in the Preprocessor definitions list, and add it if it is not there:

   ```
   UNICODE, _UNICODE, PS_WIN32
   ```

   This will enable Unicode string support during compilation.

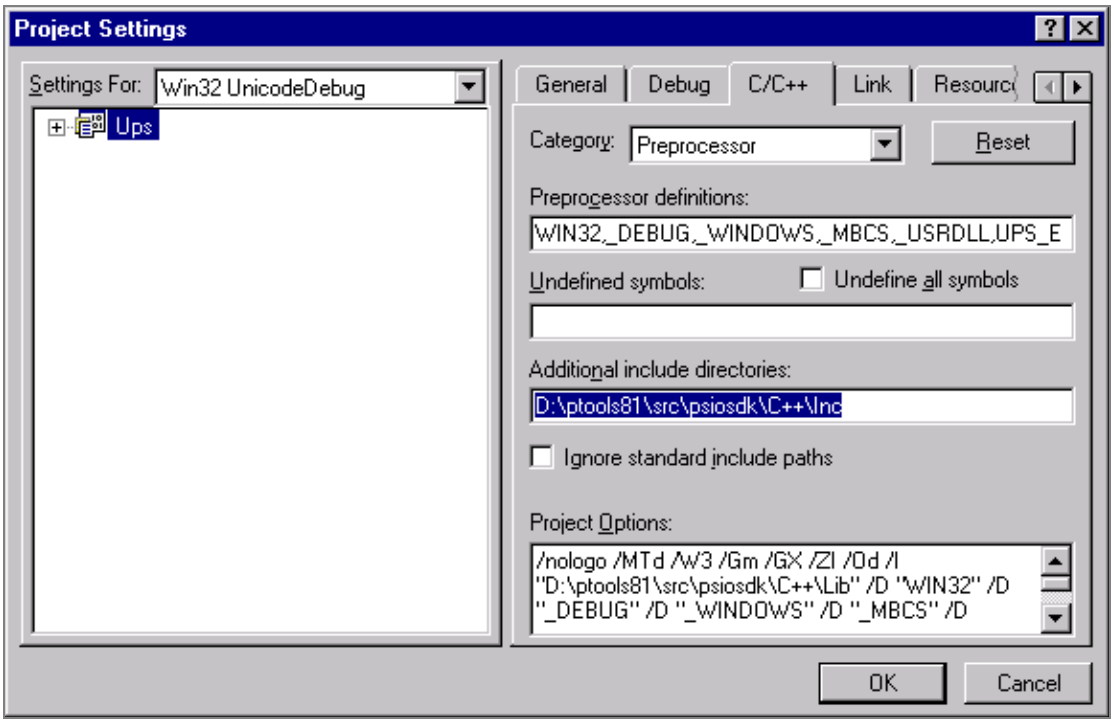


Project Settings dialog box, C/C++ tab, General category

10. From the Category drop down box, choose the Preprocessor category. In the Additional include directories edit box, type in the following path:

    ```
    <PS_HOME>\SDK\PSINTELINKS\src\C++\Inc
    ```

    This is the directory that the compiler will search for the standard PeopleSoft Business Interlink Software Development Kit header files.



Project Settings dialog box, C/C++ tab, Preprocessor category

11. Click on the Link tab. With the Category drop down box showing General, edit the Output file name edit box and the Object/library modules edit box as follows:

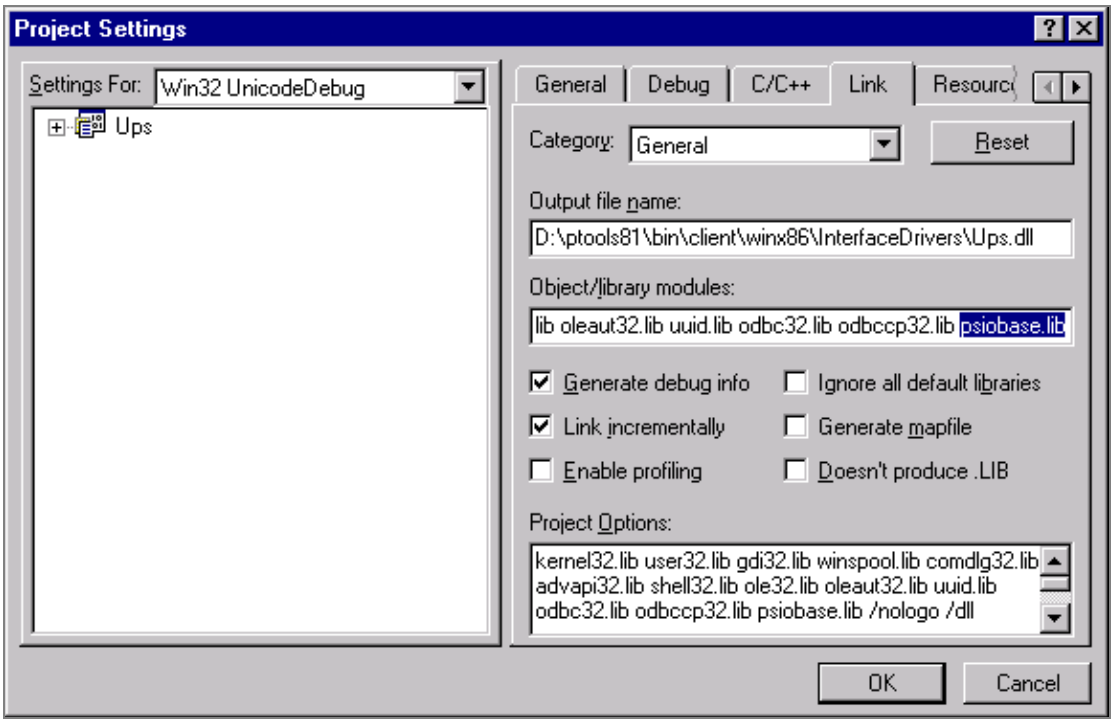
Within the Output file name edit box, enter the following path and the name of the output file for your project:

```
<PS_HOME>\bin\client\winx86\InterfaceDrivers\output.dll
```

where *output.dll* is the output file. You should use the same name for this output file as the name of the XML design-time plug-in. For example, if runtime is fcarrier.dll, the design-time should be fcarrier.xml.

For more information about the XML design-time plug-in, see the *PeopleSoft Business Interlink Design-Time Plug-in Programming Guide*, Writing an XML Design-Time Plug-In.

Within the Object/library modules edit box, add the filename psiobase.lib to the end of the list. This will link in the psiobase.lib library with your project during the build process.

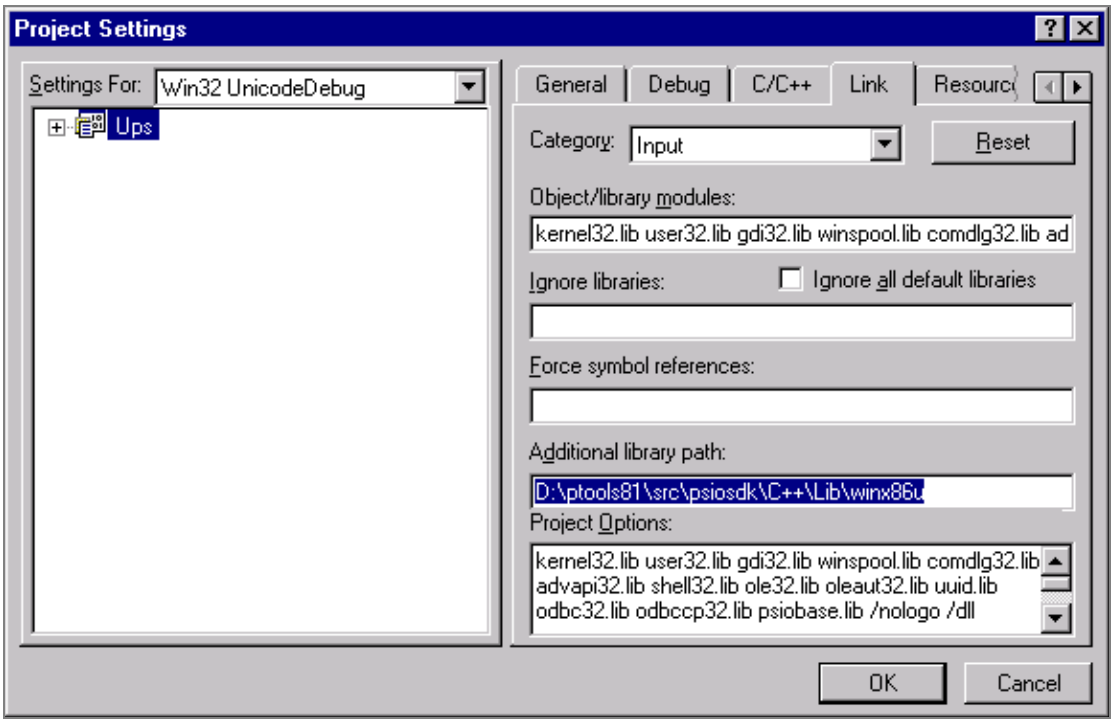


Project Settings dialog box, Link tab, General category

12. From the Category drop down box, select Input. In the Additional library path edit box, type in the following path:

```
<PS_HOME>\SDK\PSINTERLINKS\Lib\
```

This is the path for the psiobase.lib file directory.



Project Settings dialog box, Link tab, Input category

13. Press OK to accept the settings and then save your project. Your project is now configured. You can start coding.

You can save the project under the following directory, in order to keep it near the C++ Business Interlink files.

```
<PS_HOME>\SDK\PSINTERLINKS\src\C++\Samples\project_name
```

where *project_name* is the name of your project (in this case, it will likely be Ups).

14. Copy the file psiodrvr.cpp and edit that copy to create your own C++ runtime plug-in. The psiodrvr.cpp file contains most of the structure that you need for your runtime file. Copy the psiodrvr.cpp file from the following directory.

```
<PS_HOME>\SDK\src\C++\TEMPLATES
```

You can use the following code as a template for the header file for your Business Interlink class. In this code, perform the following tasks:

- Include the InterfaceObject class.
- Include the input/output table classes.
- Declare your class so it inherits the DLLBaseDriver class. In this case, the name of your class is *CUps*; you will use whatever name fits your project.
- Create a constructor for your class.

- Declare the pure virtual methods GetVersion, IsVersionCompatible, and the execution method you will use:: ExecuteTransaction, ExecuteQuery, ExecuteAdd, ExecuteUpdate, and/or ExecuteDelete. In most cases, you will use ExecuteTransaction.

```
// The include files.
#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#include "iodrvr.h"   // Contains the InterfaceObject class
#include "ioutil.h"   // Contains the input/output table classes
// Also include any classes you need to use your external system.

// Declare your class, and inherit the DLLBaseDriver class.
class CUps : public DLLBaseDriver
{
public:
   // Constructor for your class.
   CUps() : IntObj(0) {}

   // Declare the virtual methods
   virtual const TCHAR* GetVersion() const;
   virtual BOOL  IsVersionCompatible(const TCHAR* version);
   virtual EIOCEXECSTATUS
     ExecuteTransaction(InterfaceObject * IntObj,
     const int nBatchMode);

// Create the InterfaceObject pointer.
public:
   InterfaceObject* IntObj;
};
```

Setting up the Development Environment in C++ Under UNIX

In order to write a Business Interlink runtime plug-in using C++, you set up the environment for Buisiness Interlinks on the UNIX system.

To create the C++ project that you use to write your Business Interlink Plug-in under UNIX

1. Change directories to the <PS_HOME> directory and run the following command:

```
. ./psconfig.sh
```

2. If you wish, you can test the setup by running the Business Interlink tester on the files in the simple directory. If the tester works, the setup installed properly.

For more information on running the Business Interlink Tester under UNIX, see Using the Business Interlink Tester: UNIX.

3. Create the directory in which you will create your plug-in. (A recommended location is <PS_HOME>/sdk/psinterlinks/src/C++/samples.) Within that directory, create a directory named unix.

```
mkdir yourplugindir
cd yourplugindir
mkdir unix
```

where *yourplugindir* is the name of the directory in which you will create your plug-in.

4. Copy the files named makefile and makeunix.mak from the newplugin/unix directory to *yourplugindir*/unix, where *yourplugindir* is the name of the directory that you created in the previous step.

```
cd ..
cp newplugin/unix/* yourplugindir/unix/*
```

5. Edit the file named makefile in the *yourplugindir*/UNIX directory.

Change the following line:

```
name=          newplugin
```

replacing newplugin.

```
name=          yourplugindir
```

where *yourplugindir* is the name of the directory you created earlier in which you will create your plug-in.

Change the following lines, replacing newfile1 with the name of any .cpp file that you will have within the *yourplugindir* directory. You must copy this code and make a similar change for every .cpp file that you have within the *yourplugindir* directory.

```
newfile1.o : $(srcdir)/newfile1.cpp
     $(CC) $(dbgflags) $(cmpopts) $(typedefine) $(defines) $(unicodedefs)
$(libincludes) $(includes) $(tuxincludes) $(srcdir)/newfile1.cpp -E >
newfile1.x
     wconv $(unicodedefs) newfile1.x newfile1.i
     $(CC) $(cmpopts) $(dbgflags) $(typedefine) $(defines) $(unicodedefs)
$(libincludes) $(includes) $(tuxincludes) newfile1.i -c
     rm newfile1.x
```

After replacing newfile1 with yourplugin:

```
yourplugin.o : $(srcdir)/yourplugin.cpp
     $(CC) $(dbgflags) $(cmpopts) $(typedefine) $(defines) $(unicodedefs)
$(libincludes) $(includes) $(tuxincludes) $(srcdir)/yourplugin.cpp -E >
yourplugin.x
```

```
    wconv $(unicodedefs) yourplugin.x yourplugin.i
    $(CC) $(cmpopts) $(dbgflags) $(typedefine) $(defines) $(unicodedefs)
$(libincludes) $(includes) $(tuxincludes) yourplugin.i -c
    rm yourplugin.x
```

6. Copy the file named psiodrvr.cpp into the *yourplugindir* directory. Change the following lines to match the names of your plug-in (*yourplugin*):

```
#include                "windows.h"
#include                "iodrvr.h"
#include                "yourplugin.h"
#define EXPORT_PSTYP __declspec(dllexport)

static yourplugin baseDriver;
```

7. Create your Business Interlink Runtime plug-in. Store the .cpp and .h files within the *yourplugindir* directory. You can copy, rename, and use the .cpp and .h files contained in the simple directory as a template.

8. To compile and link your runtime plug-in, run the make command:

```
make yourplugin
```

where *yourplugin* is the name of your runtime plug-in.

Your runtime plug-in should use the same name as the name of the XML design-time plug-in. For example, if runtime is fcarrier.dll, the design-time should be fcarrier.xml.

For more information about the XML design-time plug-in, see the *PeopleSoft Business Interlink Design-Time Plug-in Programming Guide*, Writing an XML Design-Time Plug-In.

Creating a C++ Template

You can use the following code as a template for the header of your Business Interlink class. In this code, perform the following tasks:

- Include the InterfaceObject class.
- Include the input/output table classes.
- Declare your class so it inherits the DLLBaseDriver class. In this case, the name of your class is *CUps*; you will use whatever name fits your project.
- Create a constructor for your class.
- Declare the pure virtual methods GetVersion, IsVersionCompatible, and the execution method you will use:: ExecuteTransaction, ExecuteQuery, ExecuteAdd, ExecuteUpdate, and/or ExecuteDelete. In most cases, you will use ExecuteTransaction.

```
// The include files.
#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#include "iodrvr.h"   // Contains the InterfaceObject class
#include "ioutil.h"   // Contains the input/output table classes
// Also include any classes you need to use your external system.

// Declare your class, and inherit the DLLBaseDriver class.
class yourplugin : public DLLBaseDriver
{
public:
   // Constructor for your class.
   yourplugin() : IntObj(0) {}

   // Declare the virtual methods
   virtual const TCHAR* GetVersion() const;
   virtual BOOL  IsVersionCompatible(const TCHAR* version);
   virtual EIOCEXECSTATUS
     ExecuteTransaction(InterfaceObject * IntObj,
     const int nBatchMode);

// Create the InterfaceObject pointer.
public:
   InterfaceObject* IntObj;
};
```

Setting up the Development Environment in Visual Basic

In order to write your own plug-in in Microsoft Visual Basic, you must have access to the following files:

- IoCollection.DLL
- PsIntObj.DLL
- PsIoDriver.tlb

These files are included in your Business Interlink SDK installation.

You must register the IoCollection.DLL and PsIntObj.DLL files in your development workstation to use the Visual Basic/COM capabilities.

To register IoCollection.DLL and PsIntObj.DLL:

1. Go to the command line. Select Start:Programs:Command Prompt.
2. Change your directory to the directory where the above files are stored. For example, if you have installed PeopleTools on your D: drive in a directory names ptool81, you might use the following command to change to the directory containing IoCollection.DLL and PsIntObj.DLL:.

```
cd <PS_HOME>\bin\client\winx86
```

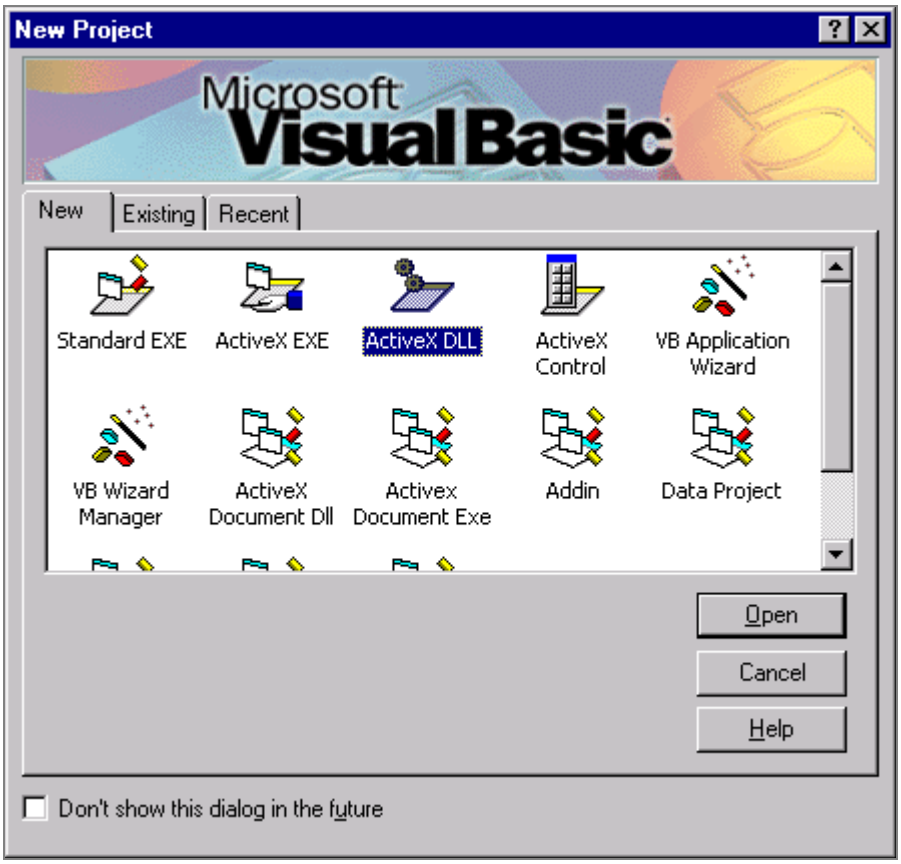
3. Execute the \winnt\system32\regsvr32.exe with each DLL (the following command unregisters before each register command as a precaution):

```
C:\winnt\system32\regsvr32.exe u IoCollection.DLL
C:\winnt\system32\regsvr32.exe IoCollection.DLL

C:\winnt\system32\regsvr32.exe u PsIntObj.DLL
C:\winnt\system32\regsvr32.exe PsIntObj.DLL
```

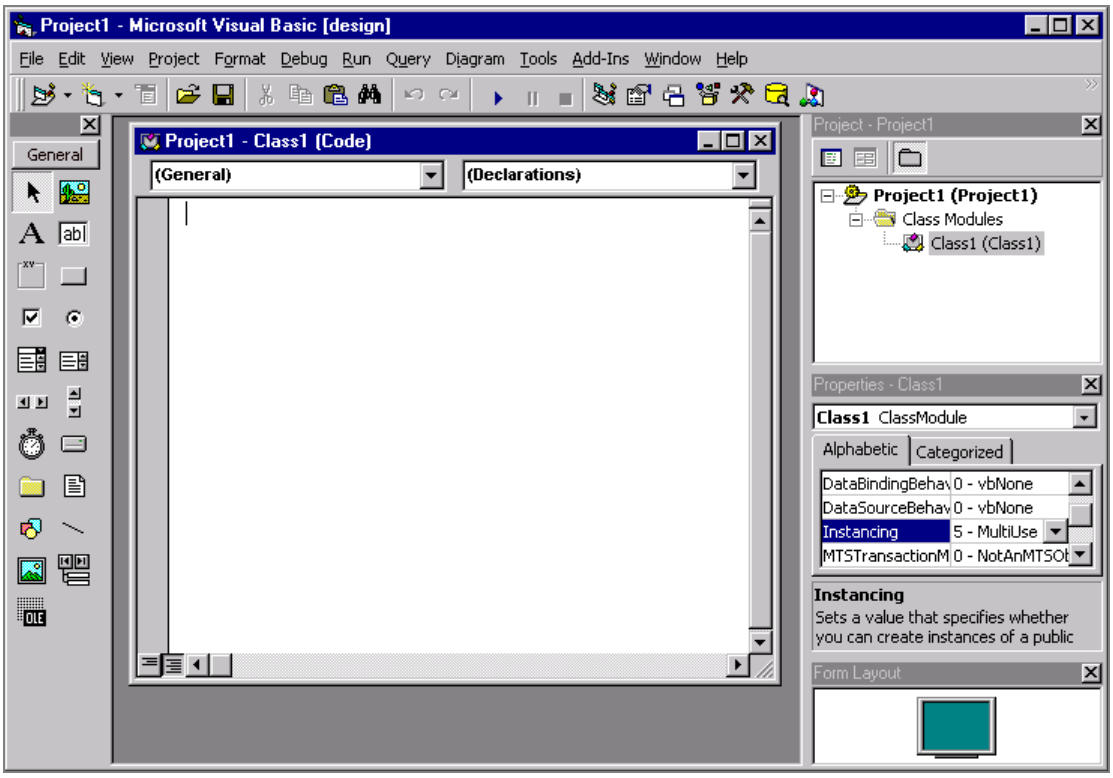
Setting Up Visual Basic Application Development

To create the Visual Basic project that you use to write your Business Interlink Plug-in

1. Launch Microsoft Visual Basic. Microsoft Visual Basic 6.0 is needed.
2. Select ActiveX.DLL from the new Project window. A New Project appears.

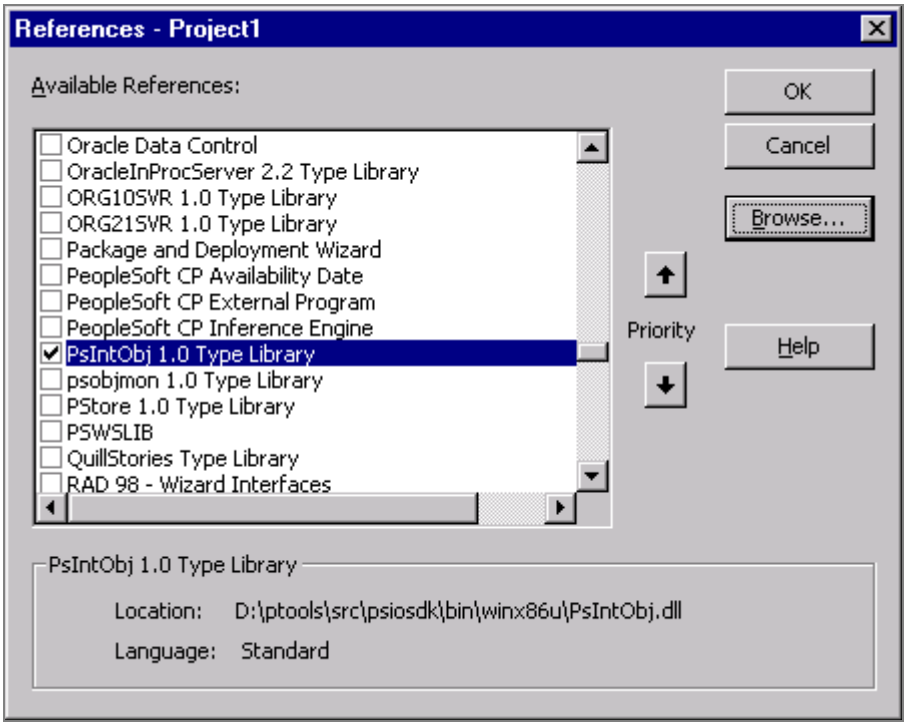


Selecting ActiveX.DLL



A New Project Window

3. From the main menu, select Project:References. The References window appears.



Available References Window

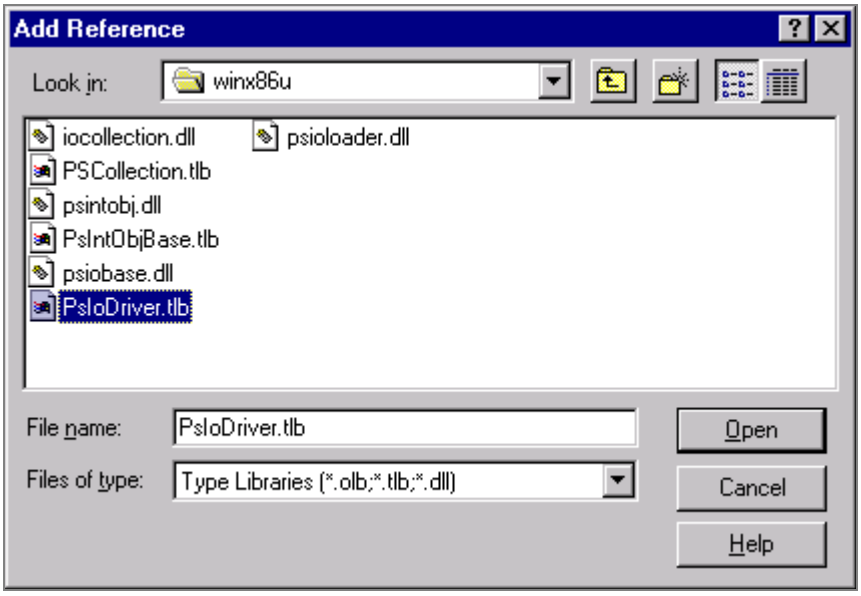
4. From References option menu check out the following DLLs and TLB:

   PsIoDriver.tlb. Press the Browse button to open the Add References window, navigate to the directory containing PsIoDriver.tlb, and select PsIoDriver.tlb. The directory is located at:

   ```
   <PS_HOME>\bin\client\winx86
   ```

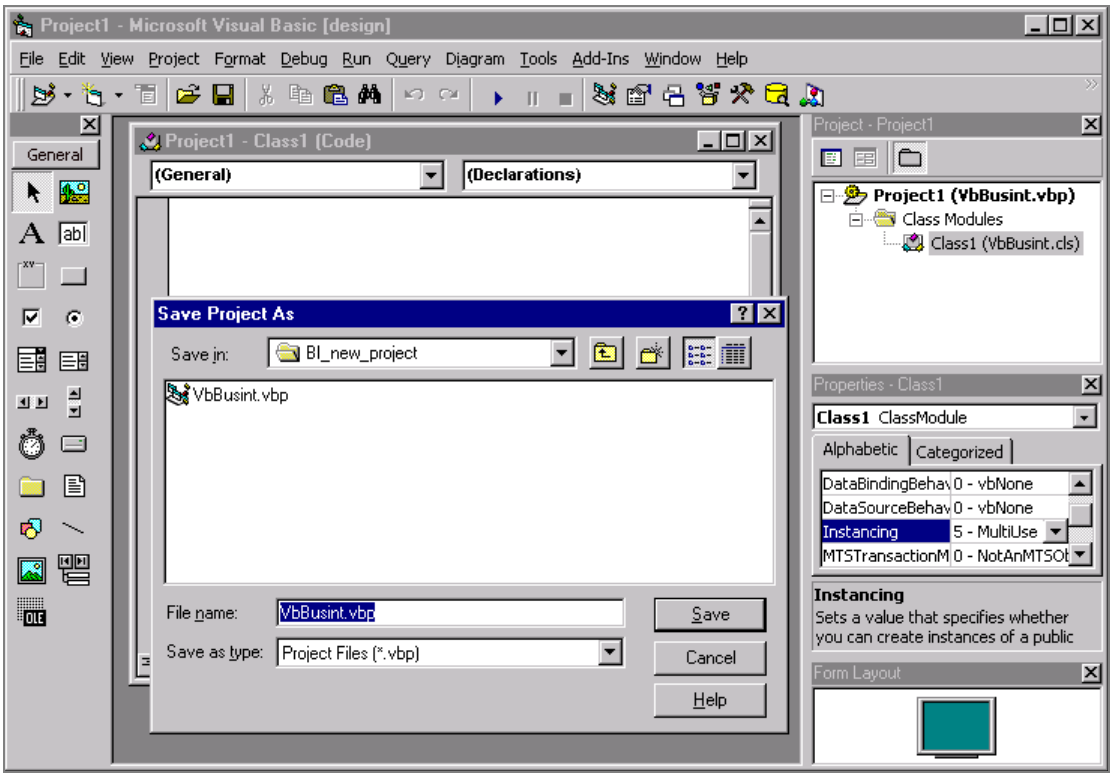
   IoCollection.DLL and PsIntObj.DLL. You can either use the References window and check PsIntObj 1.0 Type Library and IoCollection 1.0 Type Library, or you can use the Add References window to navigate to the directory containing IoCollection.DLL and PsIntObj.DLL, and select them. The directory is located at:

   ```
   <PS_HOME>\bin\client\winx86
   ```



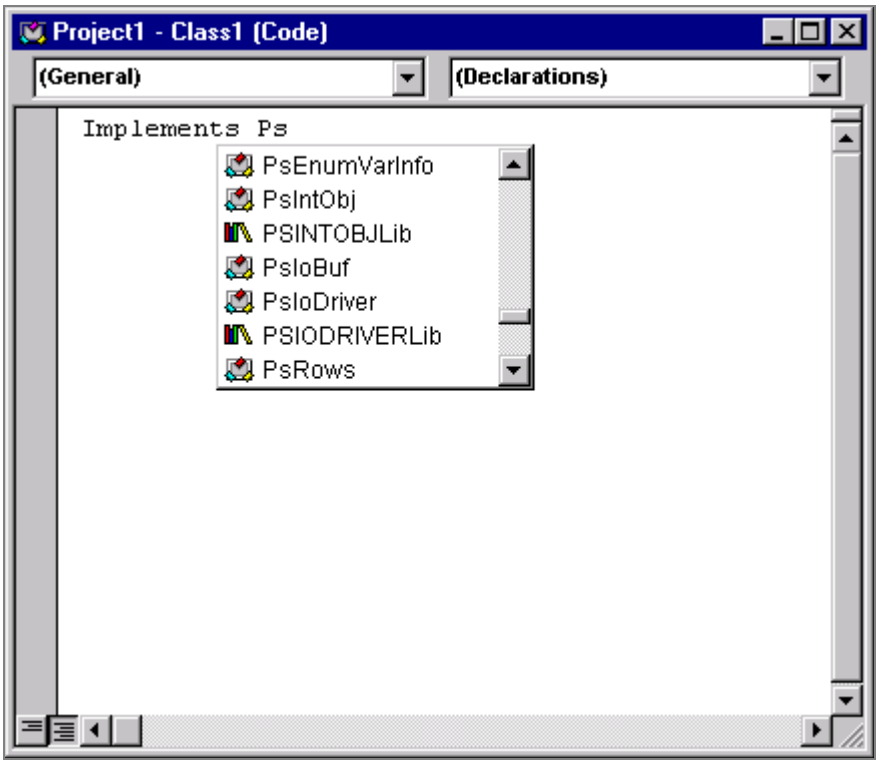
Add References Window

5. From main menu select Save Project. Navigate to the directory where you want to save your project, and save it.



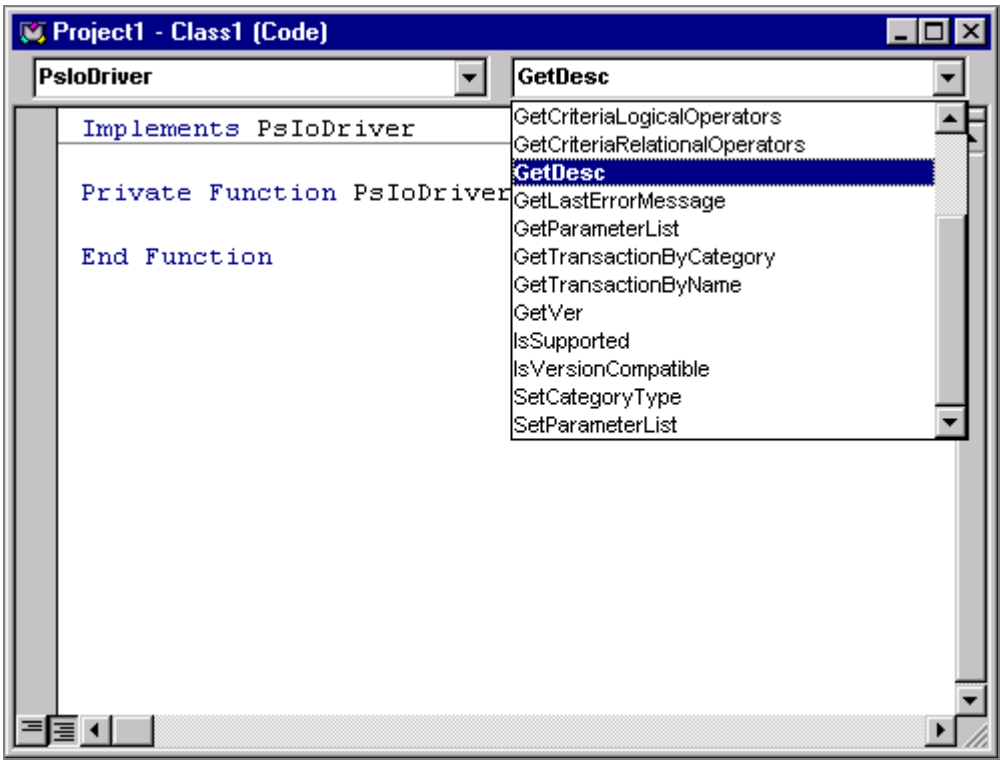
Saving The Project

6. Type in “Implements PsIoDriver” in your Class (code) screen, then select PsIoDriver from the Class category pop-up list box. .



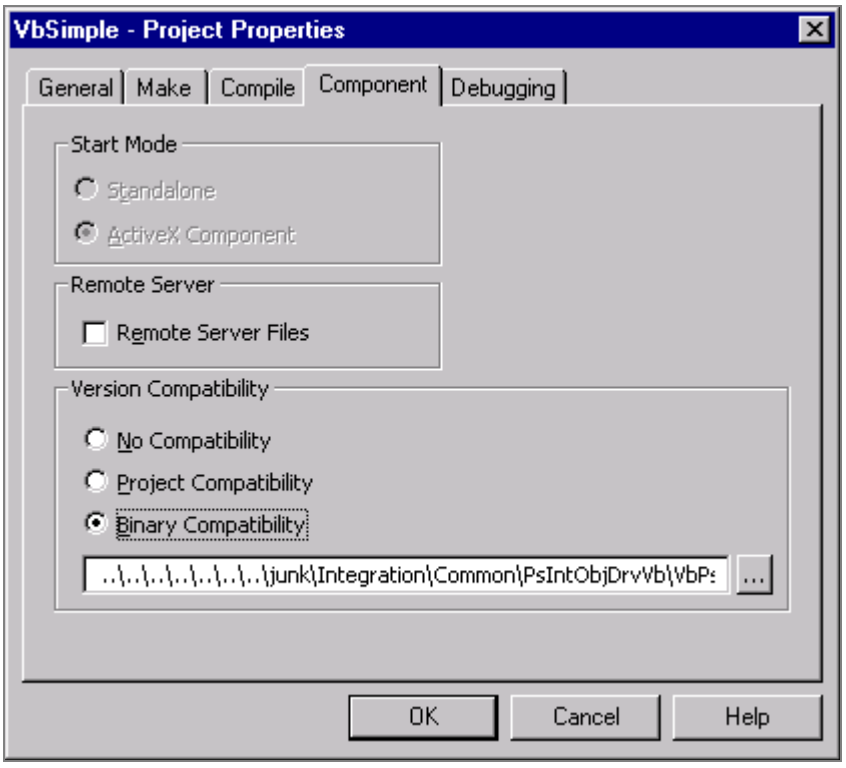
Implements Window

7. Select all the functions from Function category List Box (Select one function at a time). As you select each function, the function will be enabled in the Function category list box.



Select Functions Listbox

8. Set the level of version compatibility of your VB project to Binary. Open your Visual Basic project, and from the Project menu, select *yourproject* Properties, where *yourproject* is the name of your project. In the picture below, the project is names VbSimple. Then click the Binary Compatability radio button and click OK.



Select Binary Compatability

9. Implement the functions provided by PsIoDriver that you will need for your project.

For more information on the functions that you implement, refer to Writing the Version Methods for a Business Interlink Runtime Plug-In and Writing the Execution Method for a Runtime Plug-In.

If there is no Business Interlink XML design-time plug-in for your project, you will need to implement some or all of the dynamic methods.

For more information on writing methods instead of an XML design-time plug-in, see Writing the Design-Time Functionality Using Dynamic Catalog Methods.

Your runtime plug-in should use the same name as the name of the XML design-time plug-in. For example, if runtime is fcarrier.dll, the design-time should be fcarrier.xml.

For more information about the XML design-time plug-in, see the *PeopleSoft Business Interlink Design-Time Plug-in Programming Guide*, Writing an XML Design-Time Plug-In.

Setting up the Development Environment in Java

This section contains directions for setting up the development environment in Java under UNIX and under Windows NT.

Setting up the Development Environment in Java Under Windows NT

Note. For Windows NT, the Java Plug-In must be developed on, and reside on, the same system where the application server is installed.

To create the Java project that you use to write your Business Interlink Plug-in under Windows NT

1. If you have not done so, install Java Development Kit version 1.2.2 and Java Runtime Environment 1.2.2.

   Java Development Kit version 1.2.2 is located at:

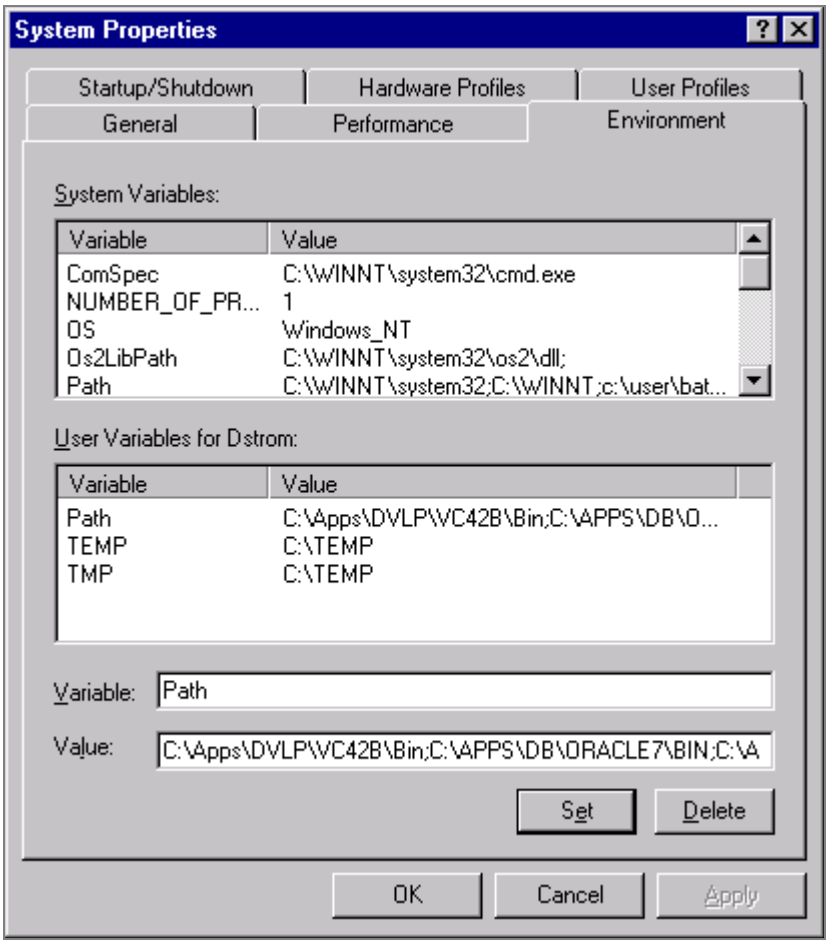
   ```
   http://java.sun.com/products/jdk/1.2/
   ```

   Java Runtime Environment is located at:

   ```
   http://java.sun.com/products/jdk/1.2/jre/index.html
   ```

2. Append the psinterlinks.jar file to your CLASSPATH variable (if you do not have one, create it). In Windows NT, right-click on the icon for your computer, select Properties, and in the System Properties panel, Environment tab, append the following text in the Value edit box:

   ```
   C:\<PS_HOME>\class\psinterlinks.jar
   ```



System Properties panel, Environment Tab

3. If you are using a 2-tier system (you are not using an application server), then you must set the following lines in the PATH variable:

```
C:\Program Files\Javasoft\Jre\1.2\bin

C:\Program Files\Javasoft\Jre\1.2\bin\Classic
```

   If you are using a 3-tier system (you are using an application server), you must enter the following line into the PATH variable in the psappsrv.env file (located in the directory \appserv\<name of application server domain>):

```
C:\Program Files\JavaSoft\Jre\1.2\bin\Classic
```

4. Copy the file named psjavaproxy.dll into the InterfaceDrivers directory, which is where you will place your Java runtime plug-in. Then rename the copied file to match the name of your Java runtime plug-in. psjavaproxy.dll is located in the InterfaceDrivers directory:

```
<PS_HOME>\bin\client\winx86\InterfaceDrivers\psjavaproxy.dll
```

5. You can use a java file provided in the following directory to use as a template for your Java runtime plug-in.

```
<PS_HOME>\sdk\PSINTERLINKS\Src\Java\Samples
```

6. You must append the location of your Java runtime plug-in to your CLASSPATH variable.

   If the runtime plug-in is a .class file, you need only add the directory location:

```
<PS_HOME>\class\
```

   If the runtime plug-in is a .jar files, you must include the .jar file name as well.

```
<PS_HOME>\class\yourjavaplugin.jar
```

In Java, you must enclose all of the Java Business Interlink methods for your runtime plug-in within a public class that implements PSBusInterlink. For example, the following code would be the public class for a Freight Carrier class names Fcarrier.

```
Public class Fcarrier extends Object implements PSBusinterlink {

   // Include all the methods you use here: GetVersion, IsVersionCompatible,
   // ExecuteTransaction.
}
```

Your runtime plug-in should use the same name as the name of the XML design-time plug-in. For example, if runtime is fcarrier.dll, the design-time should be fcarrier.xml.

For more information about the XML design-time plug-in, see the *PeopleSoft Business Interlink Design-Time Plug-in Programming Guide*, Writing an XML Design-Time Plug-In.

Setting up the Development Environment in Java Under UNIX

To create the Java project that you use to write your Business Interlink Plug-in under UNIX

1. If you have not done so, install Java Development Kit version 1.2.2 and Java Runtime Environment 1.2.2.

   Java Development Kit version 1.2.2 is located at:

```
http://java.sun.com/products/jdk/1.2/
```

   Java Runtime Environment is located at:

```
http://java.sun.com/products/jdk/1.2/jre/index.html
```

2. Copy the libpsjavaproxy file into the directory where you are developing your Java plug-in. Name the copied libpsjavaproxy file to be the same as that of your Java runtime plug-in. The libpsjavaproxy file is located at:

```
<PS_HOME>\bin\client\interfacedrivers\libpsjavaproxy
```

Within that directory, you could use the following command to do the copy and naming, assuming that your plug-in is a Freight Carrier plug-in, and you want to name your runtime plug-in Fcarrier.

```
cp libpsjavaproxy.so libFcarrier.so
```

The libpsjavaproxy file is named:

- on Solaris, libpsjavaproxy.so
- on AIX, libpsjavaproxy.a
- on HP UNIX, libpsjavaproxy.sl

3. Write your Java runtime plug-in. The <PS_Home>/appserv/classes directory is the recommended location where you place your Java runtime plug-in. If you are writing a Freight Carrier plug-in, then your plug-in could be named Fcarrier.class.

   You can use java files provided in the following directory to use as a template for your Java runtime plug-in.

```
<PS_HOME>\sdk\psinterlinks\Src\Java\Samples
```

For more information on writing the runtime plug-in, see Writing the Version Methods for a Business Interlink Runtime Plug-In and Writing the Execution Method for a Runtime Plug-In.

In Java, you must enclose all of the Java Business Interlink methods for your runtime plug-in within a public class that implements PSBusInterlink. For example, the following code would be the public class for a Freight Carrier class names Fcarrier.

```
Public class Fcarrier extends Object implements PSBusinterlink {

   // Include all the methods you use here: GetVersion, IsVersionCompatible,
   // ExecuteTransaction.
}
```

Your runtime plug-in should use the same name as the name of the XML design-time plug-in. For example, if runtime is fcarrier.dll, the design-time should be fcarrier.xml.

For more information about the XML design-time plug-in, see the *PeopleSoft Business Interlink Design-Time Plug-in Programming Guide*, Writing an XML Design-Time Plug-In.

Understanding the Business Interlink SDK Directory Structure

This section describes the Business Interlink directory structures for Windows NY and UNIX.

Understanding the Business Interlink SDK Directory Structure: UNIX

You might need to untar the SDK directory structure. To do so, run the following commands:

```
cd $PS_HOME
./ untarsdk.sh
```

The Business Interlink directory structure on your UNIX installation consists of several directories:

- $PS_HOME, which is the location of the PeopleTools installation. This directory contains a file named psconfig.sh, and has a path leading to all the other directories listed here: $PS_HOME/sdk/*UNIXinstall*/, where *UNIXinstall* is the directory containing the UNIX sdk for your system.
- bin, which contains the file bitest, which is the executable for the UNIX Business Interlink tester.
- src/C++, which contains the C++ directories inc and samples.
  - inc contains the include files used by Business Interlink runtime plug-ins written in C++.
  - samples contains the directories newplugin and simple. These are samples of C++ runtime plug-ins.
  - simple contains the files psiodrvr.cpp, psiodrvr_simple.cpp, and psiodrvr_simple.h. psiodrvr_simple.cpp is the runtime plug-in for the simple Business Interlink. It also contains a directory named unix that contains the files makefile and makeunix.mak.
  - newplugin, which contains a directory named unix that contains the files makefile and makeunix.mak.
- src/java, which contains the samples directory.
  - samples contains the directory simple, which contains a sample of a Java runtime plug-in.

```
$PS_HOME
   psconfig.sh
   bin
      bitest
      client\winx86\interfacedrivers  (for Windows NT)
      client\interfacedrivers         (for UNIX)
         libpsjavaproxy
   sdk
      src
         C++
            inc
            samples
               newplugin
```

```
            unix
               makefile
               makeunix.mak
         simple
            psiodrvr.cpp
            psiodrvr_simple.h
            psiodrvr_simple.cpp
            unix
               makefile
               makeunix.mak
Java
   samples
      simple
         simple.class
```

Understanding the Business Interlink SDK Directory Structure: Windows NT

The Business Interlink directory structure on your Windows NT installation consists of several directories:

- <PS_HOME>, which is the location of the PeopleTools installation. This directory contains all the other directories listed here.
- bin\client\winx86\InterfaceDrivers, which contains the XML design-time plug-ins and the runtime plug-ins for Business Interlinks.
- Sdk\PSINTERLINKS\Src\C++, which contains:
  - inc contains the include files used by Business Interlink runtime plug-ins written in C++.
  - samples contains the samples of C++ runtime plug-ins.
  - TEMPLATES contains the template for a C++ plug-in, psiodrvr.cpp; the template for an XML design-time plug-in, template; and the template for an XML design-time plug-in using the pshttpenable runtime plug-in.
- Sdk\PSINTERLINKS\Src\Com\Samples, which contains the samples of Visual Basic runtime plug-ins.
- Sdk\PSINTERLINKS\Src\Java\Samples, which contains the samples of Java runtime plug-ins.

CHAPTER 3

Writing the Version Methods for a Business Interlink Runtime Plug-In

The version information consists of writing the following methods:

- GetVersion or GetVer. This method provides the version number for your Business Interlink Plug-in.
- IsVersionCompatible. This method tells whether or not your Business Interlink Plug-in is compatible with previous versions of your Business Interlink Plug-in.
- InstantiateDriverInstance. This method creates an instance of your Business Interlink Plug-in class. (C++ only)

Writing GetVersion for Your Business Interlink Plug-in Class

Write the GetVersion method to supply your Business Interlink Plug-in with a version number. This allows you to write future versions of your Business Interlink Plug-in, and to uniquely identify each version with a number.

For the Freight Carrier Business Interlink Plug-in, GetVersion is coded as follows:

```
const TCHAR * CUps::GetVersion() const
{
return _T("8.00");
}
```

For Visual Basic, the method PsIoDriver_GetVer returns the version number.

```
Private Function PsIoDriver_GetVer() As String
    '
    PsIoDriver_GetVer = "1.0.0"
    '
End Function
```

Writing IsVersionCompatible for Your Business Interlink Plug-in Class

You write the IsVersionCompatible pure virtual method to have your Business Interlink Plug-in tell if it is compatible with previous versions of your XML design-time plug-in. It should

compare the version number in GetVersion to the version number in the XML design-time plug-in, and it should determine if those versions are compatible.

The input parameter, version, is a version number that will be supplied by the Business Interlink Framework. The framework will call the IsVersionCompatible method and test it against every version that exists for your Business Interlink Plug-in, testing it for compatibility with previous versions.

For the Freight Carrier Business Interlink Plug-in, you can code IsVersionCompatible as follows:

```
BOOL  IsVersionCompatible(const TCHAR* version) { return TRUE; }


// There is only one version of the UPS Business Interlink Plug-in;
// return TRUE because it is compatible with itself.
```

Suppose your XML design-time plug-in has more than one version, and your run-time plug-in will be compatible with versions 8.0 and 8.1. Then you could write IsVestionCompatible as follows:

```
BOOL  IsVersionCompatible(const TCHAR* version)
{
   if (version == _T("8.00"))
      return TRUE;
   else if (version == _T("8.10"))
      return TRUE;
   else
      return FALSE;
}
```

For Visual Basic, the method is PsIoDriver_IsVersionCompatible.

```
Private Function PsIoDriver_IsVersionCompatible(ByVal bstrVerion As String) As
Long
    '
    PsIoDriver_IsVersionCompatible = True
    '
End Function
```

Writing InstantiateDriverInstance for Your Business Interlink Plug-in (C++)

With C++, you need to write the method InstantiateDriverInstance. This method is called when an instance of your class needs to be instantiated.

You do not need to create a corresponding method to destroy instances of your class; destruction of the instances of your class is automatically done for you.

For the Freight Carrier Business Interlink Plug-in, InstantiateDriverInstance is coded as follows:

```
extern DLLBaseDriver* InstantiateDriverInstance();
```

CHAPTER 4

Writing the Execution Method for a Runtime Plug-In

The execution method performs the bulk of the work for a runtime plugin. Taking the Interlink Object as input, it performs the following tasks:

- Extracts the inputs form the Interlink Object.
- Calls the external system to perform a certain function, passes the inputs to the external system to use as its inputs.
- Receives outputs from the external system.
- Adds the outputs to the Interlink Object.

This section discusses how to write the ExecuteTransaction execution method. This is the method to write when you are creating Business Interlink Transactions, which is the most common type of Business Interlink. It uses inputs and outputs, but not criteria.

The ExecutionObjectAdd method can be coded in a similar way to ExecuteTransaction. It extracts inputs from the Business Interlink Object, but does not add outputs to it.

For more information about writing ExecuteObjectQuery, ExecuteObjectUpdate, or ExecuteObjectDelete, which use the Criteria methods, see Writing The Execution Method for a Runtime Plug-In (Criteria Data).

For more information about the methods that are mentioned here, or any other Business Interlink methods that you can use when writing a runtime plug-in, refer to Understanding The Business Interlink Methods.

This chapter explains the tasks to perform to write the ExecuteTransaction of a Freight Carrier runtime plug-in. The tasks, as explained here, show the code needed for the Calculate Cost transaction.

For a listing of the ExecuteTransaction written in C++, see ExecuteTransaction in C++.

For a listing of the ExecuteTransaction written in Visual Basic, see ExecuteTransaction in Visual Basic.

Take A Business Interlink Object As Input

The execution methods that you write all receive a Business Interlink Object as an input.

In C++:

```
EIOCEXECSTATUS CUps::ExecuteTransaction(InterfaceObject * IntObj, const int
nBatchMode)
```

In Visual Basic:

```
Private Function ExecuteTransaction(ByVal pIntObj As PsIntObj, ByVal
lBatchMode As Long) As ENUM_EIOCEXECSTATUS
```

In Java:

```
public int ExecuteTransaction(PSJInterfaceObject intobj)
```

Note. The nBatchMode and lBatchMode parameters are not currently used.

Get The Name of Your Business Interlink Object: GetObjName

Call the InterfaceObject method GetObjName to get the name of your Business Interlink Object. For the ExecuteTransaction method, this retrieves the name of the transaction. This is the name set in the XML design-time plug-in.

For more information about writing an XML design-time plug, see *PeopleSoft Business Interlink Design-Time Plug-in Programming Guide,* Writing an XML Design-Time Plug-In.

In the Freight Carrier example, the following code tests for the transaction named “Calculate Cost(Domestic)”, or in the cast of Java, “Shipping Time”.

In C++:

```
if(IntObj->GetObjName() == _T("Calculate Cost"))
{
// Insert the transaction code here: get input parameters, call
// the external system, insert output values.
}
```

In Visual Basic:

```
If pIntObj.ObjName = "Calculate Cost" Then
```

In Java:

```
if (intobj.GetObjName().equals("Shipping Time"))
```

For more information about the GetObjName method, see GetObjName, ObjName.

Get The Configuration Parameters

In the case of the Freight Carrier plug-in, there are no configuration parameters that are used in ExecuteTransaction.

For more information about getting configuration parameters, see GetConfigParams.

Get The Input Document: GetInputDocs, ResetCursor

Use the InterfaceObject method **GetInputDocs** to get the input document, which is a CBIDocs or PsBIDocs object. Then set the cursor to the top with the CBIDocs or PsBIDocs method **ResetCursor**. Setting the cursor to the top allows you to get the first of the input documents; or rather, the first set of input parameters.

In C++:

```
CBIDocs docsIn = IntObj->GetInputDocs(_T(""));
docsIn.ResetCursor();
```

In Visual Basic:

```
Set objInputDocs = pIntObj.GetInputDocs
objInputDocs.ResetCursor
```

In Java:

```
Set objInputDocs = pIntObj.GetInputDocs
objInputDocs.ResetCursor
```

For more information about the GetInputDocs method and the ResetCursor method, see GetInputDocs and ResetCursor.

Get The Output Document: GetOutputDocs, Clear

Use the InterfaceObject method **GetOutputDocs** to get the output document, which is a CBIDocs or BIDocs object. Then clear the output document with the CBIDocs or BIDocs method **Clear**. Clearing the output values allows you to later add values and documents to a cleared document that has no previous values in it.

In C++:

```
CBIDocs docsOut = IntObj->GetOutputDocs(_T(""));
```

```
docsOut.Clear();
```

In Visual Basic:

```
Set objOutputDocs = pIntObj.GetOutputDocs
objOutputDocs.Clear
```

In Java:

```
PSJBIDocs docsOut = intobj.GetOutputDocs("");
docsOut.Clear();
```

For more information about the GetOutputDocs method and the Clear method, see GetOutputDocs and Clear.

Get The Output Parameter Information: GetOutputParams, size

Use the InterfaceObject method **GetOutputParams** to retrieve information about the output parameters, which will be contained in the returned VARINFOLIST or PsEnumVarInfo object. In C++, use the VARINFOLIST method **size** to get the number of output parameters to retrieve.

The following code will get the information for the output parameters Rate and return_status by setting indexes pointing to the parameter information.

Note. Service_Rate is a document that is the output parameter for this transaction, so the names of the output parameters in the VARINFOLIST object are prefixed with "Service_Rate.".

Note. The output parameters return_status_msg and return_status are not created at design time, unlike the other output parameters. The return_status_msg and return_status output parameters are automatically created for every set of output parameters for a Business Interlink.

In C++:

Note. The C++ example for Calculate Cost uses one output parameter called Service_Rate, type object, containing the strings Service_Type, Guaranteed_By, and Rate.

```
int iReturnStatus = -1;
int iRate = -1;

int iServiceType = -1;
int iRate = -1;
int iGuaranteed = -1;
```

```
const VARINFOLIST& varListOutput = IntObj->GetOutputParams();
for(int i = 0; i < varListOutput.size(); i++)
{
   if(varListOutput[i]->m_strName == _T("return_status_msg"))
      iReturnMessage = i;
   else if(varListOutput[i]->m_strName == _T("return_status"))
      iReturnStatus = i;
   else if(varListOutput[i]->m_strName ==
      _T("Service_Rate.Service_Type")) iServiceType = i;
   else if(varListOutput[i]->m_strName ==
      _T("Service_Rate.Guaranteed_By")) iGuaranteed = i;
   else if(varListOutput[i]->m_strName == _T("Service_Rate.Rate"))
      iRate = i;
}
```

m_strName in objVarInfoName is part of the VARINFOLIST parameter list structure.

In Visual Basic:

Note. The Visual Basic example for Calculate Cost uses one output parameter called rate, type string.

```
Dim lResult As Long
Dim lIndex As Long
Dim objOutputParams As PsEnumVarInfo
Dim objVarInfo As VarInfo


Set objOutputParams = pIntObj.GetOutputParams
For Each objVarInfo In objConfigParams
   If objVarInfo.Name = "Rate" Then
      lResult = lIndex
   ElseIf objVarInfo.Name = "return_status_msg" Then
      lReturnMessage = lIndex
   ElseIf objVarInfo.Name = "return_status" Then
      lReturnStatus = lIndex
   End If
   lIndex = lIndex + 1
Next
```

Name in objVarInfoName is part of the VarInfo parameter list structure.

In Visual Basic:

```
int nCountOut = 0;
int noError = 0;
int iReturnMessage = -1
```

```
int iReturnStatus = -1
int iTime = -1
nCountOut = intobj.GetOutputParamCount();
for (int i=0; i<nCountOut; i++) {
   try
   {
      PSJVarInfo varinfo = intobj.GetOutputParam(i);
      if(varinfo.strName().equals("return_status_msg"))
         iReturnMessage = i;
      else if(varinfo.strName().equals("return_status"))
         iReturnStatus = i;
      else if(varinfo.strName().equals("Time")) iTime = i;
   }
   catch ( NoObjectReferenceException e)
   {
      return PSReturnStatus.FAILED;
   }
} //end for (int i=0; i<nCountOut; i++)
```

For more information about the parameter list structure used by VarInfo, PsEnumVarInfo, and VARINFOLIST, see Parameter Lists.

For more information about the GetOutputParams and size methods, refer to GetOutputParams, GetOutputParam, GetOutputParamCount and size.

Loop To Get Every Input Document and the Input Values: GetStatus, GetDoc, GetValue, GetNextDoc

A Business Interlink Object can have many input documents, or sets of input parameters, whose values you extract in the **ExecuteTransaction** method.

In this example, edited to show the **GetValue** methods better, use the CBIDocs or PsBIDocs method **GetStatus** method to loop through the input documents. Within the loop, use the CBIDocs or PsBIDocs method **GetDoc** to get the input parameters. Do this because the input parameters in this example are documents instead of basic types (such as integer, string, or float), so you must get the documents for these input parameters before you can get the parameter values. Then use the CBIDocs or PsBIDocs method **GetValue** to get the input values. At the end of the loop, use the CBIDocs or PsBIDocs method **GetNextDoc** to go to the next input document.

Note. You could also use the **GetCount** method to loop though each set of input parameters.

```
while(docsIn.GetStatus() == eNoError)  // loop at root level
{
```

```
   CBIDocs docFrom = docsIn.GetDoc(_T("From"));
   CBIDocs docTo = docsIn.GetDoc(_T("To"));
   CBIDocs docPackageInfo = docsIn.GetDoc(_T("Package_Info"));

   const TCHAR *pStr = docFrom.GetValue(_T("Country"));
   pStr = docFrom.GetValue(_T("Postal_Code"));

   pStr = docTo.GetValue(_T("Country"));
   pStr = docTo.GetValue(_T("Postal_Code"));
   pStr = docTo.GetValue(_T("City"));
   pStr = docTo.GetValue(_T("Address_Type"));

   pStr = docPackageInfo.GetValue(_T("Drop_off_Pickup"));
   pStr = docPackageInfo.GetValue(_T("Packaging"));
   pStr = docPackageInfo.GetValue(_T("Weight"));
   pStr = docPackageInfo.GetValue(_T("Length"));
   pStr = docPackageInfo.GetValue(_T("Width"));
   pStr = docPackageInfo.GetValue(_T("Height"));

/* Call the external system using these input values, insert the output values
into the output document (code not shown here) */

   docsIn.GetNextDoc();
} //end while
```

In Visual Basic:

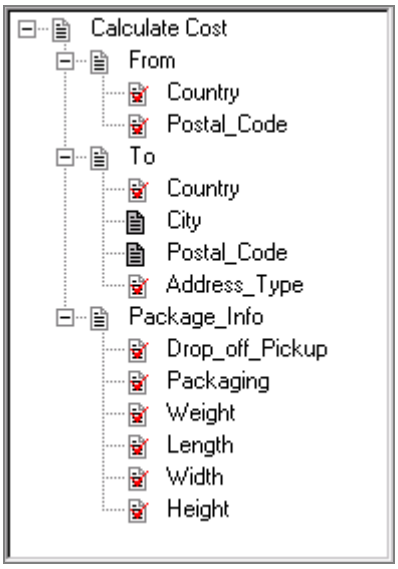
```
Dim eNoError As Integer
Dim objInputDocs As PsBIDocs
Dim objDocFrom As PsBIDocs
Dim objDocTo As PsBIDocs
Dim objDocPackInfo As PsBIDocs


While objInputDocs.GetStatus = eNoError

   Set objDocFrom = objInputDocs.GetDoc("From")
   Set objDocTo = objInputDocs.GetDoc("To")
   Set objDocPackInfo = objInputDocs.GetDoc("Package_Info")

   lOrigCountry = objDocFrom.GetValue("OriginCountry")
   lOrigPostalCode = objDocFrom.GetValue("OriginPostal_Code")

   lDestCountry = objDocTo.GetValue("DestCountry")
   lDestPostalCode = objDocTo.GetValue("DestPostal_Code")
```

```
    lWeight = objDocPackInfo.GetValue("Weight")
    lLength = objDocPackInfo.GetValue("Length")
    lWidth = objDocPackInfo.GetValue("Width")
    lHeight = objDocPackInfo.GetValue("Height")
    lPackaging = objDocPackInfo.GetValue("Packaging")
    lDrop_Off_Pickup = objDocPackInfo.GetValue("Drop_Off_Pickup")

    ' Call the external system using these input values, insert the
    ' output values into the output document (code not shown here)

    objInputDocs.GetNextDoc
Wend
```

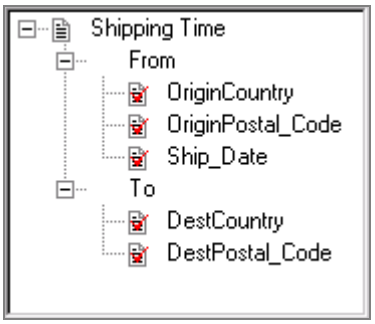
In Java:

```
while (docsIn.getStatus() == noError) {
    docsIn.ResetCursor();
    PSJBIDocs from = docsIn.GetDoc("From");
    PSJBIDocs to = docsIn.GetDoc("To");

    String fromOriginCountry = from.GetValue("OriginCountry");
    String fromOriginPostal_Code = from.GetValue("OriginPostal_Code");
    String fromShip_Date = from.GetValue("Ship_Date");
    String toDestCountry = to.GetValue("DestCountry");
    String toDestPostal_Code = to.GetValue("DestPostal_Code");
    docsIn.getNextDoc();

    // Call the external system using these input values, insert the
    // output values into the output document (code not shown here)
}
docsIn.destroy();
```

The figures below shows the input documents for this example. For Calculate Cost, the input document contains four input parameters: From, To, Package_Info, and Account_Info. For Shipping Time, the input document contains two input parameters: From and To. Since the input parameters are documents, you must get the documents if you want to get their values. The figures below show the documents that **GetDoc** gets in this example.



Example Input Document: Calculate Cost



Example Input Document: Shipping Time

For more information about GetCount, GetDoc, GetNextDoc, GetStatus, and GetValue, see GetCount, GetDoc, GetNextDoc, GetStatus, and GetValue.

Call The External System

Within the loop to get the input values, call the external system, passing it the input parameter values. The code you write here depends upon your system. This will provide the values that you will add to the output documents.

Add The Output Documents and Their Output Values: AddDoc, AddValue, AddNextDoc

Loop to add each output document (set of output parameters) and add the output values to the output document.

Note. In this example, for each set of input parameters, there is a corresponding set of output parameters. Therefore, the loop to add the output values is the loop to get the input values.

Use the method **AddNextDoc** to add an output document, testing with the method **Empty** to make sure that you are adding to an existing output document structure. Then use the method **AddDoc** to add the output parameters. You do this because some of the output parameters in this example are documents instead of basic types (such as integer, string, or float), so you must add the documents for these output parameters before you can add the parameter values. Then use the method **AddValue** to add the output values to the output parameters.

The output parameters return_status and return_status_msg are used in every Business Interlink object. return_status is any status number that you would like to have the Business Interlink Object return; return_status_Msg is any status message that you would like to have the Business Interlink Object return. They are not documents, so you do not use **AddDoc** to add them before you add their values with **AddValue**.

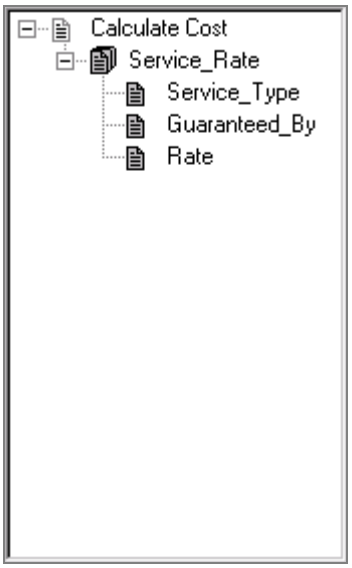
The following code adds values for Rate, which is part of the Service_Rate output parameter/document, and the output parameter return_status. This code is contained within the loop to get each set of input parameters.

Note. The **GetOutputDocs** method provided the pointer to the first output document.

```
// Loop to get a set of input parameters and call the external
// system (code not shown).

if(!docsOut.Empty()) docsOut.AddNextDoc();
   IOSTRINGLIST listServiceType, listGuaranteed, listRates;
   if(upsinfo.GetCost(listServiceType, listGuaranteed, listRates))
   {
      CBIDocs docServer = docsOut.AddDoc(_T("Service_Rate"));
         for(int i = 0; i < listServiceType.size(); i++)
         {
            if(iServiceType != -1)
               docServer.AddValue(_T("Service_Type"),
                  listServiceType[i].c_str());
            if(iGuaranteed != -1)
               docServer.AddValue(_T("Guaranteed_By"),
                  listGuaranteed[i].c_str());
            if(iRate)
               docServer.AddValue(_T("Rate"), listRates[i].c_str());

            if(i < listServiceType.size() - 1)
               docServer.AddNextDoc();
         }
     }
   }
   docsOut.AddValue(_T("return_status"), _T("0"));
```

```
    docsOut.AddValue(_T("return_status_msg"), _T("Ok"));

// End of loop (code not shown)
```

The figure below shows the output document for the C++ example. It contains one output parameter: Service_Rate. Since the output parameters is a document, it must be added if you want to add its values. (Actually, since it is a document list, you need to add it for every document in the document list.) The figure below shows the Service_Rate document that **AddDoc** adds in the C++ example.



Example CBIDocs Output Document for C++

In Visual Basic:

Note. In the Visual Basic example, the output parameters are not contained in a document; they are of simple type.

```
' Set up a loop to go through all 8 service types
Dim CNT As Integer
Dim lResult As Long
Dim lResult2 As Long
Dim lService_Type As String
Dim objOutputDocs As PsBIDocs


CNT = 1
While CNT < 8

   Select Case CNT
      Case 1
```

```
            lService_Type = "Next Day Air Early A.M."
         Case 2
            lService_Type = "Next Day Air"
         ' rest of cases not shown
         End Select

      ' Set values for lRate, lGuarantee (code not shown)

      ' Test the CBIDocs output document to make sure it is not
      ' empty, and then add a document to it so you can add
      ' values to a set of output parameters.
      If objOutputDocs.Empty <> 1 Then
         lResult = objOutputDocs.AddNextDoc
      End If

      If lResult <> -1 Then
         lResult = objOutputDocs.AddValue("Rate", lRate)
         lResult = objOutputDocs.AddValue("Guarenteed_by",
                   lGuarantee)
         lResult = objOutputDocs.AddValue("Service_Type",
                   lService_Type)
      End If

      If lResult <> -1 Then
         lResult2 = objOutputDocs.AddValue("return_status_msg",
                    "Success")
      End If

      If lResult <> -1 Then
         lResult2 = objOutputDocs.AddValue("return_status", "0")
      End If

      CNT = CNT + 1
   Wend
   objInputDocs.GetNextDoc
```

The figure below shows the output document for the Visual Basic example. It contains three output parameters: Rate, Guarenteed_by, and Service_Type.

Example PsBIDocs Output Document for Visual Basic

In Visual Basic:

Note. In the Java example, the output parameters are not contained in a document; they are of simple type.

```
docsOut.AddValue("Time",sTime);
docsOut.AddValue("return_status_msg", return_status_msg);
docsOut.AddValue("return_status", "0" );
```

Example PSJBIDocs Output Document for Java

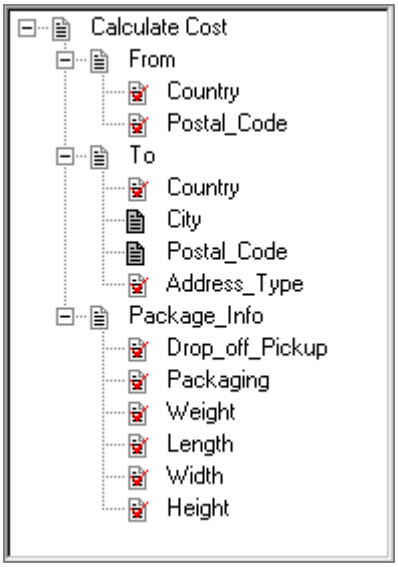
For more information about Empty, AddDoc, AddValue, and AddNextDoc, see Empty, AddDoc, AddValue, AddValueInt, AddValueFloat, AddValueDouble, and AddNextDoc.

CHAPTER 5

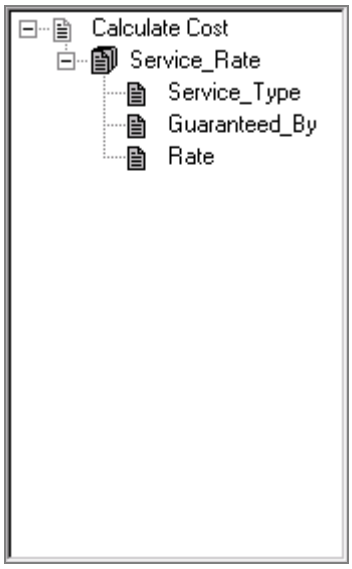
Examples of Business Interlink Runtime Plug-in Code

ExecuteTransaction in C++

For reference, here is the structure of the input and output documents for the Calculate Cost Business Interlink Object.



Calculate Cost Business Interlink Object Inputs



Calculate Cost Business Interlink Object Outputs

The following code shows the ExecuteTransaction for a UPS plug-in. It determines which transaction is being passed in with this Business Interlink Object, and processes the inputs and outputs for that transaction. The transaction is the Calculate Cost transaction.

```
EIOCEXECSTATUS CUps::ExecuteTransaction(InterfaceObject * IntObj, const int
nBatchMode)
{

// Create the CBIDocs output document and get a reference to it, and
// then clear the output document.
   CBIDocs docsOut = IntObj->GetOutputDocs(_T(""));
   docsOut.Clear();

// Create the CBIDocs input document and get a reference to it.
   CBIDocs docsIn = IntObj->GetInputDocs(_T(""));
   docsIn.ResetCursor();
   CTime t;
   int iReturnMessage = -1;
   int iReturnStatus = -1;

   // Get the transaction name. You need the name in order to know
   //  which inputs and outputs you will use, and how you want to
   // process them.
   if(IntObj->GetObjName() == _T("Calculate Cost"))
   {
      int iServiceType = -1;
      int iRate = -1;
      int iGuaranteed = -1;
```

```
      // Get the output parameter names.
      const VARINFOLIST& varListOutput = IntObj->GetOutputParams();
      for(int i = 0; i < varListOutput.size(); i++)
      {
         if(varListOutput[i]->m_strName == _T("return_status_msg"))
               iReturnMessage = i;
         else if(varListOutput[i]->m_strName == _T("return_status"))
               iReturnStatus = i;
         else if(varListOutput[i]->m_strName ==
               _T("Service_Rate.Service_Type")) iServiceType = i;
         else if(varListOutput[i]->m_strName ==
               _T("Service_Rate.Guaranteed_By")) iGuaranteed = i;
         else if(varListOutput[i]->m_strName ==
               _T("Service_Rate.Rate")) iRate = i;
      }

// Start of loop to read each set of input documents (sets of input
// parameters) for this transaction.
      while(docsIn.GetStatus() == eNoError) // loop at root level
      {
// Get the input parameters that are documents (documents are
// structures: they contain parameters)
         CBIDocs docFrom = docsIn.GetDoc(_T("From"));
         CBIDocs docTo = docsIn.GetDoc(_T("To"));
         CBIDocs docPackageInfo = docsIn.GetDoc(_T("Package_Info"));


         CString strPostData =
               _T("accept_UPS_license_agreement=yes");

// Get the value of Country, which is part of the From
// input parameter/document
         const TCHAR *pStr = docFrom.GetValue(_T("Country"));
         if(pStr)
         {
            strPostData += _T("&14_origCountry=");
            strPostData += GetValue(tableCountry, pStr);
         }

// Get the value of Postal_Code, which is part of the From
// input parameter/document
         pStr = docFrom.GetValue(_T("Postal_Code"));
         if(pStr)
         {
            strPostData += _T("&15_origPostal=");
```

```
            strPostData += pStr;
         }

// Get the value of Country, which is part of the To
// input parameter/document
         pStr = docTo.GetValue(_T("Country"));
         if(pStr)
         {
// Get the value of Postal_Code, which is part of the To
// input parameter/document
            strPostData += _T("&22_destCountry=");
            strPostData += GetValue(tableCountry, pStr);
         }
         pStr = docTo.GetValue(_T("Postal_Code"));
         if(pStr)
         {
            strPostData += _T("&19_destPostal=");
            strPostData += pStr;
         }
         /*
// Get the value of City, which is part of the To
// input parameter/document
         pStr = docTo.GetValue(_T("City"));
         if(pStr)
         {
            strPostData += _T("&20_origCity=");
            strPostData += pStr;
         }
         */
         strPostData += _T("&20_origCity=");
         strPostData += pStr;
// Get the value of Address_Type, which is part of the To
// input parameter/document
         pStr = docTo.GetValue(_T("Address_Type"));
         if(pStr)
         {
            strPostData += _T("&49_residential=");
            strPostData +=  GetValue(tableAddressType, pStr);
         }

// Get the value of Drop_off_Pickup, which is part of the
// Package_Info input parameter/document
         pStr = docPackageInfo.GetValue(_T("Drop_off_Pickup"));
         if(pStr)
         {
```

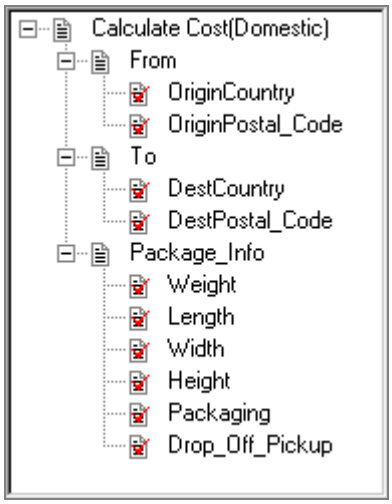
```
            strPostData += _T("&47_rate_chart=");
            strPostData +=  _T("Regular Daily Pickup"); // pStr;
         }

// Get the value of Packaging, which is part of the
// Package_Info input parameter/document
         pStr = docPackageInfo.GetValue(_T("Packaging"));
         if(pStr)
         {
            strPostData += _T("&48_container=");
            strPostData +=  _T("00");
               // GetValue(tableContainer, pStr);
         }

// Get the value of Weight, which is part of the
// Package_Info input parameter/document
         pStr = docPackageInfo.GetValue(_T("Weight"));
         if(pStr)
         {
            strPostData += _T("&23_weight=");
            strPostData +=  pStr;
            strPostData += _T("&weight_std=lbs.");
         }

// Get the value of Length, which is part of the
// Package_Info input parameter/document
         pStr = docPackageInfo.GetValue(_T("Length"));
         if(pStr)
         {
            strPostData += _T("&25_length=");
            strPostData +=  pStr;
         }

// Get the value of Width, which is part of the
// Package_Info input parameter/document
         pStr = docPackageInfo.GetValue(_T("Width"));
         if(pStr)
         {
            strPostData += _T("&26_width=");
            strPostData +=  pStr;
         }

// Get the value of Height, which is part of the
// Package_Info input parameter/document
         pStr = docPackageInfo.GetValue(_T("Height"));
```

```
            if(pStr)
            {
               strPostData += _T("&27_height=");
               strPostData +=  pStr;
            }

            strPostData += _T("&length_std=in.");
            strPostData.Replace(_T(" "), _T("+"));

            CInet iNet;
            PSIOString strFileName = _T("c:\\temp\\abc.html");
               //GenerateTempFileName();

// Execute the transaction by calling the external system
            if(iNet.PostRequest(
                _T("http://www.ups.com/using/services/rave/qcost.cgi"),
                (TCHAR*)(LPCTSTR)strPostData,
                strPostData.GetLength(),
                (TCHAR *)strFileName.c_str()))
            {
               CUpsInfo upsinfo(strFileName);
// Test the output document to make sure it is not empty,
// and then add a document to it so you can add values to a set of
// output parameters.
               if(!docsOut.Empty()) docsOut.AddNextDoc();

               IOSTRINGLIST listServiceType, listGuaranteed,
                  listRates;
               if(upsinfo.GetCost(listServiceType, listGuaranteed,
                  listRates))
               {
// Since the output parameter Service_Rate is a document, add a
// Service_Rate document to allow its values to be added.
                  CBIDocs docServer =
                        docsOut.AddDoc(_T("Service_Rate"));
// Since Service_Rate is a document list, loop to add all of the
// Service Rate data that you have.
                  for(int i = 0; i < listServiceType.size(); i++)
                  {
// Add a value for Service_Type, which is part of the Service_Rate
// output parameter/document.
                     if(iServiceType != -1)
                        docServer.AddValue(_T("Service_Type"),
                        listServiceType[i].c_str());
// Add a value for Guaranteed_By, which is part of the Service_Rate
```

```
// output parameter/document.
                  if(iGuaranteed != -1)
                     docServer.AddValue(_T("Guaranteed_By"),
                     listGuaranteed[i].c_str());
// Add a value for Rate, which is part of the Service_Rate
// output parameter/document.
                  if(iRate)
                     docServer.AddValue(_T("Rate"),
                     listRates[i].c_str());

// Add another Service_Rate output parameter/document to the
// document list.
                  if(i < listServiceType.size() - 1)
                     docServer.AddNextDoc();
               }   // end for loop to add all of the // Service Rate
                    // data that you have.
            }   // end if (upsinfo.GetCost)
         }  //end if (execute transaction)
         // Add values for the output parameters return_status_msg
         // and return_status.
         docsOut.AddValue(_T("return_status"), _T("0"));
         docsOut.AddValue(_T("return_status_msg"), _T("Ok"));
         docsIn.GetNextDoc();
      }  // end of while loop to read each set of input documents
   }  // end of if that gets the transactions name (Calculate Cost)
   return INTOBJ_SUCCEED;
}
```

ExecuteTransaction in Visual Basic

For reference, here is the structure of the input and output documents for the Calculate Cost(Domestic) Business Interlink Object.



Calculate Cost(Domestic) Business Interlink Object Inputs

Calculate Cost(Domestic) Business Interlink Object Outputs

The following code shows the ExecuteTransaction for a Freight Carrier plug-in. It determines which transaction is being passed in with this Business Interlink Object, and processes the inputs and outputs for that transaction. The transaction is the Calculate Cost(Domestic) transaction.

```
Private Function ExecuteTransaction(ByVal pIntObj As PsIntObj, ByVal
lBatchMode As Long) As ENUM_EIOCEXECSTATUS
    '
    Dim bRtnVal As Boolean
    Dim objVariantInfo As Variant
    Dim objVarInfo As VarInfo
    Dim objOutputParams As PsEnumVarInfo
    Dim lResult As Long
    Dim lResult2 As Long
    Dim lIndex As Long
```

```
  Dim lReturnMessage As Long
  Dim lReturnStatus As Long
  Dim objOutputDocs As PsBIDocs
  Dim objInputDocs As PsBIDocs
  Dim objDocFrom As PsBIDocs
  Dim objDocTo As PsBIDocs
  Dim objDocPackInfo As PsBIDocs
  Dim objDocService As PsBIDocs
  Dim objPsBstr As PsBstr
  Dim eNoError As Integer
  eNoError = 0
  bRtnVal = True
  lIndex = 0
  lResult = -1
  lReturnMessage = -1
  lReturnStatus = -1
  '

' Create the CBIDocs output document and get a reference to it,
' and then clear the output document.
  Set objOutputDocs = pIntObj.GetOutputDocs
  objOutputDocs.Clear

' Get the output parameter information.
  Set objOutputParams = pIntObj.GetOutputParams
  For Each objVarInfo In objOutputParams
      If objVarInfo.Name = "Rate" Or objVarInfo.Name = "Time" Then
          lResult = lIndex
      ElseIf objVarInfo.Name = "return_status_msg" Then
          lReturnMessage = lIndex
      ElseIf objVarInfo.Name = "return_status" Then
          lReturnStatus = lIndex
      End If
      lIndex = lIndex + 1

  Next

  Dim pValue As String
  Dim iSum As Integer
  Dim iDResult As Integer
  Dim varValue1 As Variant
  Dim varValue2 As Variant
  Dim varValue3 As Variant

  Dim lOrigCountry As String
```

```
    Dim lOrigPostalCode As String
    Dim lDestCountry As String
    Dim lDestPostalCode As String
    Dim lLength As String
    Dim lWeight As String
    Dim lWeight_Type As String
    Dim lWidth As String
    Dim lHeight As String
    Dim lPackaging As String
    Dim lDrop_Off_Pickup As String

' Get the name of the Calculate Cost transaction
    If pIntObj.ObjName = "Calculate Cost" Then
        Dim lRate As String
        Dim lGuarantee As String
        Dim lService_Type As String

      ' Create the CBIDocs input document and get a reference to it.
        Set objInputDocs = pIntObj.GetInputDocs
        objInputDocs.ResetCursor

     ' Get the input parameters that are documents

      ' Start of loop to read each set of input documents (sets of
      ' input parameters) for this transaction.
        While objInputDocs.GetStatus = eNoError

            Set objDocFrom = objInputDocs.GetDoc("From")
            Set objDocTo = objInputDocs.GetDoc("To")
            Set objDocPackInfo = objInputDocs.GetDoc("Package_Info")

            lRate = ""
            lGuarantee = ""
            lService_Type = ""

          ' Get the value of OriginCountry, OriginPostal_Code, and
          ' Ship_Date, which are part of the From input
          ' parameter/document
            lOrigCountry = objDocFrom.GetValue("OriginCountry")
            lOrigPostalCode =
               objDocFrom.GetValue("OriginPostal_Code")

          ' Get the value of DestCountry and DestPostal_Code, which
          ' are part of the To input parameter/document
            lDestCountry = objDocTo.GetValue("DestCountry")
```

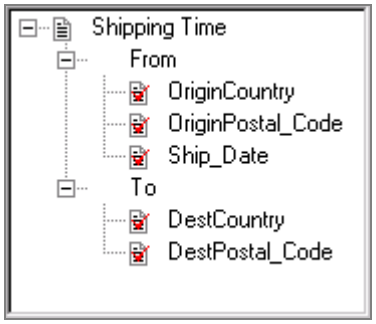
```
lDestPostalCode = objDocTo.GetValue("DestPostal_Code")

' Get the value of Weight, Weight_Type, Declared_Value, and
' Service_Type, which are part of the Package_Info input
' parameter/document
lWeight = objDocPackInfo.GetValue("Weight")
lLength = objDocPackInfo.GetValue("Length")
lWidth = objDocPackInfo.GetValue("Width")
lHeight = objDocPackInfo.GetValue("Height")
lPackaging = objDocPackInfo.GetValue("Packaging")
lDrop_Off_Pickup =
   objDocPackInfo.GetValue("Drop_Off_Pickup")

'lRate = CInt(Left(lOrigPostalCode, 1) &
'Right(lDestPostalCode, 1))
 Dim CNT As Integer
 CNT = 1
 While CNT < 8

 Select Case CNT
 Case 1
     lService_Type = "Next Day Air Early A.M."
 Case 2
     lService_Type = "Next Day Air"
 Case 3
     lService_Type = "Next Day Air Saver"
 Case 4
     lService_Type = "2nd Day Air Early A.M."
 Case 5
     lService_Type = "2nd Day Air"
 Case 6
     lService_Type = "3rd Day Select"
 Case 7
     lService_Type = "Ground"
 End Select

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

Set objFreight = New Freight

objFreight.OriginCountry = lOrigCountry
objFreight.OriginPostalCode = lOrigPostalCode
objFreight.DestCountry = lDestCountry
objFreight.DestPostalCode = lDestPostalCode
```

```
    objFreight.ServiceType = lService_Type
    objFreight.Weight = lWeight
    objFreight.Length = lLength
    objFreight.Width = lWidth
    objFreight.Height = lHeight
    objFreight.Packaging = lPackaging
    objFreight.Drop_Off_Pickup = lDrop_Off_Pickup

    lRate = objFreight.Cost
    lGuarantee = objFreight.Guarantee_By

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

  ' Test the CBIDocs output document to make sure it is not
  ' empty, and then add a document to it so you can add
  ' values to a set of output parameters.
    If objOutputDocs.Empty <> 1 Then
      lResult = objOutputDocs.AddNextDoc
    End If

    If lResult <> -1 Then

      ' Add values for the output parameters Rate,
      ' Guarenteed_by, and Service_Type.
      lResult = objOutputDocs.AddValue("Rate", lRate)
      lResult = objOutputDocs.AddValue("Guarenteed_by",
                lGuarantee)
      lResult = objOutputDocs.AddValue("Service_Type",
                lService_Type)

    End If
      ' Add values for the output parameters
      ' return_status_msg and return_status.
    If lResult <> -1 Then
      lResult2 = objOutputDocs.AddValue("return_status_msg",
                 "Success")
    End If
    If lResult <> -1 Then
      lResult2 = objOutputDocs.AddValue("return_status", "0")
    End If


    CNT = CNT + 1
  Wend
    objInputDocs.GetNextDoc
```

```
        Wend



    End If

    ExecuteTransaction = ENUM_INTOBJ_SUCCEED
    '
End Function
```

ExecuteTransaction in Java

For reference, here is the structure of the input and output documents for the Shipping Time Business Interlink Object.



Shipping Time Business Interlink Object Inputs

Shipping Time Business Interlink Object Outputs

The following code shows the ExecuteTransaction for a Freight Carrier plug-in. It determines which transaction is being passed in with this Business Interlink Object, and processes the inputs and outputs for that transaction. The transaction is the Shipping Time transaction.

```
public int ExecuteTransaction(PSJInterfaceObject intobj){

    String varstring = new String();

    String return_status_msg = new String("Succeded");
```

```
int nCountOut = 0;
int noError = 0;
// Get the output parameters.
nCountOut = intobj.GetOutputParamCount();
for (int i=0; i<nCountOut; i++) {
   try
   {
      PSJVarInfo varinfo = intobj.GetOutputParam(i);
   }
   catch ( NoObjectReferenceException e)
   {
      return PSReturnStatus.FAILED;
   }
} //end for (int i=0; i<nCountOut; i++)

// if transaction is "Shipping Time"
// Get the transaction name. You need the name in order to know
//  which inputs and outputs you will use, and how you want to
// process them.
if (intobj.GetObjName().equals("Shipping Time"))
{
   int ship_time;
   ship_time=0;
   // Standard initialization code.  PSJBIDocs allocate
   // memory on the C++ side so we must be sure to call the
   // destroy routine prior to leaving scope.
   try
   {
      // Create the CBIDocs input document and get a reference to it.
      PSJBIDocs docsIn = intobj.GetInputDocs("");
   }
   catch ( NoObjectReferenceException e)
   {
      return_status_msg = "Failed to get Input";
      return PSReturnStatus.FAILED;
   }
// Start of loop to read each set of input documents (sets of input
// parameters) for this transaction.
   while (docsIn.getStatus() == noError) {
      docsIn.ResetCursor();
      // Get the input parameters that are documents (documents are
      // structures: they contain parameters)
      PSJBIDocs from = docsIn.GetDoc("From");
      PSJBIDocs to = docsIn.GetDoc("To");
```

```
// Get the values of OriginCountry, OriginPostal_Code, and Ship_Date, which
are
// part of the From input parameter/document
         String fromOriginCountry = from.GetValue("OriginCountry");
         String fromOriginPostal_Code = from.GetValue("OriginPostal_Code");
         String fromShip_Date = from.GetValue("Ship_Date");

// Get the values of DestCountry and DestPostal_Code, which are
// part of the To input parameter/document
         String toDestCountry = to.GetValue("DestCountry");
         String toDestPostal_Code = to.GetValue("DestPostal_Code");

// Execute the transaction by calling the external system
         ship_time = shipTime(
            toDestCountry,toDestPostal_Code,fromOriginCountry,
            fromOriginPostal_Code);

         // process output definition
         try {
            // Create the PSJBIDocs output document and get a reference to it,
            // and then clear the output document.
            PSJBIDocs docsOut = intobj.GetOutputDocs("");
            docsOut.Clear();
            String sTime = java.lang.Integer.toString(ship_time);

            // Fill in the root entry
// Add a value for Time, which is an output parameter.
            docsOut.AddValue("Time",sTime);
// Add values for the output parameters return_status_msg
// and return_status.
            docsOut.AddValue("return_status_msg", return_status_msg);
            docsOut.AddValue("return_status", "0" );

            // We explicitly call this to cleanup C++ memory behind the
            // Java Doc objects which were created during Java processing.
            // The other objects, such as the PSJInterface object are
allocated
            // by the caller (framework) and is just wrappered for use here so
            // we are not responsible for freeing that memory.
            docsOut.destroy();
         }
         catch ( NoObjectReferenceException e)
         {
            return PSReturnStatus.FAILED;
         }
         docsIn.getNextDoc();
```

```
      }
      //end while that loops through input document
      docsIn.destroy();

   }  //end if transaction is "Shipping Time"

   return PSReturnStatus.OK;
}//end Execute()

public int shipTime ( String country_to, String zip_to, String country_from,
String zip_from)
{
   int zip_diff= java.lang.Integer.parseInt(zip_from, 10) -
java.lang.Integer.parseInt(zip_to, 10);
   int country_diff= country_to.compareTo( country_from);

   if (country_diff == 0){

      if (zip_diff == 0)
         return 1;
      else
         return zip_diff* 2;
      }
   else
   {
      return (zip_diff + country_diff) * 2;
   }
}
```

CHAPTER 6

Deploying A Business Interlink Runtime Plug-in

After you have written and tested your Business Interlink runtime plug-ins, you can deploy them for others to use.

Placing on PeopleSoft Application Clients for Testing

On Windows NT, if you want to test your Business Interlink runtime plug-in using the Business Interlink tester, you must place it on the PeopleSoft application client.

For more information on the Business Interlink tester, see *PeopleSoft Business Interlink Application Developer Guide*, Business Interlink Tester Page: Testing Your Business Interlink Definition.

To place your Business Interlink runtime plug-in on a PeopleSoft application client, ensure that its DLL file (or equivalent if you are using UNIX) is located in the following location:

```
<PS_HOME>\bin\client\winx86\interfacedrivers
```

Where <PS_Home> is the directory where PeopleTools is installed.

Deploying on PeopleSoft Application Servers

To deploy your Business Interlink runtime plug-in on a PeopleSoft application server, ensure that its DLL file (or equivalent if you are using UNIX) is placed into the following location:

```
<PS_Home>\bin\client\winx86\interfacedrivers
```

Where <PS_Home> is the directory where PeopleTools is installed.

Deploying on Web Servers

To deploy your Business Interlink runtime plug-in on a Windows NT web server where the Business Interlink runtime environment has been installed, ensure that its DLL file is placed into the following location:

For Microsoft IIS, the default location is:

```
c:\inetpub\wwwroot\PsInterlinks\interfacedrivers
```

For Apache Web Server, the default location is:

```
c:\program files\apache jserv 1.1\PsInterlinks\interfacedrivers
```

If the default location was not used when Business Interlink runtime environment was installed, contact the person who installed it to get the location.

For more information about installation, see the Installing External Installation chapter in ***PeopleSoft 8 Installation and Administration***.

For this deployment to work, your runtime plug-in must use the same name as the name of the XML design-time plug-in. For example, if runtime is fcarrier.dll, the design-time must be fcarrier.xml.

For more information about the XML design-time plug-in, see the *PeopleSoft Business Interlink Design-Time Plug-in Programming Guide*, Writing an XML Design-Time Plug-In.

CHAPTER 7

Testing A Business Interlink Plug-in

The Business Interlink Tester allows you to test your Business Interlink. If you are creating a Business Interlink runtime plug-in, you can test it using the Business Interlink Tester. If you have PeopleTools installed, you can design a Business Interlink Definition and use the Business Interlink Tester within the PeopleTools Application Designer to test the Business Interlink Definition. In each case, the Business Interlink Tester executes the runtime plug-in, sending it input values and then receiving output values.

Using the Business Interlink Tester: Windows NT

In order to run the Business Interlink Tester for Windows NT, you must have deployed both the runtime plug-in and the XML design-time plug-in. The instruction for running the Business Interlink tester for Windows NT are in the PeopleSoft Business Interlink Application Developer Guide.

For more information about the XML design-time plug-in, see PeopleSoft Business Interlink Design-Time Plug-in Programming Guide.

For more information on running the Business Interlink Tester under Windows NT, see the *PeopleSoft Business Interlink Application Developer Guide*, Business Interlink Tester Page: Testing Your Business Interlink Definition.

Using the Business Interlink Tester: UNIX

Once you have set up Business Interlinks on UNIX, you can test your Business Interlink runtime plug-ins using the Business Interlink tester for UNIX. You can test runtime plug-ins that you write, and you can test the simple runtime plug-in.

To run the Business Interlink tester for UNIX:

1. Create an Input Doc XML file.

For more information on creating an Input Doc XML file, see Example of an Input Doc XML file and Writing the Tags in an Input Doc XML File.

---

2. Save the Business Interlink XML design-time plug-in file and the Input Doc XML file into the <PS_HOME>/bin directory.

   For more information about the XML design-time plug-in, see PeopleSoft Business Interlink Design-Time Plug-in Programming Guide.

3. Within the <PS_HOME>/bin directory, run the following command to run the Business Interlink Tester for UNIX.

   ```
   bitest  designplugin.xml inputdoc.xml [output.xml]
   ```

   If any of the XML files are not in the same directory as the bitest executable, you must specify the path for them. If you are not currently in the same directory as the bitest executable, you must specify the path for it.

   The 1st parameter, *designplugin.xml*, is the Business Interlink XML design time plug-in (or XML design-time plug-in). The file named simple.xml, which is provided and untared with the Setting up Business Interlinks on UNIX instructions, is an XML design-time plug-in.

   The 2nd parameter, *inputdoc.xml*, is an Input Doc XML file. The files named addnumber.xml, catstr.xml, and machine.xml, which are provided and untared with the Setting up Business Interlinks on UNIX instructions, are all Input Doc XML files.

   The 3rd parameter, *output.xml*, is the Output Doc XML file, where the output values from the execution of the runtime plug-in will be stored. This parameter is optional; if you do not supply it, the default file name is output.xml. If you provide a path with the file name, the Output Doc XML file will be located in that path. Otherwise, the file will be located in the bin directory (the same directory as bitest.exe).

Example of an Input Doc XML file

The Input Doc XML file contains the input values that the Business Interlink Tester needs to execute the runtime plug-in and return output values.

Here is the XML design-time plug-in code that contains a transaction named “add numbers”. This transaction adds two numbers.

```
<trans_catalog>
   <category name="simple transactions">
      <transaction name="add numbers">
         <input_list>
            <input name="number_1" type="int" required="true"/>
            <input name="number_2" type="int" />
         </input_list>
         <output_list>
            <output name="sum" type="int"/>
         </output_list>
```

```
      </transaction>
   </category>
</trans_catalog>
```

Here is the corresponding Input Doc XML file that adds input data for the “add numbers” transaction.

```
<?xml version="1.0"?>
<Business_Interlink>
   <Definition>
      <Header>
         <InterfaceName Value="Untitled">
         </InterfaceName>
      </Header>
   </Definition>
   <Inputs>
      <number_1>3</number_1>
      <number_2>23</number_2>
   </Inputs>
</Business_Interlink>
```

Writing the Tags in an Input Doc XML File

The main tag for the Business Interlink XML design-time plug-in is **<Business_Interlink>**. It contains the **<Definition>** and **<Inputs>** tags.

<Definition>, <Header>, <InterfaceName> Write the definition for this Business Interlink

In the **<Definition>** tag, write the **<InterfaceName>** tag to contain the name of this Business Interlink definition.

Here is an example of how to set the **<Definition>**, **<Header>**, and **<InterfaceName>** tags.

```
<Definition>
   <Header>
      <InterfaceName Value="Untitled">
      </InterfaceName>
   </Header>
</Definition>
```

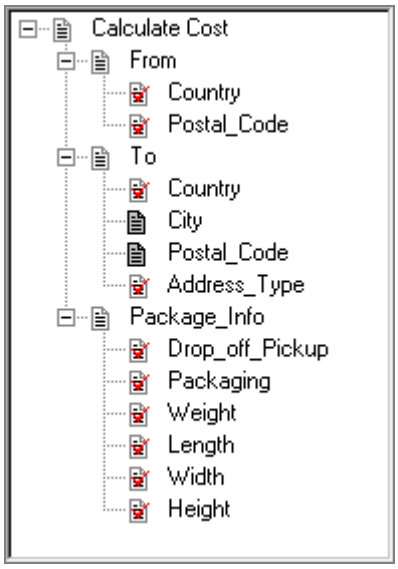
<Inputs> Write the input values for this Business Interlink

In the **<Inputs>** tag, write a tag to set the values for each input parameter.

Here is an example of how to set the **<Inputs>** tag. It sets the values of the input parameters number_1 and number_2 to 3 and 23.

```
<Inputs>
   <number_1>3</number_1>
   <number_2>23</number_2>
</Inputs>
```

Here is an example of an input doc for the Freight Carrier plug-in. It shows how to set the Inputs tag for the following structure:



Calculate Cost Business Interlink Object Inputs

```
<?xml version="1.0"?>
<Business_Interlink>
   <Definition>
      <Header>
         <InterfaceName Value="Test">
         </InterfaceName>
      </Header>
   </Definition>
   <Inputs>
      <root>
         <From>
            <Country>United States</Country>
            <Postal_Code>05550</Postal_Code>
         </From>
         <To>
            <Country>United States</Country>
            <City>San Mateo</City>
            <Postal_Code>94404</Postal_Code>
         </To>
         <Package_Info>
            <Drop_off_Pickup>Regular Daily Pickup</Drop_off_Pickup>
```

```
            <Packaging>Your Packaging</Packaging>
            <Weight>1</Weight>
            <Length>4</Length>
            <Width>4</Width>
            <Height>4</Height>
          </Package_Info>
      </root>
   </Inputs>
</Business_Interlink>
```

CHAPTER 8

Understanding The Business Interlink Methods

InterfaceObject Class Methods: Use these methods when you write your execution method (or methods) for your Interlink class.

Plug-in Class Methods: Runtime. Write these methods when you are creating a runtime plug-in. The execution methods are in this class.

CBIDocs methods: Use these methods to access the input and output values of a Business Interlink Object when the input and output is hierarchical.

PSIOString methods: Use these methods to access the information in a PSIOString, which is how the input and output parameter information (names, data types) for a Business Interlink Object are stored.

InterfaceObject Class Methods: Accessing Interlink Object Data

The InterfaceObject Class contains the methods that you use when you write your execution method (or methods) for your Interlink class. A Business Interlink Object is passed to your execution methods, and you use the InterfaceObject class methods to retrieve information about the Business Interlink Object.

You must include the header file iocls.h for this class.

The following table shows which InterfaceObject methods you use to retrieve Business Interlink Object information.

To retrieve:	***Use this method***
A text description of the Business Interlink Object	GetDescription(), GetDesc
The name of the Business Interlink Definition.	GetInterfaceName()
The value of one configuration parameter.	GetConfigParam(strParamName)
The list of configuration parameters for this Business Interlink Object.	GetConfigParams()
The input parameter information for this Business Interlink Object.	GetInputParams()
The input table containing the input values for this Business Interlink Object.	GetInputTable()

To retrieve:	*Use this method*
The type of this Business Interlink Object.	GetInterfaceType()
The name of this Business Interlink Object.	GetObjName()
The output parameter information for this Business Interlink Object.	GetOutputParams()
The output table into which you will insert output values for this Business Interlink Object.	GetOutputTable()

ConfigParam

See GetConfigParam, ConfigParam, GetConfigParamCount.

Description

See GetDescription, Description.

GetDescription, Description

Syntax: C++

```
PSIOString& GetDescription()
```

Syntax: Visual Basic

```
string Description
```

Syntax: Java

```
String GetDescription()
```

Class

C++: InterfaceObject

Visual Basic: IpsIntObj

Java: PSJInterfaceObject

Description

The C++ **GetDescription** method and the Visual Basic property **Description** get the description of this Business Interlink Plug-in. This description is set when you write the XML design-time plug-in or write the **GetDesc** method.

Parameters

None.

Return Value

Value	*Meaning*
C++: PSIOString Visual Basic: string Java: String	A string containing the description for this Business Interlink Plug-in.

Example

In the following C++ example, IntObj is a pointer to a Business Interlink Object.

```
PSIOString& intobj_desc = IntObj->GetDescription();
```

In the following Visual Basic example, IntObj is a pointer to a Business Interlink Object.

```
Dim intobj_desc As String
intobj_desc = IntObj.Description
```

In the following Java example, intobj is a pointer to a Business Interlink Object.

```
String intobj_desc = intobj.GetDescription();
```

In the case of where XML code is set as follows,

```
<general_info>
   <description>UPS services</description>
   <version>1</version>
</general_info>
```

the description returned would be “UPS services”.

GetInterfaceName, InterfaceName

Syntax: C++

```
PSIOString& GetInterfaceName()
```

Syntax: Visual Basic

```
string InterfaceName
```

Syntax: Java

```
String GetInterfaceName()
```

Class

C++: InterfaceObject

Visual Basic: IpsIntObj

Java: PSJInterfaceObject

Description

The C++ method **GetInterfaceName** and the Visual Basic property **InterfaceName** get the name of a Business Interlink Definition. This name is created in the PeopleSoft Application Designer.

Parameters

None.

Return Value

Value	Meaning
C++: PSIOString Visual Basic, Java: string	A string containing the Business Interlink Definition name.

Example

Within PeopleCode, you use the name of the Business Interlink Definition to instantiate a Business Interlink Object, as in the following PeopleCode:

```
&QE_RP_SRAALL_1 =
   GetBusinessInterlink(BUSINESSINTERLINK.QE_UPS_COST);
```

The following code within a Business Interlink Plug-in would fill the IntName variable with the Business Interlink Definition name of "QE_UPS_COST". IntObj is a pointer to the Business Interlink Object.

C++:

```
PSIOString IntName;
IntName = IntObj->GetInterfaceName();
```

Visual Basic:

```
Dim IntName As String
IntName = IntObj.InterfaceName
```

Java:

```
String IntName = intobj.GetInterfaceName();
```

GetConfigParam, ConfigParam, GetConfigParamCount

Syntax: C++

```
PSIOString& GetConfigParam(PSIOString&);
```

Syntax: Visual Basic

```
string ConfigParam
```

Syntax: Java

```
PSJVarInfo GetConfigParam(int index)

int GetConfigParamCount()
```

Class

C++: InterfaceObject

Visual Basic: IPsIntObj

Java: PSJInterfaceObject

Description

The method **GetConfigParam** and the property **ConfigParam**, when given the name of a configuration parameter in this Business Interlink Object, gets the value of one configuration parameter. This is used when the Interlink Object only contains one configuration parameter. For Java, the method **GetConfigParamCount** gets the number of configuration parameters.

Parameters

strParamName: Input. A PSIOString variable containing the name of the configuration parameter. This name is set either in the XML design-time plug-in or by the

Return Value

Value	*Meaning*
PSIOString&: C++ string: Visual Basic, Java	A string containing the value of the configuration parameter named in the strParamName input parameter.

Example

In the following examples, IntObj is a pointer to a Business Interlink Object.

The following C++ code retrieves the value of the configuration parameter named “ConfigParameter”.

```
const PSIOString& value = IntObj->GetConfigParam("ConfigParameter");
```

The following Visual Basic code retrieves the value of the configuration parameter named "ConfigParameter".

```
Dim ConfigName As String
ConfigName = IntObj.InterfaceName
```

The following Java code will set indexes pointing to the two configuration parameters named config1 and config2.

```
int nCountOut = 0;
nCountOut = intobj.GetConfigParamCount();
for (int i=0; i<nCountOut; i++) {
   try
   {
      PSJVarInfo varinfo = intobj.GetConfigParam(i);
      if(varinfo.strName.equals("config1"))
         int index_config1 = i;
      else if(varinfo.strName.equals("config2"))
         int index_config2 = i;
   }
   catch ( NoObjectReferenceException e)
   {
      return PSReturnStatus.FAILED;
   }
```

GetConfigParams

Syntax: C++

```
VARINFOLIST& GetConfigParams()
```

Syntax: Visual Basic

```
GetConfigParams() As IPsEnumVarInfo
```

Class

C++: InterfaceObject

Visual Basic: IPsIntObj

Description

The method **GetConfigParams** gets the information about the configuration parameters for a Business Interlink Object—name, data type, default value, required—and places it into a list of type VARINFOLIST/PsEnumVarInfo. The configuration parameters are created in the XML design-time plug-in.

Parameters

None.

Return Value

A list of type VARINFOLIST in C++, and PSEnumVarInfo in Visual Basic.

For more information about VARINFOLIST and PSEnumVarInfo, see Parameter Lists.

Example

In the following examples, IntObj is a pointer to a Business Interlink Object.

The following C++ code retrieves the configuration parameter information and stores it in a VARINFOLIST variable.

```
VARINFOLIST varList = IntObj->GetConfigParams();
```

The following Visual Basic code retrieves the configuration parameter information and stores it in a VarInfo list named objConfigParams.

```
Set objConfigParams = pIntObj.GetConfigParams
```

In C++, after you use the GetConfigParams method to get the configuration parameters, use the following methods to retrieve the information about the parameters:

- c_str if your software system uses UniCode data.
- c_astr if your software system uses ASCII data.

For example, to get the names of the configuration parameters using UniCode data, use the c_str() method and the m_strName

```
char *config_name1 = (char *)varList[0]->m_strName.c_str();
char *config_name2 = (char *)varList[1]->m_strName.c_str();
```

After you use the Visual Basic **GetConfigParams** method to get the configuration parameters, you access the parameter information, as in the following example.

```
Dim objVarInfo As VarInfo
Dim objConfigParams As PsEnumVarInfo



Set objConfigParams = pIntObj.GetConfigParams
For Each objVarInfo In objConfigParams
   If objVarInfo.Name = "config1" Then
      string Configdata1 = objVarInfo.Data
   End If
   If objVarInfo.Name = "config2" Then
      string Configdata2 = objVarInfo.Data
```

```
   End If
Next
```

GetGroup

Syntax: C++

```
CriteriaNodeList* GetGroup()
```

Class

C++: CriteriaGroup

Description

The method **GetGroup** gets a CriteriaNodeList structure. This structure will contain the grouping for the criteria rows for a query or update.

For more information about criteria rows and the CriteriaNodeList structure, see Understanding Business Interlink Object Criteria.

Parameters

None.

Return Value

CriteriaNodeList. This structure consists of pointers and indexes that form the groupings of the criteria rows for a query or update.

Example

The following code shows a call to a routine named GenerateCriteria, which has two inputs: the address of the first CriteriaNodeList structure, and a call to GetRows, which in this case, returns a pointer to the first CriteriaRowList structure. Then the GenerateCriteria routine uses the address to the first CriteriaNodeList structure to call GetGroup, getting the first CriteriaNode within the CriteriaNodeList structure. m_CriteriaGroup is a class of type CriteriaGroup that contains the first CriteriaNodeList for a Business Interlink Object.

```
GenerateCriteria(strCriteria, &IntObj->m_CriteriaGroup,
   IntObj->m_CriteriaGroup.GetRows());



void RecordDriver::GenerateCriteria(PSIOString& strCriteria,
   CriteriaGroup* Group, CriteriaRowList *rowList)
{
```

```
CriteriaNodeList *gList = Group->GetGroup();
```

GetInputDocs

Syntax: C++

```
CBIDocs& GetInputDocs(const TCHAR*)
```

Syntax: Visual Basic

```
GetInputDocs() As IPsBIDocs
```

Syntax: Java

```
PSJBIDocs GetInputDocs(String)
```

Class

C++: InterfaceObject

Visual Basic: IpsIntObj

Java: PSJInterfaceObject

Description

The **GetInputDocs** method gets the top input document at the root level for a Business Interlink Object. This is a hierarchical structure containing the values for the inputs for this Business Interlink Object.

Parameters

None. (C++ and Java pass in an empty string.)

Return Value

Value	*Meaning*
CBIDocs&, PsBIDocs, PSJBIDocs	The input document for a Business Interlink Object. It contains the values for the input parameters.

Example

In the following C++ example, the CBIDocs input document for Calculate Cost, or the root level document, is created with the **GetInputDocs** method.

```
// Get the root level (Calculate Cost) CBIDocs input document
CBIDocs docsIn = IntObj->GetInputDocs(_T(""));
```

```
docsIn.ResetCursor();

/* You can now get values (GetValue) and documents (GetNextDoc) from this
CBIDOcs input document */
```

In the following Visual Basic example, the PsBIDocs input document for Calculate Cost, or the root level document, is created with the **GetInputDocs** method.

```
' Get the root level (Calculate Cost) PsBIDocs input document
Set objInputDocs = pIntObj.GetInputDocs
objInputDocs.ResetCursor

' You can now get values (GetValue) and documents (GetNextDoc) from
' this PsBIDOcs input document */
```

In the following Java example, the PSJBIDocs input document for Calculate Cost, or the root level document, is created with the **GetInputDocs** method.

```
PSJBIDocs docsIn = intobj.GetInputDocs("");
docsIn.ResetCursor();
// You can now get values (GetValue) and documents (GetNextDoc) from
// this PSJBIDOcs input document */
// We explicitly call destroy to cleanup C++ memory behind the
// Java Doc objects which were created during Java processing.
docsIn.destroy();
```

Related Topics

Doc Class Methods: Accessing Hierarchical Input/Output Values

GetInputParams, GetInputParam, GetInputParamCount

Syntax: C++

```
VARINFOLIST& GetInputParams()
```

Syntax: Visual Basic

```
GetInputParams() As IPsEnumVarInfo
```

Syntax: Java

```
PSJVarInfo GetInputParam(int index)

int GetInputParamCount()
```

Class

C++: InterfaceObject

Visual Basic: IPsIntObj

Java: PSJInterfaceObject

Description

The method **GetInputParams** and **GetInputParam** gets the information about the input parameters for a Business Interlink Object—name, data type, default value, required—and places it into a list of type VARINFOLIST/PsEnumVarInfo. The input parameters for a transaction Business Interlink Object are created in the XML design-time plug-in; for a Business Interlink Object of type object update, object add, or object delete, they are selected from members of a class. For Java, the method **GetInputParamCount** gets the number of input parameters because the Java method GetInputParam gets one input parameter at a time.

Parameters

None for GetInputParams or GetInputParamCount.

For GetInputParam:

Parameters

index	The location of the input parameter in the parameter list. Its value can be 0 to (number of parameters - 1).

Return Value

A list of type VARINFOLIST in C++, and IPsEnumVarInfo in Visual Basic. In Java, for GetInputParam, one input parameter of type PSJVarInfo is returned, and for GetInputParamCount, an integer containing the number of parameters is returned.

For more information about VARINFOLIST and IPSEnumVarInfo, see Parameter Lists.

Example

In the following examples, IntObj is a pointer to a Business Interlink Object.

The following code retrieves the input parameter information and stores it in a VARINFOLIST named varList.

```
VARINFOLIST varList = IntObj->GetInputParams();
```

If the XML design-time plug-in for this Business Interlink Object contains the following code,

```
<transaction name="transaction1">
   <input_list>
      <input name="input1"  type="string" required="true"/>
      <input name="input2"  type="string" required="false"/>
```

```
      <input name="input3"  type="int" required="true"/>
   </input_list>
   <output_list>
      <input name="output1"  type="string"/>
      <input name="output2"  type="string"/>
      <input name="output3"  type="int"/>
   </output_list>
</transaction>
```

then the following C++ code will set indexes pointing to each input based upon the names of the input parameters.

```
if(IntObj->GetObjName() == "transaction1")
{
   int index_input1, index_input2, index_input3;
   VARINFOLIST varListInput = IntObj->GetInputParams();
   for(int i = 0; i < varListInput.size(); i++)
   {
      if(varListInput[i]->m_strName == "input1")
         index_input1 = i;
      else if(varListInput[i]->m_strName == "input2")
         index_input2 = i;
      else if(varListInput[i]->m_strName == "input3")
         index_input3 = i;
   }
}
```

In Visual Basic, the following code will set indexes pointing to each input based upon the names of the input parameters.

```
lIndex = 0
index_input1 = -1
index_input2 = -1
index_input3 = -1

Set objInputParams = pIntObj.GetInputParams
For Each objVarInfo In objInputParams
   If objVarInfo.Name = "input1" Then
      index_input1 = lIndex
   ElseIf objVarInfo.Name = "input2" Then
      index_input2 = lIndex
   ElseIf objVarInfo.Name = "input3" Then
      index_input3 = lIndex
   End If
   lIndex = lIndex + 1
Next
```

The following Java code will set indexes pointing to each input based upon the names of the input parameters.

```
if(intobj.GetObjName().equals("transaction1"))
{
   int nCountOut = 0;
   nCountOut = intobj.GetInputParamCount();
   for (int i=0; i<nCountOut; i++) {
      try
      {
         PSJVarInfo varinfo = intobj.GetInputParam(i);
         if(varinfo.strName.equals("input1"))
            int index_input1 = i;
         else if(varinfo.strName.equals("input2"))
            int index_input2 = i;
         else if(varinfo.strName.equals("input3"))
            int index_input3 = i;
      }
      catch ( NoObjectReferenceException e)
      {
         return PSReturnStatus.FAILED;
      }
   }
}
```

GetInterfaceType

Syntax: C++

```
EIOCINTERFACESUPPORTED GetInterfaceType()
```

Syntax: Visual Basic

```
string InterfaceType
```

Syntax: Java

```
int GetInterfaceType()
```

Class

C++: InterfaceObject

Visual Basic: IPsIntObj

Java: PSJInterfaceObject

Description

The **GetInterfaceType** method and the **InterfaceType** property get the type of this Business Interlink Object. Use this method to determine what type this Business Interlink Object is: transaction, object query, object add, object update, or object delete. If your Business Interlink Plug-in only supports one type of Business Interlink Object, your Business Interlink Plug-in will not need to call this method.

Parameters

None.

Return Value

Value	*Meaning*
EIOCINTERFACESUPPORTED	The type of the Business Interlink Object. This can be any of the following: IOC_TRANSACTION (transaction) IOC_OBJADD (object add) IOC_OBJQUERY (object query) IOC_OBJUPDATE (object update) IOC_OBJDELETE (object delete)

Example

The following C++ code uses the GetInterfaceType method to determine which Business Interlink Object execution method to call. IntObj is a pointer to a Business Interlink Object.

```
EIOCEXECSTATUS eRt = INTOBJ_SUCCEED;
switch (IntObj->GetInterfaceType())
{
   case IOC_TRANSACTION:
      eRt = ExecuteTransaction(IntObj, nBatchMode);
      break;
   case IOC_OBJQUERY:
      eRt = ExecuteObjectQuery(IntObj, nBatchMode);
      break;
   case IOC_OBJADD:
      eRt = ExecuteObjectAdd(IntObj, nBatchMode);
      break;
   case IOC_OBJDELETE:
      eRt = ExecuteObjectDelete(IntObj, nBatchMode);
      break;
```

```
    case IOC_OBJUPDATE:
       eRt = ExecuteObjectUpdate(IntObj, nBatchMode);
       break;
    default:
       eRt = INTOBJ_FAILED;
}
```

GetObjName, ObjName

Syntax: C++

```
PSIOString& GetObjName()
```

Syntax: Visual Basic

```
string ObjName
```

Syntax: Java

```
int GetObjName()
```

Class

C++: InterfaceObject

Visual Basic: IPsIntObj

Java: PSJInterfaceObject

Description

The C++ **GetObjName** method and the Visual Basic property **ObjName** get the name of the Business Interlink Object. The name of the Business Interlink Object is creates in the XML design-time plug-in.

Parameters

None.

Return Value

Value	*Meaning*
C++: PSIOString& Visual Basic: string Java: String	A string containing the Business Interlink Object name.

Example

The following example uses the **GetObjName** method to see if the IntObj Business Interlink Object is named “transaction1”.

```
if(IntObj->GetObjName() == _T("transaction1")
   { /* send an email message */ }
```

For example, if the XML design-time plug-in uses the following code to create a transaction Business Interlink Object,

```
<transaction name="transaction1">
   <input_list>
      <input name="input1"  type="string" required="true"/>
      <input name="input2"  type="string" required="false"/>
      <input name="input3"  type="int" required="true"/>
   </input_list>
   <output_list>
      <input name="output1"  type="string"/>
      <input name="output2"  type="string"/>
      <input name="output3"  type="int"/>
   </output_list>
</transaction>
```

then the following C++ code will check for that transaction Business Interlink Object name.

```
if(IntObj->GetObjName() == _T("transaction1")
{ /* Perform processing for transaction1 */ }
```

The following Visual Basic code will check for that transaction Business Interlink Object name.

```
If pIntObj.ObjName = "transaction1" Then
   ' Perform processing for transaction1
EndIf
```

The following Java code will check for that transaction Business Interlink Object name.

```
if (intobj.GetObjName().equals("transaction1"))
{ /* Perform processing for transaction1 */ }
```

GetOutputDocs

Syntax: C++

```
CBIDocs& GetOutputDocs(const TCHAR*)
```

Syntax: Visual Basic

```
GetOutputDocs() As IPsBIDocs
```

Syntax: Java

```
PSJBIDocs GetOutputDocs(String)
```

Class

C++: InterfaceObject

Visual Basic: IPsIntObj

Java: PSJInterfaceObject

Description

The **GetOutputDocs** method gets the top output document at the root level for a Business Interlink Object. This is a hierarchical structure that will contain the values for the outputs for this Business Interlink Object.

Parameters

None. (C++ and Java pass in an empty string.)

Return Value

Value	Meaning
CBIDocs&, IPsBIDocs, PSJBIDocs	The output document for a Business Interlink Object. It will contain the values for the output parameters.

Example

In the following C++ example, the CBIDocs Output document for Calculate Cost, or the root level document, is created with the **GetOutputDocs** method.

```
// Get the root level (Calculate Cost) CBIDocs output document
CBIDocs docsOut = IntObj->GetOutputDocs(_T(""));
docsOut.Clear();

/* Get the output document names, and the member names, using GetOutputParams
(code not shown) */

/* You can now add values (AddValue) and documents (AddNextDoc) to this
CBIDocs output document */
```

In the following Visual Basic example, the PsBIDocs output document for Calculate Cost, or the root level document, is created with the **GetOutputDocs** method.

```
' Get the root level (Calculate Cost) PsBIDocs output document
Set objOutputDocs = pIntObj.GetOutputDocs
objOutputDocs.Clear
```

```
' You can now add values (AddValue) and documents (AddNextDoc) to
' this PsBIDocs output document.
```

In the following Java example, the PSBIDocs output document for Calculate Cost, or the root level document, is created with the **GetOutputDocs** method.

```
PSJBIDocs docsOut = intobj.GetOutputDocs("");
docsOut.Clear();
// You can now add values (AddValue) and documents (AddNextDoc) to
// this PsBIDocs output document.
// We explicitly call destroy to cleanup C++ memory behind the
// Java Doc objects which were created during Java processing.
docsOut.destroy();
```

Related Topics

Doc Class Methods: Accessing Hierarchical Input/Output Values

GetOutputParams, GetOutputParam, GetOutputParamCount

Syntax: C++

```
VARINFOLIST& GetOutputParams()
```

Syntax: Visual Basic

```
GetOutputParams() As IPsEnumVarInfo
```

Syntax: Java

```
PSJVarInfo GetOutputParam(int index)

int GetOutputParamCount()
```

Class

C++: InterfaceObject

Visual Basic: IpsIntObj

Java: PSJInterfaceObject

Description

The methods **GetOutputParams** and **GetOutputParam** get the information about the output parameters for a Business Interlink Object—name, data type, default value, required—and places it into a parameter list. The output parameters for a transaction Business Interlink Object are created in the XML design-time plug-in; for a Business Interlink Object of type object query, they are selected from data members of a class. For Java, the method

GetOutputParamCount gets the number of output parameters because the Java method GetOutputParam gets one output parameter at a time.

Parameters

None for GetOutputParams or GetOutputParamCount.

For GetOutputParam:

Parameters

index	The location of the output parameter in the parameter list. Its value can be 0 to (number of parameters - 1).

Return Value

A list of type VARINFOLIST in C++, and IPsEnumVarInfo in Visual Basic. In Java, for GetOutputParam, one output parameter of type PSJVarInfo is returned, and for GetOutputParamCount, an integer containing the number of parameters is returned.

For more information about paremeter lists, see Parameter Lists.

Example

In the following examples, IntObj is a pointer to a Business Interlink Object.

The following code retrieves the input parameter information and stores it in a VARINFOLIST named varListOutput.

```
VARINFOLIST varListOutput = IntObj->GetOutputParams();
```

For example, if the XML design time plug-in uses the following code to create the output parameters,

```
<transaction name="transaction1">
   <input_list>
      <input name="input1"  type="string" required="true"/>
      <input name="input2"  type="string" required="false"/>
      <input name="input3"  type="int" required="true"/>
   </input_list>
   <output_list>
      <input name="output1"  type="string"/>
      <input name="output2"  type="string"/>
      <input name="output3"  type="int"/>
   </output_list>
</transaction>
```

then the following C++ code will set indexes pointing to each output based upon the names of the output parameters.

```
if(IntObj->GetObjName() == "transaction1")
{
   int index_output1, index_output2, index_output3;
   VARINFOLIST varListInput = IntObj->GetOutputParams();
   for(int i = 0; i < varListOutput.size(); i++)
   {
      if(varListOutput [i]->m_strName == "output1")
         index_output1 = i;
      else if(varListOutput [i]->m_strName == "output2")
         index_output2 = i;
      else if(varListOutput [i]->m_strName == "output3")
         index_output3 = i;
   }
}
```

and the following Visual Basic code will set indexes pointing to each output based upon the names of the output parameters.

```
Dim objVarInfo As VarInfo
Dim objOutputParams As PsEnumVarInfo
Dim lIndex As Long
Dim index_output1 As Long
Dim index_output2 As Long
Dim index_output3 As Long


Set objOutputParams = pIntObj.GetOutputParams
For Each objVarInfo In objOutputParams
   If objVarInfo.Name = "output1" Then
      index_output1 = lIndex
   ElseIf objVarInfo.Name = "output2" Then
      index_output2 = lIndex
   ElseIf objVarInfo.Name = "output3" Then
      index_output3 = lIndex
   End If
   lIndex = lIndex + 1
Next
```

and the following Java code will set indexes pointing to each output based upon the names of the output parameters.

```
if(intobj.GetObjName().equals("transaction1"))
{
   int nCountOut = 0;
   nCountOut = intobj.GetOutputParamCount();
   for (int i=0; i<nCountOut; i++) {
      try
```

```
        {
           PSJVarInfo varinfo = intobj.GetOutputParam(i);
           if(varinfo.strName.equal("output1"))
              int index_output1 = i;
           else if(varinfo.strName.equal("output2"))
              int index_output2 = i;
           else if(varinfo.strName.equal("output3"))
              int index_output3 = i;
        }
        catch ( NoObjectReferenceException e)
        {
           return PSReturnStatus.FAILED;
        }
     }
  }
```

GetRows

Syntax

```
CriteriaRowList* GetRows();
```

Class

C++: CriteriaGroup

Description

The method **GetRows** gets a CriteriaRowList structure. This structure will contain the criteria rows for a query or update.

For more information about criteria rows and the CriteriaRowList structure, see Understanding Business Interlink Object Criteria.

Parameters

None.

Return Value

A CriteriaRowList. The CriteriaRowList structure consists of a list of CriteriaRow structures. The CriteriaRowList and CriteriaRow structures are included with the query Business Interlink Object that is input for the ExecuteObjectQuery method. The CriteriaRow structure consists of a criteria row, and has the following form.

```
int rowNum;
PSIOString lOper;      // logical operator
```

```
PSIOString lhsExpr;    // left hand side
PSIOString rOper;      // relational operator
PSIOString rhsExpr;    // right hand side
PSIOString rhsDefault; // default value of right hand side
PSIOString dataType;   // the data type
```

Example

The following code calls a routine named GenerateCriteria, which has two inputs: the address of the first CriteriaNodeList structure, and a call to GetRows, which in this case, returns a pointer to the first CriteriaRowList structure. m_CriteriaGroup is a class of type CriteriaGroup that contains the first CriteriaNodeList for a Business Interlink Object.

```
GenerateCriteria(strCriteria, &IntObj->m_CriteriaGroup,
   IntObj->m_CriteriaGroup.GetRows());
```

InterfaceName

See GetInterfaceName.GetInterfaceName, InterfaceName.

Plug-in Class Methods: Writing a Runtime Plug-in

You write the runtime plug-in using the runtime plug-in class methods. The class for these methods is the class for your Interlink Object.

ExecuteTransaction

Syntax: C++

```
virtual EIOCEXECSTATUS ExecuteTransaction(InterfaceObject * IntObj, const int
BatchMode);
```

Syntax: Visual Basic

```
ExecuteTransaction(IntObj As PsIntObj, BatchMode As Long) As
ENUM_EIOCEXECSTATUS
```

Syntax: Java

```
int ExecuteTransaction(PSJInterfaceObject intobj)
```

Class

C++: DLLBaseDriver

Visual Basic: PsIoDriver

Java: PSJInterfaceObject

Description

The ExecuteTransaction method is the method that executes the transaction runtime plug-in. You must write the code for this method when you are executing an Interlink Object that is of type transaction.

For more information on writing the ExecuteTransaction method, see Writing the Execution Method for a Runtime Plug-In.

Parameters

IntObj	An Interlink object. This is provided by the Interlink Framework.
BatchMode	Not currently used.

Return Value

Value	*Meaning*
INTOBJ_SUCCEED	Return this if the method succeeds.
INTOBJ_FAILED	Return this if the method fails.

Example

For more information on an example of the ExecuteTransaction method, see Writing the Execution Method for a Runtime Plug-In and Examples of Business Interlink Runtime Plug-in Code.

ExecuteObjectQuery

Syntax: C++

```
virtual EIOCEXECSTATUS ExecuteObjectQuery(InterfaceObject * IntObj, const int
BatchMode);
```

Syntax: Visual Basic

```
ExecuteObjectQuery(ByVal pIntObj As PSIODRIVERLib.IntObj, ByVal BatchMode As
Long) As ENUM_EIOCEXECSTATUS
```

Class

C++: DLLBaseDriver

Visual Basic: PsIoDriver

Description

The ExecuteObjectQuery method is the method that executes the query runtime plug-in. Since ExecuteObjectQuery is a virtual method, you must write the code for this method when you are executing an Interlink Object that is of type query.

Parameters

IntObj	An Interlink object. This is provided by the Interlink Framework.
BatchMode	Not currently used.

Return Value

Value	Meaning
INTOBJ_SUCCEED	Return this if the method succeeds.
INTOBJ_FAILED	Return this if the method fails.

Example

For more information on an example of the ExecuteObjectQuery method, see Writing The Execution Method for a Runtime Plug-In (Criteria Data).

ExecuteObjectAdd

Syntax: C++

```
virtual EIOCEXECSTATUS ExecuteObjectAdd(InterfaceObject * IntObj, const int
BatchMode);
```

Syntax: Visual Basic

```
ExecuteObjectAdd(IntObj As PsIntObj, BatchMode As Long) As ENUM_EIOCEXECSTATUS
```

Class

C++: DLLBaseDriver

Visual Basic: PsIoDriver

Description

The ExecuteObjectAdd method is the method that executes the add runtime plug-in. Since ExecuteObjectAdd is a virtual method, you must write the code for this method when you are executing an Interlink Object that is of type add.

Parameters

IntObj	An Interlink object. This is provided by the Interlink Framework.
BatchMode	Not currently used.

Return Value

Value	*Meaning*
INTOBJ_SUCCEED	Return this if the method succeeds.
INTOBJ_FAILED	Return this if the method fails.

Example

For more information about coding ExecuteTransaction, which is very similar to coding ExecuteObjectAdd, see Writing the Execution Method for a Runtime Plug-In.

ExecuteObjectDelete

Syntax: C++

```
virtual EIOCEXECSTATUS ExecuteObjectDelete(InterfaceObject * IntObj, const int
BatchMode);
```

Syntax: Visual Basic

```
ExecuteObjectDelete(IntObj As PsIntObj, BatchMode As Long) As
ENUM_EIOCEXECSTATUS
```

Class

C++: DLLBaseDriver

Visual Basic: PsIoDriver

Description

The ExecuteObjectDelete method is the method that executes the delete runtime plug-in. Since ExecuteObjectDelete is a virtual method, you must write the code for this method when you are executing an Interlink Object that is of type delete.

For more information about writing ExecuteObjectDelete, which uses the Criteria methods, see Writing The Execution Method for a Runtime Plug-In (Criteria Data).

Parameters

IntObj	An Interlink object. This is provided by the Interlink Framework.
BatchMode	Not currently used.

Return Value

Value	*Meaning*
INTOBJ_SUCCEED	Return this if the method succeeds.
INTOBJ_FAILED	Return this if the method fails.

ExecuteObjectUpdate

Syntax: C++

```
virtual EIOCEXECSTATUS ExecuteObjectUpdate(InterfaceObject * IntObj, const int
BatchMode);
```

Syntax: Visual Basic

```
ExecuteObjectUpdate(IntObj As PsIntObj, BatchMode As Long) As
ENUM_EIOCEXECSTATUS
```

Class

C++: DLLBaseDriver

Visual Basic: PsIoDriver

Description

The ExecuteObjectUpdate method is the method that executes the delete runtime plug-in. Since ExecuteObjectUpdate is a virtual method, you must write the code for this method when you are executing an Interlink Object that is of type update.

For more information about writing ExecuteObjectUpdate, which uses the Criteria methods, see Writing The Execution Method for a Runtime Plug-In (Criteria Data).

Parameters

IntObj	An Interlink object. This is provided by the Interlink Framework.
BatchMode	Not currently used.

Return Value

Value	Meaning
INTOBJ_SUCCEED	Return this if the method succeeds.
INTOBJ_FAILED	Return this if the method fails.

GetVersion, PsIoDriver_GetVer

Syntax: C++

```
virtual const TCHAR * GetVersion() = 0;
```

Syntax: Visual Basic

```
PsIoDriver_GetVer() As String
```

Syntax: Java

```
String GetVersion()
```

Description

Class

C++: DLLBaseDriver

Visual Basic: PsIoDriver

Java: PSJInterfaceObject

Write this methods to return a string containing the version number of this Interface Definition. This string is used in the New Interface Definition page. Since GetVersion is a virtual method, you must write the code for this method.

For more information about these methods, see Writing the Version Methods for a Business Interlink Runtime Plug-In.

Parameters

None.

Return Value

Value	Meaning
A character string.	Contains the version number of this Interface Definition.

Example

The following C++ example sets the version number to 1.00.

```
const TCHAR * SimpleDriver::GetVersion() const
{
   return _T("1.00");
}
```

The following Visual Basic example sets the version number to 1.0.0.

```
Private Function PsIoDriver_GetVer() As String
   '
   PsIoDriver_GetVer = "1.0.0"
   '
End Function
```

The following Java example sets the version number to 2.11.00.

```
public String GetVersion(){
   return "2.11.00"; }
```

IsVersionCompatible, PsIoDriver_IsVersionCompatible

Syntax: C++

```
virtual BOOL IsVersionCompatible(TCHAR* version) = 0;
```

Syntax: Visual Basic

```
PsIoDriver_IsVersionCompatible(ByVal version As String) As Long
```

Syntax: Java

```
boolean IsVersionCompatible(String version)
```

Class

C++: DLLBaseDriver

Visual Basic: PsIoDriver

Java: PSJInterfaceObject

Description

Write this method to return true or false: true if this plug-in version number is for a version that your plug-in supports, false otherwise. The Business Interlink Framework calls this

method to see if this plug-in is compatible with other versions of this plug-in that are currently in the PeopleTools database. For example, this method tells if an older version of a plug-in is compatible with a new version. Since IsVersionCompatible is a virtual method, you must write the code for this method.

For more information and another example about IsVersionCompatible, see Writing IsVersionCompatible for Your Business Interlink Plug-in Class.

Parameters

version	the plug-in version number that is passed in from other Interface Drivers in the PeopleTools database.

Return Value

Value	*Meaning*
BOOL, Long	True if *version* is a supported type for your plug-in, False otherwise.

Example

The following C++ example returns TRUE, meaning that this plug-in is compatible with any older versions of this plug-in.

```
virtual BOOL IsVersionCompatible(const TCHAR* version) { return TRUE; }
```

The following Visual Basic example returns True, meaning that this plug-in is compatible with any older versions of this plug-in.

```
Private Function PsIoDriver_IsVersionCompatible(ByVal bstrVerion As String) As
Long
    '
    PsIoDriver_IsVersionCompatible = True
    '
End Function
```

The following Java example returns True, meaning that this plug-in is compatible with any older versions of this plug-in.

```
public boolean IsVersionCompatible() {return true;}
```

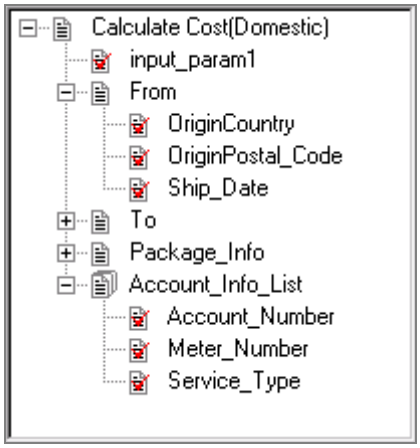
Doc Class Methods: Accessing Hierarchical Input/Output Values

You use the CBIDocs/PsCBIDocs methods to access the inputs and output values of a Business Interlink Object when the inputs and outputs are stored in a document. This is a hierarchical structure, as opposed to flat tables.

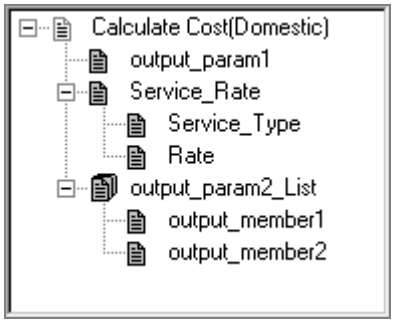
For C++, you must include the header file psbidoc.h for this class.

A Business Interlink can have hierarchical data, meaning that the inputs and outputs for a Business Interlink object are stored in the form of a CBIDocs document. Here is an example CBIDocs document: an edited version of the Calculate Cost(Domestic) transaction from the Freight Carrier plug-in.

Note: To better illustrate the CBIDocs methods, this example was created from an edited version of the Freight Carrier plug-in. The input parameter input-param1 and output parameters output_param1 and output_param2_list were added to this example, and the Account_Info input parameter was modified to be a list.



Example CBIDocs Input Document



Example CBIDocs Output Document

When you have hierarchical data, use the CBIDocs methods to add input data and get output data from the CBIDocs document.

The **GetInputDocs** method accesses a CBIDocs input document at the root level; for the example above, this is the Calculate Cost(Domestic) input document. To get another CBIDocs input document, use the **GetNextDoc** or **GetPreviousDoc** methods. Since Business Classes, when they are input parameters, are stored as documents within the CBIDocs input document, use the **GetDoc** method to get them and their members; in the example, From, To, Package_Info, and Account_Info_List are stored as documents. Since documents can also be

lists, such as Account_Info_List, use the **GetNextDoc** or **GetPreviousDoc** methods. Use the **GetValue** method to get values from either input parameters of basic type, such as input_param1 in the example, or to members of input parameters which are documents, such as OriginCountry in the From input document.

You can use the **GetCount** method to get the number of documents in an input document list, such as the number of Account_Info_List documents in the above example. To move to any document in the list without having to use **GetNextDoc** to cycle through several of them, use **MoveToDoc**.

The **GetOutputDocs** method accesses a CBIDocs output document at the root level; for the example above, this is the Calculate Cost(Domestic) output document. To add another CBIDocs output document, use the **AddNextDoc** method. Since Business Classes, when they are output parameters, are stored as documents within the CBIDocs output document, use the **AddDoc** method to add them; in the example, Service_Rate and output_param2_List are stored as documents. Since some documents can also be lists, such as output_param2_List in the example above, use the **AddNextDoc** method to add a document to the list. Use the **AddValue** method to add values to either output parameters of basic type, such as output_param1 in the example, or to members of output parameters that are documents, such as Service_Type in Service_Rate.

AddDoc

Syntax: C++

```
CBIDocs AddDoc(const TCHAR*)
```

Syntax: Visual Basic

```
AddDoc(String) As PsBIDocs
```

Syntax: Java

```
PSJBIDocs AddDoc(String)
```

Class

C++: CBIDocs

Visual Basic: PsBIDocs

Java: PSJBIDocs

Description

Adds a document to an output document. The added document is an output parameter for a Business Interlink Object that is not of simple type (such as integer or string). You must add the document to the output document before you can add values to its members with **AddValue**.

Parameters

TCHAR*, string	The name of the document that **AddDoc** adds to the output document.

Return Value

Value	*Meaning*
CBIDocs, PsBIDocs, PSJBIDocs	The document that **AddDoc** adds to the output document.

Example

In the following example, the output document for Calculate Cost, or the root level document, is created with the **GetOutputDocs** method. The Calculate Cost output parameter Service_Rate is a document, so the **AddDoc** method adds it to the output document.

C++:

```
CBIDocs docsIn = IntObj->GetInputDocs(_T(""));
docsIn.ResetCursor();

CBIDocs docsOut = IntObj->GetOutputDocs(_T(""));
docsOut.Clear();

/* Get the output document names, and the member names, using GetOutputParams
(code not shown) */

while(docsIn.GetStatus() == eNoError) {

/* Get the input values and call the external system to get values to put into
the CBIDocs output documents (code not shown) */

   if (!docsOut.Empty()) {
       docsOut.AddNextDoc();
   }

   CBIDocs docRate = docsOut.AddDoc(_T("Service_Rate"));
   PSIOString rate;
   /* Get value for serviceType (code not shown) */
   docRate.AddValue(_T("Rate"), rate.c_str());

   /* Call GetValue for output_param1, call AddDoc, GetValue, and
      GetNextDoc for output_param2_List (code not shown) */

   docsIn.GetNextDoc();
} //end while
```

Visual Basic:

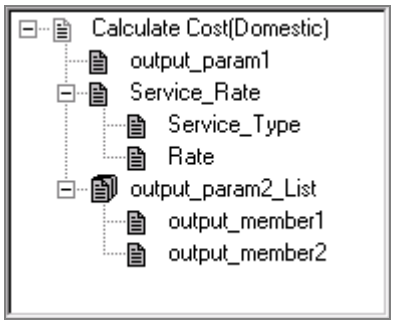
```
Dim objOutputDocs As PsBIDocs
Dim objInputDocs As PsBIDocs
Dim lResult As Long
Dim eNoError As Integer
Dim objDocRate As PsBIDocs
Dim lRate As String

Set objInputDocs = IntObj.GetInputDocs
objInputDocs.ResetCursor

Set objOutputDocs = IntObj.GetOutputDocs
objOutputDocs.Clear

' Get the output document names, and the member names, using
' GetOutputParams (code not shown)

While objInputDocs.GetStatus = eNoError

' Get the input values and call the external system to get values to
' put into the CBIDocs output documents (code not shown)

   If objOutputDocs.Empty <> 1 Then
      lResult = objOutputDocs.AddNextDoc
   End If


   Set objDocRate = objOutputDocs.AddDoc("Service_Rate")
   ' Get value for rate (code not shown)
   lResult = objDocRate.AddValue("Rate", lRate)

   ' Call GetValue for output_param1, call AddDoc, GetValue, and
   ' GetNextDoc for output_param2_List (code not shown)

   objInputDocs.GetNextDoc
Wend
```

Java:

```
PSJBIDocs docsIn = intobj.GetInputDocs("");
docsIn.ResetCursor();

PSJBIDocs docsOut = intobj.GetOutputDocs("");
docsOut.Clear();

// Get the output document names, and the member names, using GetOutputParams
```

```
// (code not shown)

// GetCount method, docsInStatus variable used instead of GetStatus
int rootCount = docsIn.GetCount("root");
int docsInStatus = -1;
if (rootCount > 0) { docsInStatus = 0;}
while(docsInStatus == 0) {

// Get the input values and call the external system to get values to put into
// the output documents (code not shown)

   docsOut.AddNextDoc();

   String serviceType = new String();
   String rate = new String();
   PSJBIDocs docRate = docsOut.AddDoc("Service_Rate");
   // Get value for serviceType (code not shown)
   docRate.AddValue("Rate", rate);
   // Call GetValue for output_param1, call AddDoc, GetValue, and
   // GetNextDoc for output_param2_List (code not shown)

   docsInStatus = docsIn.GetNextDoc();
} //end while
// Use one destroy for each GetInputDocs, GetOutputDocs, AddDoc, GetDoc
docRate.destroy();
docsOut.destroy();
docsIn.destroy();
```

The figure below shows the output document for this example. It contains three output parameters: output_param1, Service_Rate, and output_param2_List. This is a version of the Freight Carrier plug-in that was modified for this example (output_param1 and output_param2_List were added).



Example CBIDocs Output Document

AddNextDoc

Syntax: C++

Class: CBIDocs

```
int AddNextDoc();
```

Syntax: Visual Basic

```
AddNextDoc() As Long
```

Syntax: Java

```
int AddNextDoc();
```

Class

C++: CBIDocs

Visual Basic: PsBIDocs

Java: PSJBIDocs

Description

The **AddNextDoc** method adds a document to one of the following levels:

- The root level of the output document for a Business Interlink Object. This level was created with the method **GetInputDocs**.
- When a document within the output document is a list, **AddNextDoc** adds another document to the list. If you use **AddNextDoc** on a document that is not a list, **AddNextDoc** fails and returns an error.

Parameters

None.

Return Value

An integer with the following possible values:

Number	*Enum value and Meaning*
0	NoError. The method succeeded.
1	NO_DOCUMENT. The document referenced by this method does not exist.
2	LIST_OUT_RANGE. You tried to access a document in a list, but the doument list is out of range. For example, if a document list contains five documents, and then you call GetDoc/GetOutputDocs once, you can call GetNextDoc four times; the fifth time will result in this error.

Number	Enum value and Meaning
3	DOCUMENT_UNINITIALIZED. Internal error.
4	NOT_DOCUMENTTYPE. You tried to perform an operation upon a parameter that is not a document type.
5	NOT_DOCUMENTLISTTYPE. You tried to perform a GetNextDoc or AddNextDoc upon a document that is not a list.
6	NOT_LISTTYPE. You tried to perform a list operation using GetValue, AddValue, on a non-list.
7	NOT_SINGLEBASICTYPE. You tried to perform a GetValue or AddValue upon a list that does not use a single basic type: integer, float, string, time, date, datetime.
9	NO_DATA. You tried to retrieve data from a document that contained no data.
10	GENERIC_ERROR. There was an error with the transaction.

Example

In the following example, the output document for Calculate Cost, or the root level document, is created with the **GetOutputDocs** method. **AddNextDoc** will add more root level (Calculate Cost) documents at this level.

The Calculate Cost output parameter output_param2_List is a document, so it is added with the **AddDoc** method. **AddNextDoc** will add more output_param2_List documents.

C++:

```
CBIDocs docsIn = IntObj->GetInputDocs(_T(""));
docsIn.ResetCursor();

CBIDocs docsOut = IntObj->GetOutputDocs(_T(""));
docsOut.Clear();

/* Get the output document names, and the member names, using GetOutputParams
(code not shown) */

while(docsIn.GetStatus() == eNoError) {

/* Get the input values and call the external system to get values to put into
the CBIDocs output documents (code not shown) */

   if (!docsOut.Empty()) {
       docsOut.AddNextDoc();
   }

   PSIOString serviceType, rate;
   CBIDocs docRate = docsOut.AddDoc(_T("Service_Rate"));
```

```
   /* Get value for rate, serviceType (code not shown) */
   docRate.AddValue(_T("Service_Type"), serviceType.c_str());
   docRate.AddValue(_T("Rate"), rate.c_str());

// number_of_docOPList = number of docs in output_param2_List (code not shown)
   PSIOString postalcode[number_of_docOPList];
   PSIOString country[number_of_docOPList];
   CBIDocs docOPlist = docsOut.AddDoc(_T("output_param2_List"));
   /* Get values for value1 and value2 (code not shown) */
   for(int n = 0; n < number_of_docOPList; n++) {
      docOPlist.AddValue(_T("output_member1"),value1[n].c_str);
      docOPlist.AddValue(_T("output_member2"), value2[n].c_str));
      docOPlist.AddNextDoc();
   }

   docsIn.GetNextDoc();
} //end while
```

Visual Basic:

```
Dim objOutputDocs As PsBIDocs
Dim objInputDocs As PsBIDocs
Dim lResult As Long
Dim eNoError As Integer
Dim objDocRate As PsBIDocs
Dim lRate As String
Dim lServiceType As String
Dim lValue As String

Set objInputDocs = IntObj.GetInputDocs
objInputDocs.ResetCursor

Set objOutputDocs = IntObj.GetOutputDocs
objOutputDocs.Clear

' Get the output document names, and the member names, using
' GetOutputParams (code not shown)

While objInputDocs.GetStatus = eNoError

' Get the input values and call the external system to get values to
' put into the CBIDocs output documents (code not shown)

   If objOutputDocs.Empty <> 1 Then
      lResult = objOutputDocs.AddNextDoc
   End If
```

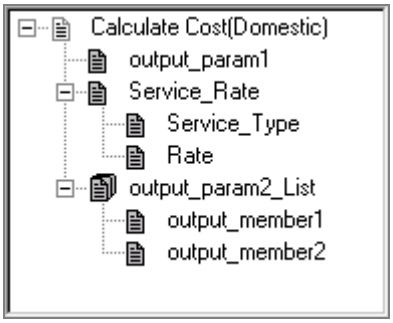
```
   Set objDocRate = objOutputDocs.AddDoc("Service_Rate")
   ' Get value for rate (code not shown)
   lResult = objDocRate.AddValue("Rate", lRate)
   lResult = objDocRate.AddValue("Service_Type", lServiceType)
   Set objDocOPlist = objOutputDocs.AddDoc("output_param2_List")

' number_of_docOPList = number of docs in output_param2_List (code not shown)
   For lIndex = 1 To number_of_docOPList
      lResult = objDocOPlist.AddValue("output_member1",
         value1[lIndex])
      lResult = objDocOPlist.AddValue("output_member2",
         value2[lIndex])
      objOutputDocs.AddNextDoc()
   Next

   objInputDocs.GetNextDoc
Wend
```

Java:

```
PSJBIDocs docsIn = intobj.GetInputDocs("");
docsIn.ResetCursor();

PSJBIDocs docsOut = intobj.GetOutputDocs("");
docsOut.Clear();

// Get the output document names, and the member names, using GetOutputParams
// (code not shown)

// GetCount method, docsInStatus variable used instead of GetStatus
int rootCount = docsIn.GetCount("root");
int docsInStatus = -1;
if (rootCount > 0) { docsInStatus = 0;}
while(docsInStatus == 0) {

// Get the input values and call the external system to get values to put into
// the output documents (code not shown)

   docsOut.AddNextDoc();

   String serviceType = new String();
   String rate = new String();
   PSJBIDocs docRate = docsOut.AddDoc("Service_Rate");
   // Get value for rate, serviceType (code not shown)
   docRate.AddValue("Service_Type", serviceType);
```

```
   docRate.AddValue("Rate", rate);

// number_of_docOPList = number of docs in output_param2_List (code not shown)
   String[] postalcode = new String[number_of_docOPList];
   String[] country = new String [number_of_docOPList];
   PSJBIDocs docOPlist = docsOut.AddDoc("output_param2_List");
   // Get values for value1 and value2 (code not shown)
   for(int n = 0; n < number_of_docOPList; n++) {
      docOPlist.AddValue("output_member1",value1[n]);
      docOPlist.AddValue("output_member2",value2[n]);
      docOPlist.AddNextDoc();
   }

   docsInStatus = docsIn.GetNextDoc();
} //end while
// Use one destroy for each GetInputDocs, GetOutputDocs, AddDoc, GetDoc
docRate.destroy();
docsOut.destroy();
docsIn.destroy();
```

The figure below shows the output document for this example. It contains three output parameters: output_param1, Service_Rate, and output_param2_List. This is a version of the Freight Carrier plug-in that was modified for this example (output_param1 and output_param2_List were added).



Example CBIDocs Output Document

AddValue, AddValueInt, AddValueFloat, AddValueDouble

Syntax: C++

```
int AddValue(const TCHAR* name, const TCHAR* value);
int AddValue(const TCHAR* name, const int value);
int AddValue(const TCHAR* name, const double value);
int AddValue(const TCHAR* name, const float value);
```

Syntax: Visual Basic

```
AddValue(name As String, value As String) As Long
AddValueInt(name As String, value As Int) As Long
AddValueFloat(name As String, value As Float) As Long
AddValueDouble(name As String, value As Double) As Long
```

Syntax: Java

```
AddValue(String name, String value);
int AddValue(String name, Integer value);
int AddValue(String name, Float value);
int AddValue(String name, Double value);
```

Class

C++: CBIDocs

Visual Basic: PsBIDocs

Java: PSBIDocs

Description

Adds a value to a member of a document within the output document for a Business Interlink Object.

Parameters

name	The name of the member of the document that is having a value added to it.
value	The value that is added.

Return Value

An integer with the following possible values:

Number	Enum value and Meaning
0	NoError. The method succeeded.
1	NO_DOCUMENT. The document referenced by this method does not exist.
2	LIST_OUT_RANGE. You tried to access a document in a list, but the doument list is out of range. For example, if a document list contains five documents, and then you call GetDoc/GetOutputDocs once, you can call GetNextDoc four times; the fifth time will result in this error.
3	DOCUMENT_UNINITIALIZED. Internal error.
4	NOT_DOCUMENTTYPE. You tried to perform an operation upon a parameter that is not a document type.
5	NOT_DOCUMENTLISTTYPE. You tried to perform a GetNextDoc or AddNextDoc upon a document that is not a list.

Number	Enum value and Meaning
6	NOT_LISTTYPE. You tried to perform a list operation using GetValue, AddValue, on a non-list.
7	NOT_SINGLEBASICTYPE. You tried to perform a GetValue or AddValue upon a list that does not use a single basic type: integer, float, string, time, date, datetime.
9	NO_DATA. You tried to retrieve data from a document that contained no data.
10	GENERIC_ERROR. There was an error with the transaction.

Example

In the following example, the output document for Calculate Cost, or the root level document, is created with the **GetOutputDocs** method. **AddValue** adds a value to the output parameter output_param1. The Calculate Cost output parameter Service_Rate is a document, so the **AddDoc** method adds it to the CBIDocs output document. Then the **AddValue** method adds values to the Service_Rate document members.

C++:

```
CBIDocs docsIn = IntObj->GetInputDocs(_T(""));
docsIn.ResetCursor();

CBIDocs docsOut = IntObj->GetOutputDocs(_T(""));
docsOut.Clear();

/* Get the output document names, and the member names, using GetOutputParams
(code not shown) */

while(docsIn.GetStatus() == eNoError) {

/* Get the input values and call the external system to get values to put into
the CBIDocs output documents (code not shown) */

   if (!docsOut.Empty()) {
       docsOut.AddNextDoc();
   }

   docsOut.AddValue(_T("output_param1"), _T("string_value"));

   CBIDocs docRate = docsOut.AddDoc(_T("Service_Rate"));
   docRate.AddValue(_T("Service_Type"), serviceType.c_str());
   docRate.AddValue(_T("Rate"), rate.c_str());

   /* Call AddDoc, AddValue, and AddNextDoc for output_param2_List
      (code not shown) */
```

```
   docsIn.GetNextDoc();
} //end while
```

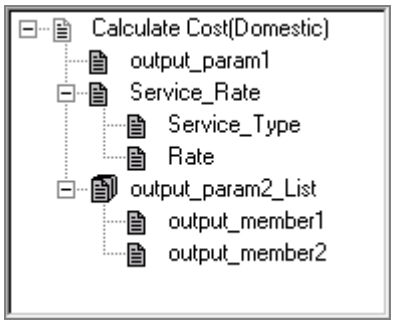
Visual Basic:

```
Dim objOutputDocs As PsBIDocs
Dim objInputDocs As PsBIDocs
Dim lResult As Long
Dim eNoError As Integer
Dim objDocRate As PsBIDocs
Dim lRate As String
Dim lServiceType As String
Dim lValue As String

Set objInputDocs = IntObj.GetInputDocs
objInputDocs.ResetCursor

Set objOutputDocs = IntObj.GetOutputDocs
objOutputDocs.Clear

' Get the output document names, and the member names, using
' GetOutputParams (code not shown)

While objInputDocs.GetStatus = eNoError

' Get the input values and call the external system to get values to
' put into the CBIDocs output documents (code not shown)

   If objOutputDocs.Empty <> 1 Then
      lResult = objOutputDocs.AddNextDoc
   End If

   lResult = objOutputDocs.AddValue("output_param1", lValue)

   Set objDocRate = objOutputDocs.AddDoc("Service_Rate")
   ' Get value for rate (code not shown)
   lResult = objDocRate.AddValue("Rate", lRate)
   lResult = objDocRate.AddValue("Service_Type", lServiceType)
   Set objDocOPlist = objOutputDocs.AddDoc("output_param2_List")

   ' Call AddDoc, AddValue, and AddNextDoc for output_param2_List
   '  (code not shown) */

   objInputDocs.GetNextDoc
Wend
```

Java:

```
PSJBIDocs docsIn = IntObj.GetInputDocs("");
docsIn.ResetCursor();

try {

   PSJBIDocs docsOut = IntObj.GetOutputDocs("");
   docsOut.Clear();

   // Get the output document names, and the member names, using
GetOutputParams
   // (code not shown)

   // GetCount method, docsInStatus variable used instead of GetStatus
   int rootCount = docsIn.GetCount("root");
   int docsInStatus = -1;
   if (rootCount > 0) { docsInStatus = 0;}
   while(docsInStatus == 0) {

   // Get the input values and call the external system to get values to put
   // into the CBIDocs output documents (code not shown)

      docsOut.AddNextDoc();

      docsOut.AddValue("output_param1", "string_value");

      PSJBIDocs docRate = docsOut.AddDoc("Service_Rate");
      docRate.AddValue("Service_Type", serviceType);
      docRate.AddValue("Rate", rate);
   }

/* Call AddDoc, AddValue, and AddNextDoc for output_param2_List
         (code not shown) */

      docsInStatus = docsIn.GetNextDoc();
   } //end while
// Use one destroy for each GetInputDocs, GetOutputDocs, AddDoc, GetDoc
docRate.destroy();
docsOut.destroy();
docsIn.destroy();
```

The figure below shows the output document for this example. It contains three output parameters: output_param1, Service_Rate, and output_param2_List. This is a version of the Freight Carrier plug-in that was modified for this example (output_param1 and output_param2_List were added).



Example CBIDocs Output Document

Clear

Syntax: C++

```
void Clear()
```

Syntax: Visual Basic

```
Clear()
```

Syntax: Java

```
Clear()
```

Class

C++: CBIDocs

Visual Basic: PsBIDocs

Java: PJSBIDocs

Description

Returns the document for a Business Interlink Object to its initial state, containing no values. The Clear method is used on the output document in order to clear it before adding documents and values to it. Once you clear it, you may need to use the **GetOutputParams** method to get the names of the output parameters.

Parameters

None.

Return Value

None.

Example

The following example uses the **Clear** method to clear docsOut, which is an output document for a Business Interlink Object.

C++:

```
CBIDocs docsOut = IntObj->GetOutputDocs(_T(""));
docsOut.Clear();
```

Visual Basic:

```
Set docsOut = IntObj.GetOutputDocs
docsOut.Clear
```

Java:

```
PSJBIDocs docsOut = intobj.GetOutputDocs("");
docsOut.Clear();
```

destroy

Syntax: Java

```
PSJBIDocs destroy()
```

Class

Java: PSJBIDocs

Description

Used only with Java. Performs memory cleanup for the GetInputDocs, GetOutputDocs, GetDoc, and AddDoc methods. Adds a document to an output document. Call destroy once for each time you call these methods. The destroy method cleans up C++ memory behind the Java Doc objects which was created during Java processing.

Parameters

None.

Return Value

None.

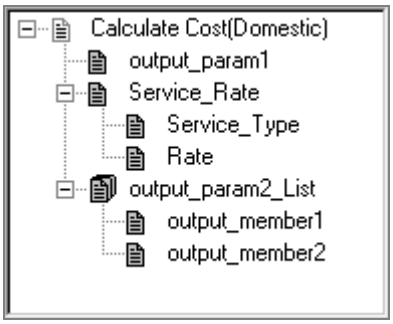
Example

In the following example, the output document for Calculate Cost, or the root level document, is created with the **GetOutputDocs** method. The Calculate Cost output parameter Service_Rate is a document, so the **AddDoc** method adds it to the output document.

Java:

```
PSJBIDocs docsIn = intobj.GetInputDocs("");
docsIn.ResetCursor();

PSJBIDocs docsOut = intobj.GetOutputDocs("");
docsOut.Clear();

// Get the output document names, and the member names, using GetOutputParams
// (code not shown)

// GetCount method, docsInStatus variable used instead of GetStatus
int rootCount = docsIn.GetCount("root");
int docsInStatus = -1;
if (rootCount > 0) { docsInStatus = 0;}
while(docsInStatus == 0) {

// Get the input values and call the external system to get values to put into
// the output documents (code not shown)

   docsOut.AddNextDoc();

   String serviceType = new String();
   String rate = new String();
   PSJBIDocs docRate = docsOut.AddDoc("Service_Rate");
   // Get value for serviceType (code not shown)
   docRate.AddValue("Rate", rate);
   // Call GetValue for output_param1, call AddDoc, GetValue, and
   // GetNextDoc for output_param2_List (code not shown)

   docsInStatus = docsIn.GetNextDoc();
} //end while
// Use one destroy for each GetInputDocs, GetOutputDocs, AddDoc, GetDoc
docRate.destroy();
docsOut.destroy();
docsIn.destroy();
```

The figure below shows the output document for this example. It contains three output parameters: output_param1, Service_Rate, and output_param2_List. This is a version of the Freight Carrier plug-in that was modified for this example (output_param1 and output_param2_List were added).



Example CBIDocs Output Document

Empty

Syntax: C++

```
bool Empty();
```

Syntax: Visual Basic

```
Empty() As Long
```

Class

C++: CBIDocs

Visual Basic: PsBIDocs

Description

Tests the document for a Business Interlink Object to see if it is empty. This is often done for the output document before calling **AddNextDoc** for the first time.

Parameters

None.

Return Value

Value	*Meaning*
0	The document is not empty (it exists).
1	The document is empty (it does not exist).

Example

The following example uses the **Empty** method to test docsOut, which is an output document for a Business Interlink Object, to see if it exists. If it does, the **AddNextDoc** method adds an output document.

C++:

```
if(!docsOut.Empty())
   docsOut.AddNextDoc();
```

Visual Basic:

```
If docsOut.Empty <> 1 Then
   lResult = docsOut.AddNextDoc
End If
```

GetCount

Syntax: C++

```
int GetCount(const TCHAR* name);
int GetCount(const PSIOString & name);
```

Syntax: Visual Basic

```
GetCount(name As String) As Long
```

Syntax: Java

```
int GetCount(String name);
```

Class

C++: CBIDocs

Visual Basic: PsBIDocs

Java: PSJBIDocs

Description

Returns the number of documents within a document list or parameter list contained within a CBIDocs input document for a Business Interlink Object.

Parameters

name	The name of the document list or parameter list.

Return Value

Value	*Meaning*
Int, Long	The number of documents in the list.

Example

In the following example, the input document for Calculate Cost, or the root level document, is created with the **GetInputDocs** method. The Calculate Cost input parameter Account_Info_List is a document list, so the **GetDoc** method gets it. The **GetCount** document gets the number of Account_Info_List documents in the list, and **GetNextDoc** gets each document from the list.

C++:

```
CBIDocs docsIn = IntObj->GetInputDocs(_T(""));
docsIn.ResetCursor();

/* Call GetValue for input_param1, GetDoc, GetValue for From, To, Package_Info
(code not shown) */

while(docsIn.GetStatus() == eNoError) {

   CBIDocs docAccountInfo = docsIn.GetDoc(_T("Account_Info_List"));
   int count = docsIn.GetCount(_T("Account_Info_List"));
   for(int i = 0; i < count; i++) {
      str1 = docAccountInfo.GetValue(_T("Account_Number"));
      str2 = docAccountInfo.GetValue(_T("Meter_Number"));
      const TCHAR *pStr =
         docAccountInfo.GetValue(_T("Service_Type"));
      docAccountInfo.GetNextDoc();
   }

/* Call the external system and put the resulting values into the CBIDocs
output documents (code not shown) */

   docsIn.GetNextDoc();
} //end while
```

Visual Basic:

```
Dim str1 As String
Dim str2 As String
Dim count As Long
Dim lIndex As Long
Dim objInputDocs As PsBIDocs
Dim objDocAccountInfo As PsBIDocs
```

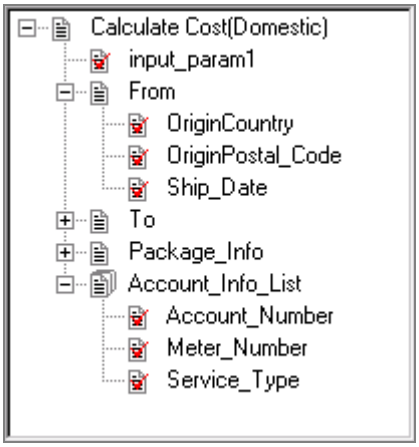
```
Set objInputDocs = IntObj.GetInputDocs
objInputDocs.ResetCursor

' Call GetValue for input_param1, GetDoc, GetValue for From, To,
' Package_Info (code not shown) */

While objInputDocs.GetStatus = eNoError

   Set objDocAccountInfo = objInputDocs.GetDoc("Account_Info_List")
   count = objInputDocs.GetCount("Account_Info_List")
   For lIndex = 1 To count
      str1 = objDocAccountInfo.GetValue("Account_Number")
      str2 = objDocAccountInfo.GetValue("Meter_Number")
      str3 = objDocAccountInfo.GetValue("Meter_Number")
      objDocAccountInfo.GetNextDoc
   Next

' Call the external system and put the resulting values into the
' output documents (code not shown)

   objInputDocs.GetNextDoc
Wend
```

Java:

```
PSJBIDocs docsIn = intobj.GetInputDocs("");
docsIn.ResetCursor();
String str1 = new String();
String str2 = new String();
String str3 = new String();

// Call GetValue for input_param1, GetDoc, GetValue for From, To, Package_Info
// (code not shown)

// GetCount method, docsInStatus variable used instead of GetStatus
int rootCount = docsIn.GetCount("root");
int docsInStatus = -1;
if (rootCount > 0) { docsInStatus = 0;}
while(docsInStatus == 0) {

   PSJBIDocs docAccountInfo = docsIn.GetDoc("Account_Info_List");
   int count = docsIn.GetCount("Account_Info_List");
   for(int i = 0; i < count; i++) {
      try
      {
```

```
            str1 = docAccountInfo.GetValue("Account_Number");
            str2 = docAccountInfo.GetValue("Meter_Number");
            str3 = docAccountInfo.GetValue("Service_Type");
            docAccountInfo.GetNextDoc();
            }
            catch ( NoObjectReferenceException e)
            {
               return PSReturnStatus.FAILED;
            }
         }

         // Call the external system and put the resulting values into the PSJBIDocs
         // output documents (code not shown) */

         docsInStatus = docsIn.GetNextDoc();
      } //end while
      // Use one destroy for each GetInputDocs, GetOutputDocs, AddDoc, GetDoc
      docAccountInfo.destroy();
      docsIn.destroy();
```

The figure below shows the input document for this example. It contains five input parameters: input_param1, From, To, Package_Info, and Account_Info_List. This is a version of the Freight Carrier plug-in that was edited for this example.



Example CBIDocs Input Document

GetDoc

Syntax: C++

```
CBIDocs GetDoc(const TCHAR* docname);
```

Syntax: Visual Basic

```
GetDoc(docname As String) As PsBIDocs
```

Syntax: Java

```
PSJBIDocs GetDoc(String docname);
```

Class

C++: CBIDocs

Visual Basic: PsBIDocs

Java: PSJBIDocs

Description

Gets a document from an input document. The document is an input parameter for a Business Interlink Object that is not of simple type (such as integer or string). You must get the document from the input document before you can get values from its members with **GetValue**.

You can call **GetDoc** using a nesting feature. This feature allows you to access deeply nested documents with one call to **GetDoc**, rather than calling **GetDoc** for each nesting. See the Example section below for an example of this.

Parameters

docname	The name of the input document that **GetDoc** gets.

Return Value

Value	Meaning
CBIDocs, PsBIDocs, PSJBIDocs	The document received from the input document.

Example

In the following example, the input document for Calculate Cost(Domestic), or the root level document, is created with the **GetInputDocs** method. The Calculate Cost input parameter From is a document, so the **GetDoc** method gets it from the input document.

C++:

```
CBIDocs docsIn = IntObj->GetInputDocs(_T(""));
docsIn.ResetCursor();

while(docsIn.GetStatus() == eNoError) {

   PSIOString str0 = docsIn.GetValue(_T("input_param1"));
```

```
   CBIDocs docFrom = docsIn.GetDoc(_T("From"));
   PSIOString strPCode = docFrom.GetValue(_T("OriginPostal_Code"));
   PSIOString strCountry = docFrom.GetValue(_T("OriginCountry"));

/* Call GetDoc, GetValue for To, Package_Info, call GetDoc, GetValue,
GetNextDoc for Account_Info_List (code not shown) */

/* Call the external system and put the resulting values into the CBIDocs
output documents (code not shown) */

   docsIn.GetNextDoc();
} //end while
```

Visual Basic:

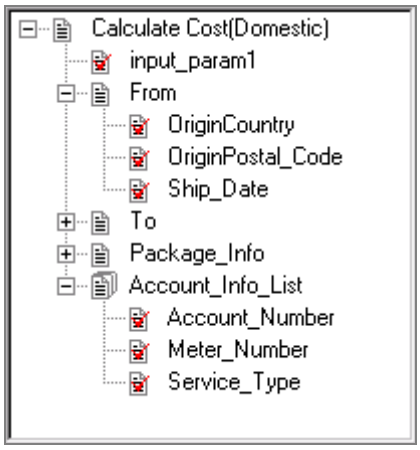
```
Dim str0 As String
Dim str1 As String
Dim str2 As String
Dim objInputDocs As PsBIDocs
Dim objDocFrom As PsBIDocs

Set objInputDocs = IntObj.GetInputDocs
objInputDocs.ResetCursor

While objInputDocs.GetStatus = eNoError

   str0 = objInputDocs.GetValue("input_param1")

   Set objDocFrom = objInputDocs.GetDoc("From")
   str1 = objDocFrom.GetValue("OriginPostal_Code")
   str2 = objDocFrom.GetValue("OriginCountry")

' Call GetDoc, GetValue for To, Package_Info, call GetDoc, GetValue,
' GetNextDoc for Account_Info_List (code not shown)

' Call the external system and put the resulting values into the
' CBIDocs output documents (code not shown)

   objInputDocs.GetNextDoc
Wend
```

Java:

```
PSJBIDocs docsIn = intobj.GetInputDocs("");
docsIn.ResetCursor();
String str0 = new String();
String strPCode = new String();
```

```
String strCountry = new String();

// GetCount method, docsInStatus variable used instead of GetStatus
int rootCount = docsIn.GetCount("root");
int docsInStatus = -1;
if (rootCount > 0) { docsInStatus = 0;}
while(docsInStatus == 0) {

   str0 = docsIn.GetValue("input_param1");

   PSJBIDocs docFrom = docsIn.GetDoc("From");
   strPCode = docFrom.GetValue("OriginPostal_Code");
   strCountry = docFrom.GetValue("OriginCountry");

/* Call GetDoc, GetValue for To, Package_Info, call GetDoc, GetValue,
GetNextDoc for Account_Info_List (code not shown) */

/* Call the external system and put the resulting values into the CBIDocs
output documents (code not shown) */

   docsInStatus = docsIn.GetNextDoc();
} //end while
// Use one destroy for each GetInputDocs, GetOutputDocs, AddDoc, GetDoc
docFrom.destroy();
docsIn.destroy();
```

The figure below shows the CBIDocs input document for this example. It contains five input parameters: input_param1, From, To, Package_Info, and Account_Info_List. This is a version of the Freight Carrier plug-in that was edited for this example.



Example CBIDocs Input Document

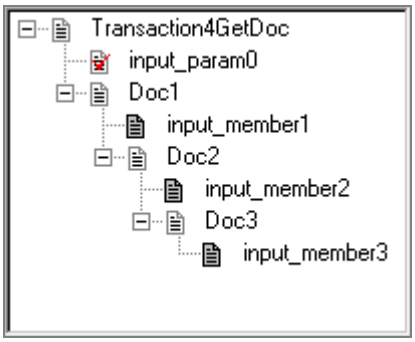
In the following C++ examples, **GetDoc** is used to access a document that is nested more deeply. If you want to access a document that is more deeply nested, you can either call **GetDoc** for each nesting, or you can call **GetDoc** once using the nesting feature.

Calling **GetDoc** with the nesting feature:

```
CBIDocs docsIn = IntObj->GetInputDocs(_T(""));
docsIn.ResetCursor();

CBIDocs Docs3 = docsIn.GetDoc(_T("Doc1.Doc2.Doc3"));
PSIOString strIn3 = Docs3.GetValue(_T("input_member3"));
```

Calling **GetDoc** without the nesting feature:

```
CBIDocs docsIn = IntObj->GetInputDocs(_T(""));
docsIn.ResetCursor();

CBIDocs Docs1 = docsIn.GetDoc(_T("Doc1"));
CBIDocs Docs2 = Docs1.GetDoc(_T("Doc2"));
CBIDocs Docs3 = Docs2.GetDoc(_T("Doc3"));
PSIOString strIn3 = Docs3.GetValue(_T("input_member3"));
```



Example BIDocs Input Document: Nested Documents

GetNextDoc

Syntax: C++

```
int GetNextDoc()
```

Syntax: Visual Basic

```
GetNextDoc() As Long
```

Syntax: Java

```
int GetNextDoc()
```

Class

C++: CBIDocs

Visual Basic: PsBIDocs

Java: PSJBIDocs

Description

The **GetNextDoc** method gets the next document from one of the following levels:

- The root level of the input document for a Business Interlink Object. This level was created with the method **GetInputDocs**.
- When a document within the input document is a list, **GetNextDoc** gets another document from the list. If you use **GetNextDoc** on a document that is not a list, **GetNextDoc** fails and returns an error.

Parameters

None.

Return Value

An integer with the following possible values:

Number	*Enum value and Meaning*
0	NoError. The method succeeded.
1	NO_DOCUMENT. The document referenced by this method does not exist.
2	LIST_OUT_RANGE. You tried to access a document in a list, but the doument list is out of range. For example, if a document list contains five documents, and then you call GetDoc/GetOutputDocs once, you can call GetNextDoc four times; the fifth time will result in this error.
3	DOCUMENT_UNINITIALIZED. Internal error.
4	NOT_DOCUMENTTYPE. You tried to perform an operation upon a parameter that is not a document type.
5	NOT_DOCUMENTLISTTYPE. You tried to perform a GetNextDoc or AddNextDoc upon a document that is not a list.
6	NOT_LISTTYPE. You tried to perform a list operation using GetValue, AddValue, on a non-list.
7	NOT_SINGLEBASICTYPE. You tried to perform a GetValue or AddValue upon a list that does not use a single basic type: integer, float, string, time, date, datetime.
9	NO_DATA. You tried to retrieve data from a document that contained no data.
10	GENERIC_ERROR. There was an error with the transaction.

Example

In this example, the input document for Calculate Cost, or the root level document, is created with the **GetInputDocs** method. The **GetNextDoc** method gets another root level document. The Calculate Cost output parameter Account_Info_List is a document, so the **GetDoc** method gets it from the input document. Account_Info_List is also a document list, so **GetNextDoc** method gets the next Account_Info_List document.

C++:

```
CBIDocs docsIn = IntObj->GetInputDocs(_T(""));
docsIn.ResetCursor();

while(docsIn.GetStatus() == eNoError) {

   const TCHAR *str0 = docsIn.GetValue(_T("input_param1"));

   /* Call GetDoc, GetValue for From, To, Package_Info (code not
      shown) */

   CBIDocs docAccountInfo = docsIn.GetDoc(_T("Account_Info_List"));
   int count = docsIn.GetCount(_T("Account_Info_List"));
   for(int i = 0; i < count; i++) {
      const TCHAR *str1 =
         docAccountInfo.GetValue(_T("Account_Number"));
      const TCHAR *str2 =
         docAccountInfo.GetValue(_T("Meter_Number"));
      const TCHAR *str3 =
         docAccountInfo.GetValue(_T("Service_Type"));
      docAccountInfo.GetNextDoc();
   }

/* Call the external system and put the resulting values into the CBIDocs
output documents (code not shown) */

   docsIn.GetNextDoc();
} //end while
```

Visual Basic:

```
Dim str1 As String
Dim str2 As String
Dim str3 As String
Dim count As Long
Dim lIndex As Long
Dim objInputDocs As PsBIDocs
Dim objDocAccountInfo As PsBIDocs

Set objInputDocs = IntObj.GetInputDocs
```

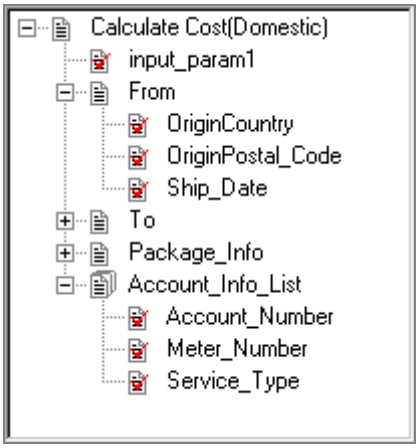
```
objInputDocs.ResetCursor

' Call GetValue for input_param1, GetDoc, GetValue for From, To,
' Package_Info (code not shown) */

While objInputDocs.GetStatus = eNoError

   Set objDocAccountInfo = objInputDocs.GetDoc("Account_Info_List")
   count = objInputDocs.GetCount("Account_Info_List")
   For lIndex = 1 To count
      str1 = objDocAccountInfo.GetValue("Account_Number")
      str2 = objDocAccountInfo.GetValue("Meter_Number")
      str3 = objDocAccountInfo.GetValue("Service_Type")
      objDocAccountInfo.GetNextDoc
   Next

' Call the external system and put the resulting values into the
' output documents (code not shown)

   objInputDocs.GetNextDoc
Wend
```

Java:

```
PSJBIDocs docsIn = intobj.GetInputDocs("");
docsIn.ResetCursor();
String str0 = new String();
String str1 = new String();
String str2 = new String();
String str3 = new String();

// GetCount method, docsInStatus variable used instead of GetStatus
int rootCount = docsIn.GetCount("root");
int docsInStatus = -1;
if (rootCount > 0) { docsInStatus = 0;}
while(docsInStatus == 0) {

   str0 = docsIn.GetValue("input_param1");

   // Call GetDoc, GetValue for From, To, Package_Info (code not
   // shown)

   PSJBIDocs docAccountInfo = docsIn.GetDoc("Account_Info_List");
   int count = docsIn.GetCount("Account_Info_List");
   for(int i = 0; i < count; i++) {
      try
```

```
        {
        str1 = docAccountInfo.GetValue("Account_Number");
        str2 = docAccountInfo.GetValue("Meter_Number");
        str3 = docAccountInfo.GetValue("Service_Type");
        docAccountInfo.GetNextDoc();
        }
        catch ( NoObjectReferenceException e)
        {
           return PSReturnStatus.FAILED;
        }
    }

    // Call the external system and put the resulting values into the PSJBIDocs
    // output documents (code not shown) */

    docsInStatus = docsIn.GetNextDoc();
} //end while
// Use one destroy for each GetInputDocs, GetOutputDocs, AddDoc, GetDoc
docAccountInfo.destroy();
docsIn.destroy();
```

The figure below shows the input document for this example. It contains five input parameters: input_param1, From, To, Package_Info, and Account_Info_List. This is a version of the Freight Carrier plug-in that was edited for this example.



Example CBIDocs Input Document

GetPreviousDoc

Syntax: C++

```
int GetPreviousDoc();
```

Syntax: Visual Basic

```
GetPreviousDoc() As Long
```

Syntax: Java

```
int GetPreviousDoc();
```

Class

C++: CBIDocs

Visual Basic: PsBIDocs

Java: PSJBIDocs

Description

The **GetPreviousDoc** method gets the previous document from one of the following levels:

- The root level of the input document for a Business Interlink Object. This level was created with the method **GetInputDocs**.
- When a document within the input document is a list, **GetPreviousDoc** gets another document from the list. If you use **GetPreviousDoc** on a document that is not a list, **GetPreviousDoc** fails and returns an error.

Parameters

None.

Return Value

An integer with the following possible values:

Number	*Enum value and Meaning*
0	NoError. The method succeeded.
1	NO_DOCUMENT. The document referenced by this method does not exist.
2	LIST_OUT_RANGE. You tried to access a document in a list, but the doument list is out of range. For example, if a document list contains five documents, and then you call GetDoc/GetOutputDocs once, you can call GetNextDoc four times; the fifth time will result in this error.
3	DOCUMENT_UNINITIALIZED. Internal error.
4	NOT_DOCUMENTTYPE. You tried to perform an operation upon a parameter that is not a document type.
5	NOT_DOCUMENTLISTTYPE. You tried to perform a GetNextDoc or AddNextDoc upon a document that is not a list.
6	NOT_LISTTYPE. You tried to perform a list operation using GetValue, AddValue, on a non-list.

Number	*Enum value and Meaning*
7	NOT_SINGLEBASICTYPE. You tried to perform a GetValue or AddValue upon a list that does not use a single basic type: integer, float, string, time, date, datetime.
9	NO_DATA. You tried to retrieve data from a document that contained no data.
10	GENERIC_ERROR. There was an error with the transaction.

Example

In the following example, the input document for Calculate Cost, or the root level document, is created with the **GetInputDocs** method. The Calculate Cost input parameter Account_Info_List is a document list, so the **GetDoc** method gets it. The **GetCount** and **MoveToDoc** methods point to the last Account_Info_List document in the list. The **GetPreviousDoc** method is used in a loop to cycle through the Account_Info_List list, starting with the last and ending with the first in the list, and get each Account_Info_List document and its values.

C++:

```
CBIDocs docsIn = IntObj->GetInputDocs(_T(""));
docsIn.ResetCursor();

while(docsIn.GetStatus() == eNoError) {

   CBIDocs docAccountInfo = docsIn.GetDoc(_T("Account_Info_List"));
   int count = docsIn.GetCount(_T("Account_Info_List"));
   docAccountInfo.MoveToDoc(count-1);
   for(int i = count; i > 0; i--) {
      const TCHAR *str1 =
         docAccountInfo.GetValue(_T("Account_Number"));
      const TCHAR *str2 =
         docAccountInfo.GetValue(_T("Meter_Number"));
      const TCHAR *str3 =
         docAccountInfo.GetValue(_T("Service_Type"));
      docAccountInfo.GetPreviousDoc();
   }

/* Call the external system and put the resulting values into the CBIDocs
output documents (code not shown) */

   docsIn.GetNextDoc();
} //end while
```

Visual Basic:

```
Dim str1 As String
Dim str2 As String
```

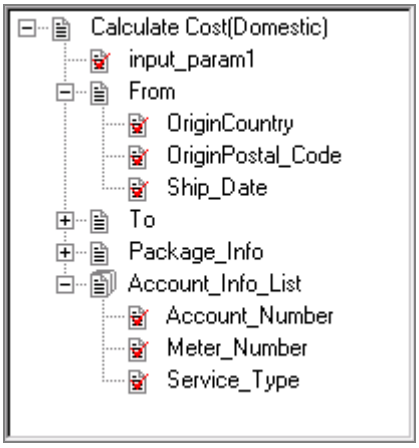
```
Dim str3 As String
Dim count As Long
Dim lIndex As Long
Dim objInputDocs As PsBIDocs
Dim objDocAccountInfo As PsBIDocs

Set objInputDocs = IntObj.GetInputDocs
objInputDocs.ResetCursor

' Call GetValue for input_param1, GetDoc, GetValue for From, To,
' Package_Info (code not shown) */

While objInputDocs.GetStatus = eNoError

   Set objDocAccountInfo = objInputDocs.GetDoc("Account_Info_List")
   count = objInputDocs.GetCount("Account_Info_List")
   objDocAccountInfo.MoveToDoc(count-1)
   For lIndex = count To 1 Step -1
      str1 = objDocAccountInfo.GetValue("Account_Number")
      str2 = objDocAccountInfo.GetValue("Meter_Number")
      str3 = objDocAccountInfo.GetValue("Service_Type")
      objDocAccountInfo.GetPreviousDoc
   Next

' Call the external system and put the resulting values into the
' output documents (code not shown)

   objInputDocs.GetNextDoc
Wend
```

Java:

```
PSJBIDocs docsIn = intobj.GetInputDocs("");
docsIn.ResetCursor();
String str1 = new String();
String str2 = new String();
String str3 = new String();

// GetCount method, docsInStatus variable used instead of GetStatus
int rootCount = docsIn.GetCount("root");
int docsInStatus = -1;
if (rootCount > 0) { docsInStatus = 0;}
while(docsInStatus == 0) {

   PSJBIDocs docAccountInfo = docsIn.GetDoc("Account_Info_List");
   int count = docsIn.GetCount("Account_Info_List");
```

```
    docAccountInfo.MoveToDoc(count-1);
    for(int i = count; i > 0; i--) {
       try
       {
       str1 = docAccountInfo.GetValue("Account_Number");
       str2 = docAccountInfo.GetValue("Meter_Number");
       str3 = docAccountInfo.GetValue("Service_Type");
       docAccountInfo.GetPreviousDoc();
       }
       catch ( NoObjectReferenceException e)
       {
          return PSReturnStatus.FAILED;
       }
    }

    // Call the external system and put the resulting values into the PSJBIDocs
    // output documents (code not shown) */

    docsInStatus = docsIn.GetNextDoc();
} //end while
// Use one destroy for each GetInputDocs, GetOutputDocs, AddDoc, GetDoc
docAccountInfo.destroy();
docsIn.destroy();
```

The figure below shows the CBIDocs input document for this example. It contains five input parameters: input_param1, From, To, Package_Info, and Account_Info_List. This is a version of the Freight Carrier plug-in that was edited for this example.



Example CBIDocs Input Document

GetStatus

Syntax: C++

```
int GetStatus()
```

Syntax: Visual Basic

```
GetStatus() As Long
```

Class

C++: CBIDocs

Visual Basic: PsBIDocs

Description

Tests the document for a Business Interlink Object. Returns the status of the document.

Parameters

None.

Return Value

An integer with the following possible values:

Number	*Enum value and Meaning*
0	NoError. The method succeeded.
1	NO_DOCUMENT. The document referenced by this method does not exist.
2	LIST_OUT_RANGE. You tried to access a document in a list, but the doument list is out of range. For example, if a document list contains five documents, and then you call GetDoc/GetOutputDocs once, you can call GetNextDoc four times; the fifth time will result in this error.
3	DOCUMENT_UNINITIALIZED. Internal error.
4	NOT_DOCUMENTTYPE. You tried to perform an operation upon a parameter that is not a document type.
5	NOT_DOCUMENTLISTTYPE. You tried to perform a GetNextDoc or AddNextDoc upon a document that is not a list.
6	NOT_LISTTYPE. You tried to perform a list operation using GetValue, AddValue, on a non-list.
7	NOT_SINGLEBASICTYPE. You tried to perform a GetValue or AddValue upon a list that does not use a single basic type: integer, float, string, time, date, datetime.
9	NO_DATA. You tried to retrieve data from a document that contained no data.
10	GENERIC_ERROR. There was an error with the transaction.

Example

The following example uses the **GetStatus** method to test docsIn, which is a CBIDocs input document for a Business Interlink Object, to see if it contains data. If it does, the **GetNextDoc** method gets a CBIDocs input document, allowing you to later get its values.

```
CBIDocs docsIn = IntObj->GetInputDocs(_T(""));
docsIn.ResetCursor();

while(docsIn.GetStatus() == eNoError)
{
   // Get input values and process them
   // Insert output values
   docsIn.GetNextDoc();
}
```

Visual Basic:

```
Dim eNoError As Long
Dim objInputDocs As PsBIDocs

Set objInputDocs = IntObj.GetInputDocs
objInputDocs.ResetCursor

While objInputDocs.GetStatus = eNoError

   ' Get input values and process them
   ' Insert output values
   objInputDocs.GetNextDoc
Wend
```

GetValue

Syntax: C++

```
const TCHAR* GetValue(const TCHAR* name);
const TCHAR* GetValue(const PSIOString & name){return
GetValue(strName.c_str());}
const int GetValue(const TCHAR* name, int&);
const int GetValue(const TCHAR* name, float&);
const int GetValue(const TCHAR* name, double&);
```

Syntax: Visual Basic

```
GetValue(name As String) As String
GetValueDouble(name As String, value As Double) As Long
```

```
GetValueFloat(name As String, value As Single) As Long
GetValueInt(name As String, value As Long) As Long
```

Syntax: Java

```
String GetValue(String name);
int GetValue(String name, Integer value);
int GetValue(String name, Float value);
int GetValue(String name, Double value);
int GetValue(String name, String time);
```

Class

C++: CBIDocs

Visual Basic: PsBIDocs

Java: PSJBIDocs

Description

Gets a value from an input parameter within a document of the input document for a Business Interlink Object.

Parameters

name	The name of the member of the document that is having its value retrieved.
value, int&, float&, double&, *time*	The retrieved value.

Return Value

Value	Meaning
value, int&, float&, double&	The value of the input parameter.
int	When the syntax is int GetValue(*name*, *value*), int is the return status of GetValue, listed in the table below.

Return Value for int

An integer with the following possible values:

Number	Enum value and Meaning
0	NoError. The method succeeded.
1	NO_DOCUMENT. The document referenced by this method does not exist.

Number	*Enum value and Meaning*
2	LIST_OUT_RANGE. You tried to access a document in a list, but the doument list is out of range. For example, if a document list contains five documents, and then you call GetDoc/GetOutputDocs once, you can call GetNextDoc four times; the fifth time will result in this error.
3	DOCUMENT_UNINITIALIZED. Internal error.
4	NOT_DOCUMENTTYPE. You tried to perform an operation upon a parameter that is not a document type.
5	NOT_DOCUMENTLISTTYPE. You tried to perform a GetNextDoc or AddNextDoc upon a document that is not a list.
6	NOT_LISTTYPE. You tried to perform a list operation using GetValue, AddValue, on a non-list.
7	NOT_SINGLEBASICTYPE. You tried to perform a GetValue or AddValue upon a list that does not use a single basic type: integer, float, string, time, date, datetime.
9	NO_DATA. You tried to retrieve data from a document that contained no data.
10	GENERIC_ERROR. There was an error with the transaction.

Example

In the following C++ example, the input document for Calculate Cost(Domestic), or the root level document, is created with the **GetInputDocs** method. The GetValue method gets the value of the input_param1 input parameter. The Calculate Cost input parameter From is a document, so the **GetDoc** method gets it from the input document. Then **GetValue** gets the values of the From members.

```
CBIDocs docsIn = IntObj->GetInputDocs(_T(""));
docsIn.ResetCursor();

while(docsIn.GetStatus() == eNoError) {

   PSIOString str0 = docsIn.GetValue(_T("input_param1"));

   CBIDocs docFrom = docsIn.GetDoc(_T("From"));
   PSIOString strPCode = docFrom.GetValue(_T("OriginPostal_Code"));
   PSIOString strCountry = docFrom.GetValue(_T("OriginCountry"));

/* Call GetDoc, GetValue for To, Package_Info, call GetDoc, GetValue,
GetNextDoc for Account_Info_List (code not shown) */

/* Call the external system and put the resulting values into the CBIDocs
output documents (code not shown) */

   docsIn.GetNextDoc();
} //end while
```

```
// Use one destroy for each GetInputDocs&GetNextDoc
docsIn.destroy();
```

In the following Visual Basic example, the input document for Calculate Cost(Domestic), or the root level document, is created with the **GetInputDocs** method. The GetValue method gets the value of the input_param1 input parameter. The Calculate Cost input parameter From is a document, so the **GetDoc** method gets it from the input document. Then **GetValue** gets the values of the From members.

```
Dim str0 As String
Dim strPCode As String
Dim strCountry As String
Dim objInputDocs As PsBIDocs
Dim objDocFrom As PsBIDocs

Set objInputDocs = pIntObj.GetInputDocs
objInputDocs.ResetCursor

While objInputDocs.GetStatus = eNoError

   str0 = objInputDocs.GetValue("input_param1")

   Set objDocFrom = objInputDocs.GetDoc("From")
   strPCode = objDocFrom.GetValue("OriginPostal_Code")
   strCountry = objDocFrom.GetValue("OriginCountry")

' Call GetDoc, GetValue for To, Package_Info, call GetDoc, GetValue,
' GetNextDoc for Account_Info_List (code not shown) */

' Call the external system and put the resulting values into the
' output documents (code not shown) */

   objInputDocs.GetNextDoc
Wend
```

In the following Java example, the input document for Calculate Cost(Domestic), or the root level document, is created with the **GetInputDocs** method. The GetValue method gets the value of the input_param1 input parameter. The Calculate Cost input parameter From is a document, so the **GetDoc** method gets it from the input document. Then **GetValue** gets the values of the From members.

```
try {

   PSJBIDocs docsIn = intobj.GetInputDocs("");
   docsIn.ResetCursor();

   // GetCount method, docsInStatus variable used instead of GetStatus
   int rootCount = docsIn.GetCount("root");
```

```
   int docsInStatus = -1;
   if (rootCount > 0) { docsInStatus = 0;}
   while(docsInStatus == 0) {

   inputstring = docsIn.GetValue("input_param1");

   PSJBIDocs docFrom = docsIn.GetDoc("From");
   String strPCode = docFrom.GetValue("OriginPostal_Code");
   String strCountry = docFrom.GetValue("OriginCountry");

/* Call GetDoc, GetValue for To, Package_Info, call GetDoc, GetValue,
GetNextDoc for Account_Info_List (code not shown) */

/* Call the external system and put the resulting values into the CBIDocs
output documents (code not shown) */

   docsInStatus = docsIn.GetNextDoc();
} //end while
// Use one destroy for each GetInputDocs, GetOutputDocs, AddDoc, GetDoc
docFrom.destroy();
docsIn.destroy();
```

MoveToDoc

Syntax: C++

```
int MoveToDoc(int iIndex);
```

Syntax: Visual Basic

```
MoveToDoc(iIndex As Long) As Long
```

Syntax: Java

```
int MoveToDoc(int iIndex);
```

Class

C++: CBIDocs

Visual Basic: PsBIDocs

Java: PSJBIDocs

Description

Within a list of documents in the output document, the **MoveToDoc** method moves to the documents given by the parameter *list_number*. After using **MoveToDoc**, the **GetValue** method gets the values of the document that is in the *list_number*+1 location in the list.

Parameters

iIndex is the number indicating the document that **MoveToDoc** moves to. After using **MoveToDoc**, the **GetValue** method gets the values of the document that is in the *list_number*+1 location in the list. For example, if *list_number* is zero, then **MoveToDoc** moves to the first document in the list.

Return Value

An integer with the following possible values:

Number	*Enum value and Meaning*
0	NoError. The method succeeded.
1	NO_DOCUMENT. The document referenced by this method does not exist.
2	LIST_OUT_RANGE. You tried to access a document in a list, but the doument list is out of range. For example, if a document list contains five documents, and then you call GetDoc/GetOutputDocs once, you can call GetNextDoc four times; the fifth time will result in this error.
3	DOCUMENT_UNINITIALIZED. Internal error.
4	NOT_DOCUMENTTYPE. You tried to perform an operation upon a parameter that is not a document type.
5	NOT_DOCUMENTLISTTYPE. You tried to perform a GetNextDoc or AddNextDoc upon a document that is not a list.
6	NOT_LISTTYPE. You tried to perform a list operation using GetValue, AddValue, on a non-list.
7	NOT_SINGLEBASICTYPE. You tried to perform a GetValue or AddValue upon a list that does not use a single basic type: integer, float, string, time, date, datetime.
9	NO_DATA. You tried to retrieve data from a document that contained no data.
10	GENERIC_ERROR. There was an error with the transaction.

Example

In the following example, the input document for Calculate Cost, or the root level document, is created with the **GetInputDocs** method. The Calculate Cost input parameter Account_Info_List is a document list, so the **GetDoc** method gets it. The **GetCount** and **MoveToDoc** methods point to the last Account_Info_List document in the list. The **GetValue** method then gets the values for the last document in the list.

C++:

```
CBIDocs docsIn = IntObj->GetInputDocs(_T(""));
docsIn.ResetCursor();

CBIDocs docAccountInfo = docsIn.GetDoc(_T("Account_Info_List"));
int count = docsIn.GetCount(_T("Account_Info_List"));
docAccountInfo.MoveToDoc(count-1);
const TCHAR *str1 = docAccountInfo.GetValue(_T("Account_Number"));
const TCHAR *str2 = docAccountInfo.GetValue(_T("Meter_Number"));
const TCHAR *str3 = docAccountInfo.GetValue(_T("Service_Type"));
```

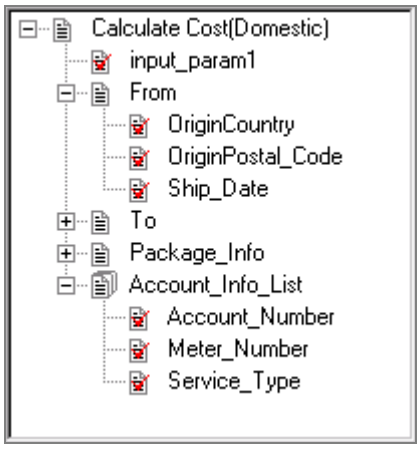
Visual Basic:

```
Dim str1 As String
Dim str2 As String
Dim str3 As String
Dim objInputDocs As PsBIDocs
Dim objDocAccountInfo As PsBIDocs

Set objInputDocs = IntObj.GetInputDocs
objInputDocs.ResetCursor


Set objDocAccountInfo = objInputDocs.GetDoc("Account_Info_List")
   count = objInputDocs.GetCount("Account_Info_List")
objDocAccountInfo.MoveToDoc(count-1)
str1 = objDocAccountInfo.GetValue("Account_Number")
str2 = objDocAccountInfo.GetValue("Meter_Number")
str3 = objDocAccountInfo.GetValue("Service_Type")
```

Java:

```
PSJBIDocs docsIn = intobj.GetInputDocs("");
docsIn.ResetCursor();

PSJBIDocs docAccountInfo = docsIn.GetDoc("Account_Info_List");
int count = docsIn.GetCount("Account_Info_List");
docAccountInfo.MoveToDoc(count-1);
String str1 = docAccountInfo.GetValue("Account_Number");
String str2 = docAccountInfo.GetValue("Meter_Number");
String str3 = docAccountInfo.GetValue("Service_Type");
// Use one destroy for each GetInputDocs, GetOutputDocs, AddDoc, GetDoc
docAccountInfo.destroy();
docsIn.destroy();
```

The figure below shows the input document for this example. It contains five input parameters: input_param1, From, To, Package_Info, and Account_Info_List. This is a version of the Freight Carrier plug-in that was edited for this example.



Example CBIDocs Input Document

ResetCursor

Syntax: C++

```
void ResetCursor();
```

Syntax: Visual Basic

```
ResetCursor()
```

Class

C++: CBIDocs

Visual Basic: PsBIDocs

Java: PSJBIDocs

Description

Resets the cursor in the CBIDocs input document for a Business Interlink Object to the top. Once you call this method, the next time you call **GetValue**, you get an input value from the first document of the CBIDocs input documents for a Business Interlink Object.

Parameters

None.

Return Value

None.

Example

The following code uses **ResetCursor** to reset the cursor in the input document to the top. IntObj is a pointer to a Business Interlink Object.

C++:

```
CBIDocs docsIn = IntObj->GetInputDocs(_T(""));
docsIn.ResetCursor();
```

Visual Basic:

```
Dim objInputDocs As PsBIDocs

Set objInputDocs = IntObj.GetInputDocs
objInputDocs.ResetCursor
```

Java:

```
PSJBIDocs docsIn = intobj.GetInputDocs("");
docsIn.ResetCursor();

// Use one destroy for each GetInputDocs, GetOutputDocs, AddDoc, GetDoc
docsIn.destroy();
```

PSIOString methods: Accessing Input/Output Parameter Information (C++ only)

You use the PSIOString methods to access the information in a PSIOString, which is how, in C++, the input and output parameter information for a Business Interlink Object is stored.

You must include the header file ioutil.h for this class.

The following table shows which PSIOString methods you use to operate on the input and output values:

Action to perform:	***Use this method***
Append a string	Append(PSIOString), operator += or +
To get a TCHAR string from a string	c_str
To find a character in a string	find
To find the length of a string in characters	length
To find a character in a string, starting from the right end of the string	rfind
	size
To get a substring from a string	substr
To perform comparisons on strings	Operators ==, !=

append

Syntax: C++

Class: PSIOString

```
CPSIOString& append(PSIOString);

CPSIOString& append(TCHAR);
```

Description

Appends a string to the end of the PSIOString string.

Parameters

TCHAR	The string that you want to append to this string.

Return Value

Value	*Meaning*
CPSIOString	The appended string.

Example

The following code uses the append method to append a comma to the end of the PSIOString named string.

```
string = string.append(_T(","));
```

c_astr

Syntax: C++

Class: PSIOString

```
char& c_astr();
```

Description

Converts a PSIOString string to an ASCII string.

The data for a Business Interlink Object is stored entirely in UniCode. If your external system uses ASCII data rather than UniCode data, its functions will require ASCII data. Use the c_astr method (instead of the c_str method) to pass the Business Interlink data to your external system functions.

Parameters

None.

Return Value

Value	*Meaning*
char	The char string, converted from the PSIOString.

Example

The following code shows the c_astr method being used to extract input parameters values in ASCII. This example is an edited version of the UPS example.

```
while(inBuf.GetNextRow(inRow))
{
   char *strFileName = "c:\\temp\\abc.tmp";

   char *strPostData = "origin=";
   strcpy(strPostData, inRow[0].c_astr());
   strcpy(strPostData, "&dest=");
   strcpy(strPostData, inRow[1].c_astr());

// Perform calculations based upon the inputs
}
```

c_str

Syntax: C++

Class: PSIOString

```
TCHAR& c_str();
```

Description

Converts a PSIOString string to a TCHAR string. Use this method to extract data from the Business Interlink Object input/output table when the input/output for your Business Interlink Plug-in is stored as UniCode.

The data for a Business Interlink Object is stored entirely in UniCode. If your software system uses UniCode data, use the c_astr method (instead of the c_astr method) to pass the Business Interlink data to your software system functions.

Parameters

None.

Return Value

Value	*Meaning*
TCHAR	The TCHAR string, converted from the PSIOString.

Example

The following code uses the c_str method to convert the PSIOString named varListOutput[i]->m_strName so that it can be used as an input for the method AddColumn. A VARINFORLIST consists of rows, and each row contains information about one input or output parameter, including PSIOString m_strName: the name of the parameter.

```
const VARINFOLIST& varListOutput = m_pIntObj->GetOutputParams();
for(int i = 0; i < varListOutput.size(); i++)
{
   outBuf.AddColumn(varListOutput[i]->m_strName.c_str());
}
```

find

Syntax: C++

Class: PSIOString

```
int find(PSIOString);

int find(TCHAR);
```

Description

Finds the location of a string within this PSIOString or TCHAR, searching from the left end of the PSIOString or TCHAR.

Parameters

PSIOString or TCHAR The string to search for.

Return Value

Value	*Meaning*
Int	The location within the PSIOString of the searched-for string. The default value is 0.

Example

The following code uses the find method to search for a period in each output parameter name, starting from the left.

```
const VARINFOLIST& varListOutput = m_pIntObj->GetOutputParams();
for(int i = 0; i < varListOutput.size(); i++)
{
   int index = varListOutput[i]->m_strName.rfind(_T('.'));
   if (index != 0)
   {
     /* Perform a process here if you found a period in the name. */
   }
}
```

length

Syntax: C++

Class: PSIOString

```
int length();
```

Description

Finds the length of this PSIOString in characters.

Parameters

None.

Return Value

Value	*Meaning*
Int	The length of this PSIOString in characters.

Example

The following code uses the find method to search for a period in each output parameter name, starting from the left.

```
const VARINFOLIST& varListOutput = m_pIntObj->GetOutputParams();
for(int i = 0; i < varListOutput.size(); i++)
{
   int index = varListOutput[i]->m_strName.rfind(_T('.'));
   if (index != 0)
   {
     /* Perform a process here if you found a period in the name. */
   }
}
```

rfind

Syntax: C++

Class: PSIOString

```
int rfind(PSIOString);

int rfind(TCHAR);
```

Description

Finds the location of a string within this PSIOString, searching from the right end of the PSIOString.

Parameters

PSIOString or TCHAR The string to search for.

Return Value

Value	*Meaning*
int	The location within the PSIOString of the searched-for string. The default value is -1.

Example

If an output parameter is a class (m_DataType = IOC_DATATYPE_OBJECT), you will retrieve the names of the class member. The name will be of the form *class.member*, where *class* is the name of the class, and *member* is the name of the class member. In that case, if you

want to use only the name of the member without the class (*member*, not *class.member*), you can use the CPSIOBuf methods rfind and substr to extract it.

The following code checks the output parameter names for the *class.member* format. It still stores the parameter name in the *class.member* format. However, if it finds a period in the parameter name, it strips off *class.* from the output parameter name, leaving only *member*. *member* is then used in the comparison to set the indexes.

```
const VARINFOLIST& varListOutput = m_pIntObj->GetOutputParams();
for(int i = 0; i < varListOutput.size(); i++)
{
   outBuf.AddColumn(varListOutput[i]->m_strName.c_str());
   PSIOString strTemp;
   int index = varListOutput[i]->m_strName.rfind(_T('.'));

   if(index == -1)
      strTemp = varListOutput[i]->m_strName;
   else
      strTemp = varListOutput[i]->m_strName.substr(index+1);

   if(strTemp ==      _T("sender"))            iSender = i;
   else if(strTemp == _T("date"))              iDate = i;
   else if(strTemp == _T("subject"))           iSubject = i;
   else if(strTemp == _T("message_id"))        iId = i;
   else if(strTemp == _T("body"))              iBody = i;
   else if(strTemp == _T("return_status_msg")) iReturnMessage = i;
   else if(strTemp == _T("return_status"))     iReturnStatus = i;
}
```

size

Syntax: C++

Class: PSIOString

```
int size();
```

Description

For PSIOString, which contains a UniCode string, returns the number of characters in the string.

For VARINFOLIST, which contains the input/output parameter information, returns the number of input/output parameters.

For IOSTRINGLIST, which contains a row of input/output parameter values, returns the number of input/output parameters.

Parameters

None.

Return Value

Value	*Meaning*
int	For PSIOString, which contains a UniCode string, returns the number of characters in the string. For VARINFORLIST, which contains the input/output parameter information, returns the number of input/output parameters. For IOSTRINGLIST, which contains a row of input/output parameter values, returns the number of input/output parameters.

Example

The following code uses the size method to set the number of output parameters in a for loop.

```
const VARINFOLIST& varListOutput = m_pIntObj->GetOutputParams();
for(int i = 0; i < varListOutput.size(); i++)
{
   int index = varListOutput[i]->m_strName.rfind(_T('.'));
   if (index != 0)
   {
     /* Perform a process here if you found a period in the name. */
   }
}
```

substr

Syntax: C++

Class: PSIOString

```
PSIOString substr(int);
```

Description

Returns a string containing the contents of this string, from the location given to the end of the string.

Parameters

int	The location in the PSIOString from which to extract the remainder of the string.

Return Value

Value	Meaning
PSIOString	A string containing the contents of this string, from the location given to the end of the string. For example, if the PSIOString name contained "class.member", then name.substr[7] would contain "member".

Example

If an output parameter is a class (m_DataType = IOC_DATATYPE_OBJECT), you will retrieve the names of the class member. The name will be of the form *class.member*, where *class* is the name of the class, and *member* is the name of the class member. In that case, if you want to use only the name of the member without the class (*member*, not *class.member*), you can use the CPSIOBuf methods rfind and substr to extract it.

The following code checks the output parameter names for the *class.member* format. It still stores the parameter name in the *class.member* format. However, if it finds a period in the parameter name, it strips off *class.* from the output parameter name, leaving only *member*. *member* is then used in the comparison to set the indexes.

```
const VARINFOLIST& varListOutput = m_pIntObj->GetOutputParams();
for(int i = 0; i < varListOutput.size(); i++)
{
   outBuf.AddColumn(varListOutput[i]->m_strName.c_str());
   PSIOString strTemp;
   int index = varListOutput[i]->m_strName.rfind(_T('.'));

   if(index == -1)
      strTemp = varListOutput[i]->m_strName;
   else
      strTemp = varListOutput[i]->m_strName.substr(index+1);

   if(strTemp ==      _T("sender"))            iSender = i;
   else if(strTemp == _T("date"))              iDate = i;
   else if(strTemp == _T("subject"))           iSubject = i;
   else if(strTemp == _T("message_id"))        iId = i;
   else if(strTemp == _T("body"))              iBody = i;
   else if(strTemp == _T("return_status_msg")) iReturnMessage = i;
   else if(strTemp == _T("return_status"))     iReturnStatus = i;
}
```

operators +, +=, ==, !=

Class: PSIOString

These operators allow the following operations to be performed upon PSIOString and TCHAR:

++	Append one PSIOString to another PSIOString.
+	Append two PSIOStrings and store the result in a third PSIOString.
++	equal comparison
!=	not equal comparison

Parameter Lists

The configuration parameters, input parameters, and output parameters from the Interlink Object all use a data structure list to store parameter information. C++ uses VARINFOLIST, and Visual Basic uses IPsEnumVarInfo. Each item in the IPsEnumVarInfo list is of type VarInfo. Java uses VarInfo to contain one parameter. Each item in the list is one parameter, and has the following structure:

C++: VARINFOLIST	*Visual Basic: IPsEnumVarInfo*	*Java: VarInfo*	*Meaning*
PSIOString m_strName	string Name	String strName()	the name of the parameter.
PSIOString m_strDefault	string Value	String strDefault()	any default value for the parameter.
DataType m_DataType	ENUM_EIOCDATA TYPE DataType	int DataType	the data type of the parameter.
EVARATTRIB UTE m_nAttribute	ENUM_VARATTRI BUTE Attribute	int nAttribute()	indicates if the parameter is required.

C++: VARINFOLIST	*Visual Basic: IPsEnumVarInfo*	*Java: VarInfo*	*Meaning*
PSIOString m_strPathName	String PathInfo	String strPathInfo()	the entire path of this parameter. For example, if this parameter is the string Country in the document From, where From is a class of type Origin and Country is an enum with values of United States and Puerto Rico, the path is object<Origin>.enum<United States,Puerto Rico>. If this parameter is the string Postal_Code in the document From, where From is a class of type Origin and Postal_Code is a string, the path is object<Origin>.string.
PSIOString m_strDisplayName	String DisplayName	String strDisplayName	This name shows in the Input Name and Output Name columns in the Application Designer for Business Interlinks.

The valid values for m_DataType or DataType are:

Data Type: C++, Visual Basic, Java	*Definition and format*
IOC_DATATYPE_INT, ENUM_IOC_DATATYPE_INT	An integer.
IOC_DATATYPE_STRING, ENUM_IOC_DATATYPE_STRING	A character string.
IOC_DATATYPE_FLOAT, ENUM_IOC_DATATYPE_FLOAT	A floating point variable.
IOC_DATATYPE_BOOLEAN, ENUM_IOC_DATATYPE_BOOLEAN	A Boolean value of TRUE or FALSE.
IOC_DATATYPE_DATE, ENUM_IOC_DATATYPE_DATE	A date. The format is YYYY-MM-DD, where YYYY is the year, MM is the month, and DD is the day.
IOC_DATATYPE_TIME, ENUMIOC_DATATYPE_TIME	A time. The format is HH:MM:SS, where HH is the hour, MM is the minutes, and SS is the seconds.
IOC_DATATYPE_DATETIME, ENUM_IOC_DATATYPE_DATETIME	A date and time. The format is YYYY-MM-DD HH:MM:SS, where YYYY is the year, MM is the month, DD is the day, HH is the hour, MM is the minutes, and SS is the seconds.
IOC_DATATYPE_ENUM, ENUM_IOC_DATATYPE_ENUM	An enumeration.

Data Type: ***C++, Visual Basic, Java***	***Definition and format***
IOC_DATATYPE_OBJECT, ENUM_IOC_DATATYPE_OBJECT	A document. This is either a **<class>** in the XML design-time plug-in, or a class defined by the GetClassByName method. Documents can contain other documents and/or any of the other data types. You use the GetDoc and AddDoc methods to access the contents of documents.
IOC_DATATYPE_PASSWORD, ENUM_IOC_DATATYPE_PASSWORD	A password. This is a character string.
Lists of the above types, such as IOC_DATATYPE_LIST_INT, ENUM_IOC_DATATYPE_LIST_INT	Each of the above types except for password can be a list.

CHAPTER 9

Writing the Design-Time Functionality Using Dynamic Catalog Methods

Once you have designed your transactions and classes by naming the transactions/classes, and naming and typing their parameters, you can introduce your transactions and classes to the Business Interlink framework. Before you can manifest your transactions as Business Interlink Definitions, you introduce them to the PeopleTools design-time environment. This is accomplished by either supplying a design-time plug-in, or by writing the design-time functionality into the run-time plug-in. The plug-in is used during the design-time to define and save Business Interlink Definitions. It can also be used at run-time to simulate the execution of the corresponding Business Interlink Objects, if you supply default output values.

For more information about writing the run-time plug-in, see Writing the Execution Method for a Runtime Plug-In.

After you have decided what services that your system should support, check the services to see if their catalogs are dynamic. A dynamic catalog means that the transactions and classes are not predefined; the input/output parameters for a transaction, or the data members for a class, are changeable in data type or number of parameters/members. (For comparison, static would mean that the transactions and classes are not changeable; a plug-in to a database would not allow the members of a class to be changed, and a new class could not be added or deleted.)

When you use dynamic catalogs, you must implement a set of methods (in C/C++, or any environment supporting COM such as Visual Basic) that are compiled into a library. In this way, your design-time plug-in can connect to the external system and dynamically query its interfaces to provide the most recent additions.

You do not need to write an XML design-time plug-in when you use dynamic catalogs, because the design-time functionality will be contained in the methods described in this chapter. If your catalogs are static, you can write the design-time functionality using either an XML design-time plug-in or the methods described in this chapter.

For more information about writing a design-time plug-in, see Writing an XML Design-Time Plug-In.

- You will perform the following actions with these methods:
- Write the categories for the transactions and classes: GetCategories to provide the categories.

- Write the configuration parameters: GetParameterList to provide the names and data types of the configuration parameters.
- Write the class catalog: GetClassByCategory to provide the class names, and GetClassByName to provide the names and data types of the data members of the classes.
- Write the transaction catalog: GetTransactionByCategory to provide the transaction names, and GetTransactionByName to provide the names and data types of the input and output parameters of the transactions.
- List any relational or logical operators (used with classes for query and update Business Interlinks):.GetCriteriaLogicalOperators for logical operators, GetCriteriaRelationalOperators for relational operators.

Writing The Design-Time Methods

This section of the chapter explains the design-time methods that you write for a Visual Basic example, and provides an example for those methods.

Write The Plug-in Description: GetDesc

Write the method GetDesc to return a short description of your plug-in. Here is a Visual Basic example.

```
Private Function PsIoDriver_GetDesc() As String
    '
    PsIoDriver_GetDesc = "Freight Carrier"
    '
End Function
```

Here is a C++ example.

```
const TCHAR * SimpleDriver::GetDesc() const
{
return _T("Simple interface driver: adds 2 #, concats 2 strs");
}
```

For more information about the GetDesc method, see GetDesc, PsIoDriver_GetDesc.

List the Supported Business Interlink Types: IsSupported

Write the method IsSupported to return True for every Business Interface Type that your Business Interlink Plug-in will support.

Here is an example of IsSupported.

```
Private Function PsIoDriver_IsSupported(ByVal eType As
PSIODRIVERLib.ENUM_EIOCINTERFACESUPPORTED) As Long
    '
    Select Case eType
        Case ENUM_IOC_TRANSACTION
            PsIoDriver_IsSupported = True
        Case ENUM_IOC_STATICCATEGORY
            PsIoDriver_IsSupported = True
        Case ENUM_IOC_ISWCHAR
            PsIoDriver_IsSupported = True
        Case Else
            PsIoDriver_IsSupported = False
    End Select
    '
End Function
```

Here is a C++ example.

```
BOOL SimpleDriver::IsSupported(EIOCINTERFACESUPPORTED option)
{
   switch(option)
   {
      case IOC_TRANSACTION:
      case IOC_STATICCATEGORY:
      return TRUE;

      case IOC_ISWCHAR:
      return  (sizeof(TCHAR) != sizeof(char));

      default:
      return FALSE;
   }
}
```

For more information about IsSupported, see IsSupported, PsIoDriver_IsSupported.

For Query and Update Plug-ins, List Relational and Logical Operators: GetCriteriaRelationalOperators, GetCriteriaLogicalOperators

If your Business Interlink is of type query or update, write the method GetCriteriaLogicalOperators to return a list of the Logical Operators.

If your Business Interlink is of type query or update, write the method GetCriteriaRelationalOperators to return a list of the Relational Operators.

Since this Business Interlink Plug-in does not use Relational or Logical Operators, the Visual Basic and C++ examples in this chapter return False for these methods.

For more information on writing GetCriteriaRelationalOperators, GetCriteriaLogicalOperators, see GetCriteriaRelationalOperators, PsIoDriver_GetCriteriaRelationalOperators and GetCriteriaLogicalOperators, PsIoDriver_GetCriteriaLogicalOperators.

List The Configuration Parameters: GetParameterList

Write the method GetParameterList to list the configuration parameters for your plug-in. Configuration parameters contain data that can be used in a variety of ways, depending upon your system. A common use is for the configuration parameters to be a set of global variables that you would use when you write the execution code for your plug-in. Another common use for configuration parameters is data that helps connect your external service to PeopleSoft.

For more information about GetParameterList, see GetParameterList, PsIoDriver_GetParameterList.

The following example shows the configuration parameter of URL being added to the configuration parameter list.

The configuration parameters are added to the plstParams list, which is a variable of type PSIODRIVERLib.IPsEnumVarInfo. Use the Add method to add the configuration parameters to the plstParams list.

```
Private Function PsIoDriver_GetParameterList(ByVal plstParams As
PSIODRIVERLib.IPsEnumVarInfo) As Long
    '
    Dim objVarInfo As VarInfo
    Set objVarInfo = plstParams.Add
    objVarInfo.Name = "URL"
    objVarInfo.DisplayName = ""
    objVarInfo.Value = "file://ups.dll"
    objVarInfo.DataType = ENUM_IOC_DATATYPE_STRING
    objVarInfo.DataDescr = "string"
    objVarInfo.Attribute = ENUM_VAR_ATTR_REQUIRED
    PsIoDriver_GetParameterList = True
    '
End Function
```

The following C++ example shows the configuration parameter of Output_File_Name being added to the configuration parameter list.

The configuration parameters are added to the VARINFOIST list, which contains parameters of type VarInfo. Use the push_back method to add the configuration parameters to the list.

```
VarInfo config(_T("Output_File_Name"), DataType(IOC_DATATYPE_STRING, _T("")),
_T(""), _T(""), _T(""), eVAR_ATTR_NONE);

BOOL  SimpleDriver::GetParameterList(VARINFOLIST& list)
{
        list.push_back(&config);
        return TRUE;
}
```

Write the Categories For the Transactions and Classes: GetCategories

Write the GetCategories method to return the list of categories for your transactions and/or classes. When you write the IsSupported method to support transactions (ENUM_IOC_TRANSACTION), return a list of your transaction categories; you must at least have the categories “All Transactions” and “Transactions”. When you write the IsSupported method to support classes (ENUM_IOC_OBJQUERY, ENUM_IOC_OBJADD, ENUM_IOC_OBJUPDATE, or ENUM_IOC_OBJDELETE, return a list of your class categories; at least, you must have the category “All Classes” and “Classes”.

The category list is of type plstCriterialNames. Use the plstCriterialNames method Add to add a category name to the list.

For more information about GetCategories, see GetCategories, PsIoDriver_GetCategories.

The following example creates the two categories All Transactions and Transactions.

```
Private Function PsIoDriver_GetCategories(ByVal eCatType As
PSIODRIVERLib.ENUM_EIOCINTERFACESUPPORTED, ByVal plstCriterialNames As
PSIODRIVERLib.IPsEnumString) As Long
    '
    Dim objPsBstr As PsBstr
    Set objPsBstr = plstCriterialNames.Add
    objPsBstr.Data = "Transactions"
    Set objPsBstr = plstCriterialNames.Add
    objPsBstr.Data = "All Transactions"
    PsIoDriver_GetCategories = True
    '
End Function
```

Write the Business Class Catalog: GetClassByCategory, GetClassByName

Write GetClassByCategory to return a list of classes under the given category name. Since this Business Interlink Plug-in only uses classes within transactions, and thus uses no class categories, this method returns False.

```
Private Function PsIoDriver_GetClassByCategory(ByVal bstrFilter As String,
ByVal bstrCategory As String, ByVal plstCategory As
PSIODRIVERLib.IPsEnumString) As Long
    '
    PsIoDriver_GetClassByCategory = False
    '
End Function
```

Write GetClassByName to return the data members for the given class name.

```
Private Function PsIoDriver_GetClassByName(ByVal bstrClassName As String,
ByVal plstClasses As PSIODRIVERLib.IPsEnumVarInfo) As Long
    '
Dim objVarInfo As VarInfo

    If bstrClassName = "Origin" Then

        Set objVarInfo = plstClasses.Add
        objVarInfo.Name = "Country"
        objVarInfo.DisplayName = "Origin Country"
        objVarInfo.Value = "United States"
        objVarInfo.DataType = ENUM_IOC_DATATYPE_ENUM
        objVarInfo.DataDescr = "United States,Puerto Rico"
        objVarInfo.Attribute = ENUM_VAR_ATTR_REQUIRED

        Set objVarInfo = plstClasses.Add
        objVarInfo.Name = "Postal_Code"
        objVarInfo.DisplayName = "Postal Code"
        objVarInfo.Value = "94588"
        objVarInfo.DataType = ENUM_IOC_DATATYPE_STRING
        objVarInfo.DataDescr = "string"
        objVarInfo.Attribute = ENUM_VAR_ATTR_REQUIRED

        PsIoDriver_GetClassByName = True


    ElseIf bstrTransName = "Destination" Then
        ' Set the data members for this class (code not shown)
        PsIoDriver_GetClassByName = True


    ElseIf bstrTransName = "Package_Information" Then
        ' Set the data members for this class (code not shown)
        PsIoDriver_GetClassByName = True


    Else
```

```
        PsIoDriver_GetClassByName = True
    End If

End Function
```

Since the C++ example uses no classes, GetClassByName returns FALSE.

For a C++ example, see GetClassByName, PsIoDriver_GetClassByName.

```
BOOL  SimpleDriver::GetClassByName(const PSIOString& strClassName, ClassDef&
classes)
{
   return FALSE;
}
```

Write the Business Transaction Catalog: GetTransactionByCategory, GetTransactionByName

Categories are used to organize transactions and classes into groups so users can select them more easily. A user designs a Business Interlink Definition in the Application Designer, and uses the Business Interlink Search page to search for the transaction or class from which they will create a Business Interlink Definition. When they select a category in that page, the transactions or classes that the user can then select are only the transactions or classes that are placed in that <category> tag in the XML design-time plug-in.

Write GetTransactionByCategory to return a list of the transactions under the given category name. There are three default categories for transactions: All Schemas, All Transactions, and Transactions.

The following example shows the transaction names “add numbers” and “concatenate strings” added to the transaction categories.

```
Private Function PsIoDriver_GetTransactionByCategory(ByVal bstrFilter As
String, ByVal bstrCriteria As String, ByVal plstTransNames As
PSIODRIVERLib.IPsEnumString) As Long
    '
    Dim objPsTransactCategory As PsBstr
    Set objPsTransactCategory = plstTransNames.Add
    objPsTransactCategory.Data = "Calculate Cost"
    Set objPsTransactCategory = plstTransNames.Add
    objPsTransactCategory.Data = "Time-in-Transit"
    Set objPsTransactCategory = plstTransNames.Add
    objPsTransactCategory.Data = "Tracking"
    PsIoDriver_GetTransactionByCategory = True
    '
End Function
```

Write GetTransactionByName to return a list of the input and output parameters for the given transaction name.

The input parameters are added to the plstInput list, which a variable of type PSIODRIVERLib.IPsEnumVarInfo. Use the Add method to add the input parameters to the plstInput list.

The output parameters are added to the plstOutput list, which a variable of type PSIODRIVERLib.IPsEnumVarInfo. Use the Add method to add the output parameters to the plstOutput list.

The following example shows the add numbers transaction using the Add method to get the input parameters of From, To, and Package_Info, and the output parameter of Service_Rate. It also shows the concatenate strings transaction getting the input parameters of string_1 and string_2, and getting the output parameter of string.

For more information about the data structure of the parameter list, see Parameter Lists. For more information about the Add method, see Add.

```
Private Function PsIoDriver_GetTransactionByName(ByVal bstrTransName As
String, ByVal plstInput As PSIODRIVERLib.IPsEnumVarInfo, ByVal plstOutput As
PSIODRIVERLib.IPsEnumVarInfo) As Long
    '
    Dim objVarInfo As VarInfo

    If bstrTransName = "Calculate Cost" Then

        Set objVarInfo = plstInput.Add
        objVarInfo.Name = "From"
        objVarInfo.DisplayName = ""
        objVarInfo.Value = ""
        objVarInfo.DataType = ENUM_IOC_DATATYPE_OBJECT
        objVarInfo.DataDescr = "Origin"
        objVarInfo.Attribute = ENUM_VAR_ATTR_REQUIRED

        Set objVarInfo = plstInput.Add
        objVarInfo.Name = "To"
        objVarInfo.DisplayName = ""
        objVarInfo.Value = ""
        objVarInfo.DataType = ENUM_IOC_DATATYPE_OBJECT
        objVarInfo.DataDescr = "Destination"
        objVarInfo.Attribute = ENUM_VAR_ATTR_REQUIRED

        Set objVarInfo = plstInput.Add
        objVarInfo.Name = "Package_Info"
        objVarInfo.DisplayName = ""
        objVarInfo.Value = ""
        objVarInfo.DataType = ENUM_IOC_DATATYPE_OBJECT
```

```
        objVarInfo.DataDescr = "Package_Information"
        objVarInfo.Attribute = ENUM_VAR_ATTR_REQUIRED

        Set objVarInfo = plstOutput.Add
        objVarInfo.Name = "Service_Rate"
        objVarInfo.DisplayName = ""
        objVarInfo.Value = ""
        objVarInfo.DataType = ENUM_IOC_DATATYPE_OBJECT
        objVarInfo.DataDescr = "Service Rate"
        objVarInfo.Attribute = ENUM_VAR_ATTR_REQUIRED

        PsIoDriver_GetTransactionByName = True

    ElseIf bstrTransName = "Tracking" Then
        '  Set the inputs and outputs for this transaction
        PsIoDriver_GetTransactionByName = True

    ElseIf bstrTransName = "Time-in-Transit" Then
        '  Set the inputs and outputs for this transaction
        PsIoDriver_GetTransactionByName = True

    Else
        PsIoDriver_GetTransactionByName = False
    End If
    '
End Function
```

The following C++ example shows the transaction names “add numbers”, “concatenate strings”, and “machine name” added to the transaction categories.

```
BOOL SimpleDriver::GetTransactionByCategory(const PSIOString& strFilter, const
PSIOString& strCriteria, IOSTRINGLIST& transNames)
{
    transNames.push_back(_T("add numbers"));
    transNames.push_back(_T("concatenate strings"));
    transNames.push_back(_T("get machine name"));
    return TRUE; // pTransNames;
}
```

The following example shows the add numbers transaction getting the input parameters of number_1 and number_2, and getting the output parameter of sum. It also shows the concatenate strings transaction getting the input parameters of string_1 and string_2, and getting the output parameter of string.

For more information about the data structure of the parameter list, see Parameter Lists. For more information about the AddInput and AddOutput methods, see AddInput and AddOutput.

```
BOOL SimpleDriver::GetTransactionByName(const PSIOString& strTransName,
TransactionDef& transDef)
{
    transDef.SetTransactionName(strTransName);

/* Add the input parameters number_1 and number_2, and the output parameter
sum, to the add numbers transaction. */
    if(strTransName == _T("add numbers"))
    {
        transDef.AddInput(VarInfo( _T("number_1"),
           DataType(IOC_DATATYPE_INT, _T("")), _T(""), _T(""), _T(""),
           eVAR_ATTR_REQUIRED));
        transDef.AddInput(VarInfo( _T("number_2"),
           DataType(IOC_DATATYPE_INT, _T("")), _T(""), _T(""), _T(""),
           eVAR_ATTR_REQUIRED));
        transDef.AddOutput(VarInfo( _T("sum"),
           DataType(IOC_DATATYPE_INT, _T("")), _T(""), _T(""), _T(""),
           eVAR_ATTR_REQUIRED));
        return TRUE;
    }
/* Add the input parameters string_1 and string_2, and the output parameter
string, to the concatenate strings transaction. */
    else if(strTransName == _T("concatenate strings"))
    {
        transDef.AddInput(VarInfo( _T("string_1"),
           DataType(IOC_DATATYPE_STRING, _T("")), _T(""), _T(""), _T(""),
           eVAR_ATTR_REQUIRED));
        transDef.AddInput(VarInfo( _T("string_2"),
           DataType(IOC_DATATYPE_STRING, _T("")), _T(""), _T(""), _T(""),
           eVAR_ATTR_REQUIRED));
        transDef.AddOutput(VarInfo( _T("string"),
           DataType(IOC_DATATYPE_STRING, _T("")), _T(""), _T(""), _T(""),
           eVAR_ATTR_REQUIRED));
        return TRUE;
    }
/* Add the output parameter machine_name to the get machine name transaction.
*/
    else if(strTransName == _T("get machine name"))
    {
        transDef.AddOutput(VarInfo( _T("machine_name"),
           DataType(IOC_DATATYPE_STRING, _T("")), _T(""), _T(""), _T(""),
           eVAR_ATTR_REQUIRED));
        return TRUE;
    }
    else
        return FALSE;
}
```

Writing a Dynamic Catalog: A Short Example

The previous example code creates a static catalog for the transactions: the number and types of the input and output parameters cannot be changed. To get a dynamic catalog, you need only write these methods to be able to change the number and type of the input and/or output parameters.

For example, you could have the IsSupported method set this Plug-in for dynamic catalogs by setting True for ENUM_IOC_DYNAMICCATEGORY instead of for ENUM_IOC_STATICCATEGORY.

```
Private Function PsIoDriver_IsSupported(ByVal eType As
PSIODRIVERLib.ENUM_EIOCINTERFACESUPPORTED) As Long
    '
    Select Case eType
        Case ENUM_IOC_TRANSACTION
            PsIoDriver_IsSupported = True
        Case ENUM_IOC_DYNAMICCATEGORY
            PsIoDriver_IsSupported = True
        Case ENUM_IOC_ISWCHAR
            PsIoDriver_IsSupported = True
        Case Else
            PsIoDriver_IsSupported = False
    End Select
    '
End Function
```

Then write the transaction to be able to add additional input or output parameters, depending upon a condition.

The following example shows GetTransactionByName coded for a transaction that adds two or three input parameters.

Note: When you write your execute method, ExecuteTransaction, make sure it can detect and extract all the input parameters.

```
Private Function PsIoDriver_GetTransactionByName(ByVal bstrTransName As
String, ByVal plstInput As PSIODRIVERLib.IPsEnumVarInfo, ByVal plstOutput As
PSIODRIVERLib.IPsEnumVarInfo) As Long
    '
    Dim objVarInfo As VarInfo
    If bstrTransName = "add numbers" Then

        Set objVarInfo = plstInput.Add
        objVarInfo.Name = "number_1"
        objVarInfo.DisplayName = ""
        objVarInfo.Value = ""
        objVarInfo.DataType = ENUM_IOC_DATATYPE_INT
```

```
        objVarInfo.DataDescr = "int"
        objVarInfo.Attribute = ENUM_VAR_ATTR_REQUIRED
'
        Set objVarInfo = plstInput.Add
        objVarInfo.Name = "number_2"
        objVarInfo.DisplayName = ""
        objVarInfo.Value = ""
        objVarInfo.DataType = ENUM_IOC_DATATYPE_INT
        objVarInfo.DataDescr = "int"
        objVarInfo.Attribute = ENUM_VAR_ATTR_REQUIRED
'
' Add the third input parameter only if a condition is true
        If (condition) Then
            Set objVarInfo = plstInput.Add
            objVarInfo.Name = "number_3"
            objVarInfo.DisplayName = ""
            objVarInfo.Value = ""
            objVarInfo.DataType = ENUM_IOC_DATATYPE_INT
            objVarInfo.DataDescr = "int"
            objVarInfo.Attribute = ENUM_VAR_ATTR_REQUIRED
        Endif
'
        Set objVarInfo = plstOutput.Add
        objVarInfo.Name = "sum"
        objVarInfo.DisplayName = "Sum"
        objVarInfo.Value = ""
        objVarInfo.DataType = ENUM_IOC_DATATYPE_INT
        objVarInfo.DataDescr = "int"
        objVarInfo.Attribute = ENUM_VAR_ATTR_REQUIRED

        PsIoDriver_GetTransactionByName = True

    Else
        PsIoDriver_GetTransactionByName = True
    End If
    '
End Function
```

Example of Design Methods (Visual Basic)

Here is a sample set of methods for a design-time plug-in. It has two transactions to add numbers and concatenate strings. Following this sample is a

- The IsSupported methods shows that the Business Interlink Plug-in supports the transaction Interlink type.

- The GetParameterList method provides a string type named Output_File_Name to allow the Business Interlink to output to a file.
- The GetTransactionByCategory method provides the transaction names of add numbers and concatenate strings. The GetTransactionByName provides, for the add numbers transaction, the integer type input parameters number_1 and number_2 and the integer type output parameter sum; and it provides for the concatenate strings transaction the string type input parameters string_1 and string_2 and the string type output parameter string.

```
' GetCategories provides the categories.
Private Function PsIoDriver_GetCategories(ByVal eCatType As
PSIODRIVERLib.ENUM_EIOCINTERFACESUPPORTED, ByVal plstCriterialNames As
PSIODRIVERLib.IPsEnumString) As Long
    '
    Dim objPsBstr As PsBstr
    Set objPsBstr = plstCriterialNames.Add
    objPsBstr.Data = "Transactions"
    Set objPsBstr = plstCriterialNames.Add
    objPsBstr.Data = "All Transactions"
    PsIoDriver_GetCategories = True
    '
End Function

' GetClassByCategory is set to False because this plug-in uses no
' classes except as transaction inputs and outputs.
Private Function PsIoDriver_GetClassByCategory(ByVal bstrFilter As String,
ByVal bstrCategory As String, ByVal plstCategory As
PSIODRIVERLib.IPsEnumString) As Long
    '
    PsIoDriver_GetClassByCategory = False
    '
End Function

' GetClassByName is set to False because this plug-in uses no
' classes.
Private Function PsIoDriver_GetClassByName(ByVal bstrClassName As String,
ByVal plstClasses As PSIODRIVERLib.IPsEnumVarInfo) As Long
    '
Dim objVarInfo As VarInfo

    If bstrClassName = "Origin" Then

        Set objVarInfo = plstClasses.Add
        objVarInfo.Name = "Country"
        objVarInfo.DisplayName = "Origin Country"
        objVarInfo.Value = "United States"
        objVarInfo.DataType = ENUM_IOC_DATATYPE_ENUM
```

```
            objVarInfo.DataDescr = "United States,Puerto Rico"
            objVarInfo.Attribute = ENUM_VAR_ATTR_REQUIRED

            Set objVarInfo = plstClasses.Add
            objVarInfo.Name = "Postal_Code"
            objVarInfo.DisplayName = "Postal Code"
            objVarInfo.Value = "94588"
            objVarInfo.DataType = ENUM_IOC_DATATYPE_STRING
            objVarInfo.DataDescr = "string"
            objVarInfo.Attribute = ENUM_VAR_ATTR_REQUIRED

            PsIoDriver_GetClassByName = True


        ElseIf bstrTransName = "Destination" Then
            Set objVarInfo = plstClasses.Add
            objVarInfo.Name = "Country"
            objVarInfo.DisplayName = "Destination Country"
            objVarInfo.Value = "United States"
            objVarInfo.DataType = ENUM_IOC_DATATYPE_ENUM
            objVarInfo.DataDescr =
"Argentina,Australia,Aruba,Brazil,Bosnia,Canada,China,Costa
Rica,Finland,France,Germany,Greece,Hong
Kong,Iran,Italy,Israel,Japan,Korea,Mexico,Russia,Spain,Taiwan,United
Kingdom,United States,Zambia"
            objVarInfo.Attribute = ENUM_VAR_ATTR_REQUIRED

            Set objVarInfo = plstClasses.Add
            objVarInfo.Name = "Postal_Code"
            objVarInfo.DisplayName = "Postal Code"
            objVarInfo.Value = "10200"
            objVarInfo.DataType = ENUM_IOC_DATATYPE_STRING
            objVarInfo.DataDescr = "string"
            objVarInfo.Attribute = ENUM_VAR_ATTR_REQUIRED

            Set objVarInfo = plstClasses.Add
            objVarInfo.Name = "Address_Type"
            objVarInfo.DisplayName = "Address Type"
            objVarInfo.Value = "Commercial"
            objVarInfo.DataType = ENUM_IOC_DATATYPE_ENUM
            objVarInfo.DataDescr = "Commercial,Residential"
            objVarInfo.Attribute = ENUM_VAR_ATTR_REQUIRED

            PsIoDriver_GetClassByName = True


        ElseIf bstrTransName = "Package_Information" Then
```

```
        Set objVarInfo = plstClasses.Add
        objVarInfo.Name = "Drop_off_Pickup"
        objVarInfo.DisplayName = "Drop-off Pickup"
        objVarInfo.Value = "One Time Pickup"
        objVarInfo.DataType = ENUM_IOC_DATATYPE_ENUM
        objVarInfo.DataDescr = "Regular Daily Pickup,On Call Air,One Time
Pickup,Letter Center,Customer Counter"
        objVarInfo.Attribute = ENUM_VAR_ATTR_REQUIRED

        Set objVarInfo = plstClasses.Add
        objVarInfo.Name = "Packaging"
        objVarInfo.DisplayName = "Packaging"
        objVarInfo.Value = "UPS Express Box"
        objVarInfo.DataType = ENUM_IOC_DATATYPE_ENUM
        objVarInfo.DataDescr = "Your Packaging,UPS Letter Envelop,UPS Tube,UPS
Express Box,UPS Worldwide 25KG Box,UPS Worldwide 10KG Box"
        objVarInfo.Attribute = ENUM_VAR_ATTR_REQUIRED

        Set objVarInfo = plstClasses.Add
        objVarInfo.Name = "Weight"
        objVarInfo.DisplayName = "Weight"
        objVarInfo.Value = "1"
        objVarInfo.DataType = ENUM_IOC_DATATYPE_INT
        objVarInfo.DataDescr = "int"
        objVarInfo.Attribute = ENUM_VAR_ATTR_REQUIRED

        Set objVarInfo = plstClasses.Add
        objVarInfo.Name = "Length"
        objVarInfo.DisplayName = "Length"
        objVarInfo.Value = "4"
        objVarInfo.DataType = ENUM_IOC_DATATYPE_INT
        objVarInfo.DataDescr = "int"
        objVarInfo.Attribute = ENUM_VAR_ATTR_REQUIRED

        Set objVarInfo = plstClasses.Add
        objVarInfo.Name = "Width"
        objVarInfo.DisplayName = "Length"
        objVarInfo.Value = "4"
        objVarInfo.DataType = ENUM_IOC_DATATYPE_INT
        objVarInfo.DataDescr = "int"
        objVarInfo.Attribute = ENUM_VAR_ATTR_REQUIRED

        Set objVarInfo = plstClasses.Add
        objVarInfo.Name = "Height"
        objVarInfo.DisplayName = "Length"
        objVarInfo.Value = "4"
```

```
            objVarInfo.DataType = ENUM_IOC_DATATYPE_INT
            objVarInfo.DataDescr = "int"
            objVarInfo.Attribute = ENUM_VAR_ATTR_REQUIRED

            PsIoDriver_GetClassByName = True


        ElseIf bstrTransName = "Service_Rate" Then
            Set objVarInfo = plstClasses.Add
            objVarInfo.Name = "Service_Type"
            objVarInfo.DisplayName = "Service Type"
            objVarInfo.Value = "UPS Next Day Air Early AM"
            objVarInfo.DataType = ENUM_IOC_DATATYPE_STRING
            objVarInfo.DataDescr = "string"
            objVarInfo.Attribute = ENUM_VAR_ATTR_REQUIRED

            Set objVarInfo = plstClasses.Add
            objVarInfo.Name = "Guaranteed_By"
            objVarInfo.DisplayName = "Guaranteed By"
            objVarInfo.Value = "8:00 AM Next Day"
            objVarInfo.DataType = ENUM_IOC_DATATYPE_STRING
            objVarInfo.DataDescr = "string"
            objVarInfo.Attribute = ENUM_VAR_ATTR_REQUIRED

            Set objVarInfo = plstClasses.Add
            objVarInfo.Name = "Rate"
            objVarInfo.DisplayName = "Rate"
            objVarInfo.Value = "50:00"
            objVarInfo.DataType = ENUM_IOC_DATATYPE_STRING
            objVarInfo.DataDescr = "string"
            objVarInfo.Attribute = ENUM_VAR_ATTR_REQUIRED

            PsIoDriver_GetClassByName = True


        Else
            PsIoDriver_GetClassByName = True
        End If


    End Function

    ' GetCriteriaLogicalOperators and GetCriteriaRelationalOperators are
    ' set to False because this plug-in is not a query or update type.
    Private Function PsIoDriver_GetCriteriaLogicalOperators(ByVal plstOperators As
    PSIODRIVERLib.IPsEnumString) As Long
```

```
    '
    PsIoDriver_GetCriteriaLogicalOperators = False
    '
End Function
Private Function PsIoDriver_GetCriteriaRelationalOperators(ByVal bstrType As
String, ByVal plstOperators As PSIODRIVERLib.IPsEnumString) As Long
    '
    PsIoDriver_GetCriteriaRelationalOperators = False
    '
End Function

' GetDesc provides a short text description of this plug-in.
Private Function PsIoDriver_GetDesc() As String
    '
    PsIoDriver_GetDesc = "Freight Carrier"
    '
End Function

' GetLastErrorMessage provides a blank error message.
Private Sub PsIoDriver_GetLastErrorMessage(ByVal pDrvMsg As
PSIODRIVERLib.IPsDriverMessage)
    '
    pDrvMsg.Message = ""
    pDrvMsg.MessageGroupId = 0
    pDrvMsg.MessageId = 0
    '
End Sub

' Provides a configuration parameter named Output_File_Name.
Private Function PsIoDriver_GetParameterList(ByVal plstParams As
PSIODRIVERLib.IPsEnumVarInfo) As Long
    '
    Dim objVarInfo As VarInfo
    Set objVarInfo = plstParams.Add
    objVarInfo.Name = "URL"
    objVarInfo.DisplayName = ""
    objVarInfo.Value = "file://ups.dll"
    objVarInfo.DataType = ENUM_IOC_DATATYPE_STRING
    objVarInfo.DataDescr = "string"
    objVarInfo.Attribute = ENUM_VAR_ATTR_REQUIRED
    PsIoDriver_GetParameterList = True
    '
End Function

' Provides the transaction names
```

```
Private Function PsIoDriver_GetTransactionByCategory(ByVal bstrFilter As
String, ByVal bstrCriteria As String, ByVal plstTransNames As
PSIODRIVERLib.IPsEnumString) As Long
    '
    Dim objPsTransactCategory As PsBstr
    Set objPsTransactCategory = plstTransNames.Add
    objPsTransactCategory.Data = "Calculate Cost"
    Set objPsTransactCategory = plstTransNames.Add
    objPsTransactCategory.Data = "Time-in-Transit"
    Set objPsTransactCategory = plstTransNames.Add
    objPsTransactCategory.Data = "Tracking"
    PsIoDriver_GetTransactionByCategory = True
    '
End Function


' Provides the input and output parameters for the transactions.
Private Function PsIoDriver_GetTransactionByName(ByVal bstrTransName As
String, ByVal plstInput As PSIODRIVERLib.IPsEnumVarInfo, ByVal plstOutput As
PSIODRIVERLib.IPsEnumVarInfo) As Long
    '
    Dim objVarInfo As VarInfo

    If bstrTransName = "Calculate Cost" Then

        Set objVarInfo = plstInput.Add
        objVarInfo.Name = "From"
        objVarInfo.DisplayName = ""
        objVarInfo.Value = ""
        objVarInfo.DataType = ENUM_IOC_DATATYPE_OBJECT
        objVarInfo.DataDescr = "Origin"
        objVarInfo.Attribute = ENUM_VAR_ATTR_REQUIRED

        Set objVarInfo = plstInput.Add
        objVarInfo.Name = "To"
        objVarInfo.DisplayName = ""
        objVarInfo.Value = ""
        objVarInfo.DataType = ENUM_IOC_DATATYPE_OBJECT
        objVarInfo.DataDescr = "Destination"
        objVarInfo.Attribute = ENUM_VAR_ATTR_REQUIRED

        Set objVarInfo = plstInput.Add
        objVarInfo.Name = "Package_Info"
        objVarInfo.DisplayName = ""
        objVarInfo.Value = ""
        objVarInfo.DataType = ENUM_IOC_DATATYPE_OBJECT
        objVarInfo.DataDescr = "Package_Information"
        objVarInfo.Attribute = ENUM_VAR_ATTR_REQUIRED
```

```
        Set objVarInfo = plstOutput.Add
        objVarInfo.Name = "Service_Rate"
        objVarInfo.DisplayName = ""
        objVarInfo.Value = ""
        objVarInfo.DataType = ENUM_IOC_DATATYPE_OBJECT
        objVarInfo.DataDescr = "Service Rate"
        objVarInfo.Attribute = ENUM_VAR_ATTR_REQUIRED

        PsIoDriver_GetTransactionByName = True

    ElseIf bstrTransName = "Tracking" Then
        '  Set the inputs and outputs for this transaction
        PsIoDriver_GetTransactionByName = True

    ElseIf bstrTransName = "Time-in-Transit" Then
        '  Set the inputs and outputs for this transaction
        PsIoDriver_GetTransactionByName = True

    Else
        PsIoDriver_GetTransactionByName = False
    End If
    '
End Function

' Provides a version number for this plug-in.
Private Function PsIoDriver_GetVer() As String
    '
    PsIoDriver_GetVer = "1.0.0"
    '
End Function

' Sets the Business Interlink type to transaction, static
' categories.
Private Function PsIoDriver_IsSupported(ByVal eType As
PSIODRIVERLib.ENUM_EIOCINTERFACESUPPORTED) As Long
    '
    Select Case eType
        Case ENUM_IOC_TRANSACTION
            PsIoDriver_IsSupported = True
        Case ENUM_IOC_STATICCATEGORY
            PsIoDriver_IsSupported = True
        Case ENUM_IOC_ISWCHAR
            PsIoDriver_IsSupported = True
        Case Else
            PsIoDriver_IsSupported = False
```

```
    End Select
    '
End Function


Private Function PsIoDriver_SetCategoryType(ByVal eCatType As
PSIODRIVERLib.ENUM_EIOCINTERFACESUPPORTED) As Long
    '
    PsIoDriver_SetCategoryType = False
    '
End Function
Private Function PsIoDriver_SetParameterList(ByVal plstParams As
PSIODRIVERLib.IPsEnumVarInfo) As Long
    '
    PsIoDriver_SetParameterList = False
    '
End Function
```

Example of Design Methods (C++)

Here is a sample set of methods for a design-time plug-in. It has two transactions to add numbers and concatenate strings. Following this sample is a

- The IsSupported methods shows that the Business Interlink Plug-in supports the transaction Interlink type.
- The GetParameterList method provides a string type named Output_File_Name to allow the Business Interlink to output to a file.
- The GetTransactionByCategory method provides the transaction names of add numbers and concatenate strings. The GetTransactionByName provides, for the add numbers transaction, the integer type input parameters number_1 and number_2 and the integer type output parameter sum; and it provides for the concatenate strings transaction the string type input parameters string_1 and string_2 and the string type output parameter string.

```
const TCHAR * SimpleDriver::GetDesc() const
{
return _T("Simple interface driver: adds 2 #, concats 2 strs");
}


const TCHAR * SimpleDriver::GetVersion() const
{
return _T("1.00");
}
```

```
BOOL SimpleDriver::IsSupported(EIOCINTERFACESUPPORTED option)
{
   switch(option)
   {
      case IOC_TRANSACTION:
      case IOC_STATICCATEGORY:
      return TRUE;

      case IOC_ISWCHAR:
      return  (sizeof(TCHAR) != sizeof(char));

      default:
      return FALSE;
   }
}

// Create the transaction categories.
BOOL SimpleDriver::GetCategories(EIOCINTERFACESUPPORTED type, IOSTRINGLIST&
criteria1Names)
{
if(type == IOC_TRANSACTION)
{
   criteria1Names.push_back(_T("Transactions"));
   criteria1Names.push_back(_T("All Transactions"));
}
return TRUE;
}

// Create the transaction names.
BOOL SimpleDriver::GetTransactionByCategory(const PSIOString& strFilter, const
PSIOString& strCriteria, IOSTRINGLIST& transNames)
{
    transNames.push_back(_T("add numbers"));
    transNames.push_back(_T("concatenate strings"));
    transNames.push_back(_T("get machine name"));
    return TRUE; // pTransNames;
}

BOOL SimpleDriver::GetTransactionByName(const PSIOString& strTransName,
TransactionDef& transDef)
{
    transDef.SetTransactionName(strTransName);

/* Add the input parameters number_1 and number_2, and the output parameter
sum, to the add numbers transaction. */
    if(strTransName == _T("add numbers"))
    {
```

```
        transDef.AddInput(VarInfo( _T("number_1"),
           DataType(IOC_DATATYPE_INT, _T("")), _T(""), _T(""), _T(""),
           eVAR_ATTR_REQUIRED));
        transDef.AddInput(VarInfo( _T("number_2"),
           DataType(IOC_DATATYPE_INT, _T("")), _T(""), _T(""), _T(""),
           eVAR_ATTR_REQUIRED));
        transDef.AddOutput(VarInfo( _T("sum"),
           DataType(IOC_DATATYPE_INT, _T("")), _T(""), _T(""), _T(""),
           eVAR_ATTR_REQUIRED));
        return TRUE;
    }
/* Add the input parameters string_1 and string_2, and the output parameter
string, to the concatenate strings transaction. */
    else if(strTransName == _T("concatenate strings"))
    {
        transDef.AddInput(VarInfo( _T("string_1"),
           DataType(IOC_DATATYPE_STRING, _T("")), _T(""), _T(""), _T(""),
           eVAR_ATTR_REQUIRED));
        transDef.AddInput(VarInfo( _T("string_2"),
           DataType(IOC_DATATYPE_STRING, _T("")), _T(""), _T(""), _T(""),
           eVAR_ATTR_REQUIRED));
        transDef.AddOutput(VarInfo( _T("string"),
           DataType(IOC_DATATYPE_STRING, _T("")), _T(""), _T(""), _T(""),
           eVAR_ATTR_REQUIRED));
        return TRUE;
    }
/* Add the output parameter machine_name to the get machine name transaction.
*/
    else if(strTransName == _T("get machine name"))
    {
        transDef.AddOutput(VarInfo( _T("machine_name"),
           DataType(IOC_DATATYPE_STRING, _T("")), _T(""), _T(""), _T(""),
           eVAR_ATTR_REQUIRED));
        return TRUE;
    }
    else
        return FALSE;
}




BOOL  SimpleDriver::GetClassByCategory(const PSIOString& strFilter, const
PSIOString& Category, IOSTRINGLIST& list)
{
        return FALSE;
```

```
}


BOOL  SimpleDriver::GetClassByName(const PSIOString& strClassName, ClassDef&
classes)
{
        return FALSE;
}


BOOL  SimpleDriver::GetCriteriaRelationalOperators(const PSIOString& strType,
IOSTRINGLIST& list)
{
        return FALSE;
}

BOOL  SimpleDriver::GetCriteriaLogicalOperators(IOSTRINGLIST& list)
{
        return FALSE;
}


VarInfo config(_T("Output_File_Name"), DataType(IOC_DATATYPE_STRING, _T("")),
_T(""), _T(""), _T(""), eVAR_ATTR_NONE);

BOOL  SimpleDriver::GetParameterList(VARINFOLIST& list)
{
        list.push_back(&config);
        return TRUE;
}
```

Dynamic Catalog Class Methods: Writing Design-Time Functionality

When you do not use an XML design-time plug-in, you write the design-time functionality using the dynamic catalog class methods. The class for these methods is the class for your Interlink Object.

FetchNextChunk, PsloDriver_FetchNextChunk

Syntax: C++

```
virtual BOOL FetchNextChunk(InterfaceObject *);
```

Syntax: Visual Basic

```
FetchNextChunk(IPsEnumString) As Long
```

Class

C++: DLLBaseDriver

Visual Basic: PsIoDriver

Description

Write this method to return the next chunk of the input or output table.

Parameters

IpsEnumString	The table from which a chunk is being returned.
InterfaceObject *	The interlink object.

Return Value

Value	*Meaning*
BOOL	True if a chunk was returned, false otherwise.
Long	

GetCategories, PsIoDriver_GetCategories

Syntax: C++

```
virtual BOOL GetCategories(EIOCINTERFACESUPPORTED type, IOSTRINGLIST& list) =
0;
```

Syntax: Visual Basic

```
GetCategories(type As ENUM_EIOCINTERFACESUPPORTED, list As IPsEnumString) As
Long
```

Class

C++: DLLBaseDriver

Visual Basic: PsIoDriver

Description

If there is no XML design-time plug-in for your runtime plug-in, you write the design-time functionality with the dynamic catalog methods. This is one of those methods. If you have an XML design-time plug-in, do not write this method.

Write this method to return a list of categories for a given interface type: transactions or classes. This is used in the Business Interlink Search page. Since GetCategories is a virtual method, you must write the code for this method.

Parameters

Type	The interface type for which to list categories.
List	The list of categories that is returned.

Return Value

Value	*Meaning*
BOOL	True if a list of categories is returned, False otherwise.
Long	
List	A list of strings containing the categories.

Example

The following C++ example sets the categories to be Transactions and All Transactions.

```
BOOL SimpleDriver::GetCategories(EIOCINTERFACESUPPORTED type, IOSTRINGLIST&
criteria1Names)
{
if(type == IOC_TRANSACTION)
{
   criteria1Names.push_back(_T("Transactions"));
   criteria1Names.push_back(_T("All Transactions"));
}
return TRUE;
}
```

The following Visual Basic example sets the categories to be Transactions and All Transactions.

```
Private Function PsIoDriver_GetCategories(ByVal eCatType As
PSIODRIVERLib.ENUM_EIOCINTERFACESUPPORTED, ByVal plstCriterialNames As
PSIODRIVERLib.IPsEnumString) As Long
    '
    Dim objPsBstr As PsBstr
    Set objPsBstr = plstCriterialNames.Add
    objPsBstr.Data = "Transactions"
```

```
    Set objPsBstr = plstCriterialNames.Add
    objPsBstr.Data = "All Transactions"
    PsIoDriver_GetCategories = True
    '
End Function
```

GetClassByName, PsIoDriver_GetClassByName

Syntax: C++

```
virtual BOOL GetClassByName(const PSIOString& strClassName, ClassDef& classes)
= 0;
```

Syntax: Visual Basic

```
PsIoDriver_GetClassByName(ByVal strClassName As String, ByVal classes As
PSIODRIVERLib.IPsEnumVarInfo) As Long
```

Class

C++: DLLBaseDriver

Visual Basic: PsIoDriver

Description

If there is no XML design-time plug-in for your runtime plug-in, you write the design-time functionality with the dynamic catalog methods. This is one of those methods. If you have an XML design-time plug-in, do not write this method.

Write this methods to return a list of the class data members for the class named *classes*. Since GetClassByName is a virtual method, you must write the code for this method.

Parameters

strClassName	A character string. GetClassByName finds all the class names containing this character string.
classes	An array of strings returned that contains the data members for the class named *strClassName.*

Return Value

Value	*Meaning*
classes	An array of strings that contains the data members for the class named *strClassName.*
BOOL, Long	True if a class data member list was returned, False otherwise.

Example

The following C++ example adds the data members named sender, cc, subject, date, and body to the class named Emessage. The plug-in is an email plug-in. Call the SetClassName method to set the class name, and use the AddDataMemberDef to add the data members to the class.

For more information about the SetClassName and AddDataMemberDef methods, see SetClassName and AddDataMemberDef.

```
BOOL  EmailDriver::GetClassByName(const PSIOString& strClassName, ClassDef&
classes)
{
   if(strClassName == "Emessage")
   {
      classes.SetClassName(strClassName);
      classes.AddDataMemberDef(VarInfo(_T("sender"),
         DataType(IOC_DATATYPE_STRING, _T("")), _T(""), _T(""), _T(""),
         eVAR_ATTR_REQUIRED));
      classes.AddDataMemberDef(VarInfo(_T("cc"),
         DataType(IOC_DATATYPE_STRING, _T("")), _T(""), _T(""), _T(""),
         eVAR_ATTR_REQUIRED));
      classes.AddDataMemberDef(VarInfo(_T("subject"),
         DataType(IOC_DATATYPE_STRING, _T("")), _T(""), _T(""), _T(""),
         eVAR_ATTR_REQUIRED));
      classes.AddDataMemberDef(VarInfo(_T("date"),
         DataType(IOC_DATATYPE_STRING, _T("")), _T(""), _T(""), _T(""),
         eVAR_ATTR_REQUIRED));
      classes.AddDataMemberDef(VarInfo(_T("body"),
         DataType(IOC_DATATYPE_STRING, _T("")), _T(""), _T(""), _T(""),
         eVAR_ATTR_REQUIRED));
      return TRUE;
   }
   return FALSE;
}
```

Visual Basic:

The following Visual Basic example adds the data members named sender, cc, subject, date, and body to the class named Emessage. The plug-in is an email plug-in. Use the Add method to add the data members to the class.

For more information about the Add method, see Add.

```
Private Function PsIoDriver_GetClassByName(ByVal bstrClassName As String,
ByVal plstClasses As PSIODRIVERLib.IPsEnumVarInfo) As Long
```

```
'
Dim objVarInfo As VarInfo

If bstrClassName = "Emessage" Then
'
   Set objVarInfo = plstClasses.Add
   objVarInfo.Name = "sender"
   objVarInfo.DisplayName = ""
   objVarInfo.Value = ""
   objVarInfo.DataType = ENUM_IOC_DATATYPE_STRING
   objVarInfo.DataDescr = "sender"
   objVarInfo.Attribute = ENUM_VAR_ATTR_REQUIRED
'
   Set objVarInfo = plstClasses.Add
   objVarInfo.Name = "cc"
   objVarInfo.DisplayName = ""
   objVarInfo.Value = ""
   objVarInfo.DataType = ENUM_IOC_DATATYPE_STRING
   objVarInfo.DataDescr = "cc"
   objVarInfo.Attribute = ENUM_VAR_ATTR_REQUIRED
'
   Set objVarInfo = plstClasses.Add
   objVarInfo.Name = "subject"
   objVarInfo.DisplayName = ""
   objVarInfo.Value = ""
   objVarInfo.DataType = ENUM_IOC_DATATYPE_STRING
   objVarInfo.DataDescr = "subject"
   objVarInfo.Attribute = ENUM_VAR_ATTR_REQUIRED
'
   Set objVarInfo = plstClasses.Add
   objVarInfo.Name = "date"
   objVarInfo.DisplayName = ""
   objVarInfo.Value = ""
   objVarInfo.DataType = ENUM_IOC_DATATYPE_STRING
   objVarInfo.DataDescr = "date"
   objVarInfo.Attribute = ENUM_VAR_ATTR_REQUIRED
'
   Set objVarInfo = plstClasses.Add
   objVarInfo.Name = "body"
   objVarInfo.DisplayName = ""
   objVarInfo.Value = ""
   objVarInfo.DataType = ENUM_IOC_DATATYPE_STRING
   objVarInfo.DataDescr = "body"
   objVarInfo.Attribute = ENUM_VAR_ATTR_REQUIRED
'
```

```
   PsIoDriver_GetClassByName = True
'
Else
   PsIoDriver_GetClassByName = False
End If
```

GetCriteriaRelationalOperators, PsIoDriver_GetCriteriaRelationalOperators

Syntax: C++

```
virtual BOOL GetCriteriaRelationalOperators(const PSIOString& strType,
IOSTRINGLIST& list) = 0;
```

Syntax: Visual Basic

```
PsIoDriver_GetCriteriaRelationalOperators(ByVal strType As String, list As
IPsEnumString) As Long
```

Class

C++: DLLBaseDriver

Visual Basic: PsIoDriver

Description

If there is no XML design-time plug-in for your runtime plug-in, you write the design-time functionality with the dynamic catalog methods. This is one of those methods. If you have an XML design-time plug-in, do not write this method.

Write this method to return a list of all the relational operators for queries. The server knows what actions to perform for each operator; this method tells the server what action is performed for what symbol. Since GetCriteriaRelationalOperators is a virtual method, you must write the code for this method.

Parameters

StrType	A character string.
List	An array of strings returned that contains the relational operators for the data member type *strType.*

Return Value

Value	Meaning
List	An array of strings that contains the relational operators for the data member type *strType.*

Value	*Meaning*
BOOL Long	True if a relational operator list was returned, False otherwise.

Example

The following C++ example creates the relational operators of =, >, <, and <>.

```
BOOL RecordDriver::GetCriteriaRelationalOperators(const PSIOString& strType,
IOSTRINGLIST& alist)
{
   alist.push_back(_T("="));
   alist.push_back(_T(">"));
   alist.push_back(_T("<"));
   alist.push_back(_T("<>"));
   return TRUE;
}
```

The following Visual Basic example creates the relational operators of =, >, <, and <>.

```
Private Function PsIoDriver_GetCriteriaRelationalOperators(ByVal bstrType As
String, ByVal plstOperators As PSIODRIVERLib.IPsEnumString) As Long
    '
    Dim objPsBstr As PsBstr
    Set objPsBstr = plstOperators.Add
    objPsBstr.Data = "="
    Set objPsBstr = plstOperators.Add
    objPsBstr.Data = ">"
    Set objPsBstr = plstOperators.Add
    objPsBstr.Data = "<"
    Set objPsBstr = plstOperators.Add
    objPsBstr.Data = "<>"
    PsIoDriver_GetCriteriaRelationalOperators = True
    '
End Function
```

GetCriteriaLogicalOperators, PsIoDriver_GetCriteriaLogicalOperators

Syntax: C++

```
virtual IOSTRINGLIST GetCriteriaLogicalOperators(const PSIOString& strType) =
0;
```

Syntax: Visual Basic

```
PsIoDriver_GetCriteriaLogicalOperators(ByVal plstOperators As
PSIODRIVERLib.IPsEnumString) As Long
```

Class

C++: DLLBaseDriver

Visual Basic: PsIoDriver

Description

If there is no XML design-time plug-in for your runtime plug-in, you write the design-time functionality with the dynamic catalog methods. This is one of those methods. If you have an XML design-time plug-in, do not write this method.

Write this method to return a list of all the logical operators for queries. The server knows what actions to perform for each operator; this method tells the server what action is performed for what symbol. Since GetCriteriaLogicalOperators is a virtual method, you must write the code for this method.

Parameters

list	An array of strings returned that contains the logical operators for queries.

Return Value

Value	Meaning
list	An array of strings that contains the logical operators for queries.
BOOL	True if a logical operator list was returned, False otherwise.

Example

The following C++ example creates the logical operators of AND, OR, and NOT.

```
BOOL  RecordDriver::GetCriteriaLogicalOperators(IOSTRINGLIST& alist)
{
   alist.push_back(_T("AND"));
   alist.push_back(_T("OR"));
   alist.push_back(_T("NOT"));
   return TRUE;
}
```

The following Visual Basic example creates the logical operators of AND, OR, and NOT.

```
Private Function PsIoDriver_GetCriteriaLogicalOperators(ByVal bstrType As
String, ByVal plstOperators As PSIODRIVERLib.IPsEnumString) As Long
    '
    Dim objPsBstr As PsBstr
    Set objPsBstr = plstOperators.Add
    objPsBstr.Data = "AND"
    Set objPsBstr = plstOperators.Add
    objPsBstr.Data = "OR"
    Set objPsBstr = plstOperators.Add
    objPsBstr.Data = "NOT"
    PsIoDriver_GetCriteriaLogicalOperators = True
    '
End Function
```

GetDesc, PsIoDriver_GetDesc

Syntax: C++

```
virtual const TCHAR * GetDesc() = 0;
```

Syntax: Visual Basic

```
PsIoDriver_GetDesc() As String
```

Class

C++: DLLBaseDriver

Visual Basic: PsIoDriver

Description

If there is no XML design-time plug-in for your runtime plug-in, you write the design-time functionality with the dynamic catalog methods. This is one of those methods. If you have an XML design-time plug-in, do not write this method.

Write this method to return a string containing the description of this Business Interlink. This string is used in the New Business Interlink page. Since GetDesc is a virtual method, you must write the code for this method.

Parameters

None.

Return Value

Value	*Meaning*
TCHAR*, String.	Contains the description of this Business Interlink.

Example

The following C++ example creates the description for a plug-in that adds numbers and concatentates strings.

```
const TCHAR * SimpleDriver::GetDesc() const
{
return _T("Simple interface plug-in: adds 2 #, concats 2 strs");
}
```

The following Visual Basic example creates the description for a plug-in that adds numbers and concatentates strings.

```
Private Function PsIoDriver_GetDesc() As String
    '
    PsIoDriver_GetDesc =
       "Simple interface plug-in: adds 2 #, concats 2 strs"
    '
End Function
```

GetLastErrorMessage

Syntax: C++

```
virtual const IODrvrMsg& GetLastErrorMessage () { return m_ErrMsg; }
```

Syntax: Visual Basic

```
GetLastErrorMessage(pDrvMsg As IPsDriverMessage)
```

Class

C++: DLLBaseDriver

Visual Basic: PsIoDriver

Description

If there is no XML design-time plug-in for your runtime plug-in, you write the design-time functionality with the dynamic catalog methods. This is one of those methods. If you have an XML design-time plug-in, do not write this method.

Write this method to return the latest error message.

Parameters

None.

Return Value

Value	Meaning
m_ErrMsg, pDrvMsg	The returned error message.

GetClassByCategory, PsIoDriver_GetClassByCategory

Syntax: C++

```
virtual BOOL GetClassByCategory(const PSIOString& strFilter, const PSIOString&
strCategory, IOSTRINGLIST& list) = 0;
```

Syntax: Visual Basic

```
PsIoDriver_GetClassByCategory(strFilter As String, strCategory As String, list
As IPsEnumString) As Long
```

Class

C++: DLLBaseDriver

Visual Basic: PsIoDriver

Description

If there is no XML design-time plug-in for your runtime plug-in, you write the design-time functionality with the dynamic catalog methods. This is one of those methods. If you have an XML design-time plug-in, do not write this method.

Write this method to return a list of class names based upon the given *Category*. This is used in the Interface Definition Search page. Since GetClassByCategory is a virtual method, you must write the code for this method.

Parameters

strFilter	A string that you can search on, such as "sales" to find all transactions/classes with "sales" in their names.
strCategory	A string containing the category.
list	The array of strings returned that contains the class names (or descriptions) in this category.

Return Value

Value	Meaning
list	The list of class names (or descriptions) for the given category.

Value	*Meaning*
BOOL, Long	True if an array is returned, false otherwise.

Example

The following C++ example creates a list of two class names: class1 and class2. It places class1 into category1 and class1 and class2 into category2.

```
BOOL SimpleDriver::GetClassByCategory
   (const PSIOString& strFilter,
    const PSIOString& strCategory,
    IOSTRINGLIST& classNames)
{
   if (strCategory == _T("category1"))
   {
      classNames.push_back(_T("class1"));
      return TRUE;
   }
   if (strCategory == _T("category2"))
   {
      classNames.push_back(_T("class1"));
      classNames.push_back(_T("class2"));
      return TRUE;
   }
return FALSE; //
}
```

The following Visual Basic example creates a list of two class names: class1 and class2.

```
Private Function PsIoDriver_GetClassByCategory
   (ByVal strFilter As String,
    ByVal strCriteria As String,
    ByVal plstClassNames As PSIODRIVERLib.IPsEnumString) As Long
   '
   Dim objPsClassCategory As PsBstr
   If (strCriteria = "category1" Then
      Set objPsClassCategory = plstClassNames.Add
      objPsClassCategory.Data = "class1"
      PsIoDriver_GetClassByCategory = True
   ElseIf (strCriteria = "category2" Then
      Set objPsClassCategory = plstClassNames.Add
      objPsClassCategory.Data = "class1"
      Set objPsClassCategory = plstClassNames.Add
      objPsClassCategory.Data = "class2"
      PsIoDriver_GetClassByCategory = True
```

```
    Else
       PsIoDriver_GetClassByCategory = False
    End If
    '
End Function
```

GetParameterList, PsIoDriver_GetParameterList

Syntax: C++

```
virtual BOOL GetParameterList(VARINFOLIST& list) = 0;
```

Syntax: Visual Basic

```
PsIoDriver_GetParameterList(list As IPsEnumVarInfo) As Long
```

Class

C++: DLLBaseDriver

Visual Basic: PsIoDriver

Description

If there is no XML design-time plug-in for your runtime plug-in, you write the design-time functionality with the dynamic catalog methods. This is one of those methods. If you have an XML design-time plug-in, do not write this method.

Write this method to return an array containing the data types and names of the configuration parameters for this system. Since GetParameterList is a virtual method, you must write the code for this method.

Parameters

list	An array of strings containing the configuration parameters.

Return Value

Value	*Meaning*
list	The list of configuration parameters.
BOOL	True if an array is returned, false otherwise.

Example

The following C++ example creates a configuration parameter named Output_File_Name. Use the push_back method to create the parameter.

For more information about the parameter data structure, see Parameter Lists.

Each push_back call creates a data member containing the following information:

- A string containing the data member name.
- The data type of the variable.
- A string containing a class name if the DataType is IOC_DATATYPE_OBJECT or IOC_DATATYPE_LIST_OBJECT.
- A string containing the default value, if any.
- A bool (TRUE or FALSE): whether or not this data member is required. eVAR_ATTR_NONE means not required, eVAR_ATTR_REQUIRED means required.

```
BOOL  SimpleDriver::GetParameterList(VARINFOLIST& list)
{
   list.push_back(new VarInfo(_T("Output_File_Name"),
      DataType(IOC_DATATYPE_STRING, _T("")), _T(""), _T(""), _T(""),
      eVAR_ATTR_NONE));
   return TRUE;
}
```

Visual Basic:

The following Visual Basic example creates a configuration parameter named Output_File_Name. Use the Add method to add the parameter.

For more information about the Add method, see Add.

```
Private Function PsIoDriver_GetParameterList(ByVal plstParams As
PSIODRIVERLib.IPsEnumVarInfo) As Long
   '
   Dim objVarInfo As VarInfo
   Set objVarInfo = plstParams.Add
   objVarInfo.Name = "Output_File_Name"
   objVarInfo.DisplayName = ""
   objVarInfo.Value = ""
   objVarInfo.DataType = ENUM_IOC_DATATYPE_STRING
   objVarInfo.Attribute = ENUM_VAR_ATTR_NONE
   PsIoDriver_GetParameterList = True
   '
End Function
```

GetTransactionByCategory, PsIoDriver_GetTransactionByCategory

Syntax: C++

```
virtual IOSTRINGLIST * GetTransactionByCategory(const PSIOString& strFilter,
const PSIOString& Category, IOSTRINGLIST& transNames) = 0;
```

Syntax: Visual Basic

```
PsIoDriver_GetTransactionByCategory(strFilter As String, bstrCriteria As
String, TransNames As IPsEnumString) As Long
```

Class

C++: DLLBaseDriver

Visual Basic: PsIoDriver

Description

If there is no XML design-time plug-in for your runtime plug-in, you write the design-time functionality with the dynamic catalog methods. This is one of those methods. If you have an XML design-time plug-in, do not write this method.

Write this method to return a list of transaction names based upon the given *Category*. This is used in the Interface Definition Search page. Since GetTransactionByCategory is a virtual method, you must write the code for this method.

Parameters

strFilter	A string that you can search on, such as "sales" to find all transactions/classes with "sales" in their names.
strCriteria	A string containing the category.
transNames	The array of strings returned that contains the transaction names (or descriptions) in this category.

Return Value

Value	Meaning
transNames	The list of transaction names (or descriptions) for the given category.
BOOL, Long	True if an array is returned, false otherwise.

Example

The following C++ example creates a list of two transaction names: add numbers and concatenate strings.

For more information about the push_back method, see push_back.

```
BOOL SimpleDriver::GetTransactionByCategory
   (const PSIOString& strFilter,
    const PSIOString& strCriteria,
    IOSTRINGLIST& transNames)
{
   if (strCategory == _T("category1"))
   {
      transNames.push_back(_T("add numbers"));
      return TRUE;
   }
   if (strCategory == _T("category2"))
   {
      transNames.push_back(_T("add numbers"));
      transNames.push_back(_T("concatenate strings"));
      return TRUE;
   }
return FALSE; //
}
```

The following Visual Basic example creates a list of two transaction names: add numbers and concatenate strings.

```
Private Function PsIoDriver_GetTransactionByCategory
   (ByVal bstrFilter As String,
    ByVal bstrCriteria As String,
    ByVal plstTransNames As PSIODRIVERLib.IPsEnumString) As Long
   '
   Dim objPsTransactCategory As PsBstr
   If (strCriteria = "category1" Then
      Set objPsTransactCategory = plstTransNames.Add
      objPsTransactCategory.Data = "add numbers"
      PsIoDriver_GetTransactionByCategory = True
   ElseIf (strCriteria = "category2" Then
      Set objPsTransactCategory = plstTransNames.Add
      objPsTransactCategory.Data = "add numbers"
      Set objPsTransactCategory = plstTransNames.Add
      objPsTransactCategory.Data = "concatenate strings"
      PsIoDriver_GetTransactionByCategory = True
   Else
      PsIoDriver_GetTransactionByCategory = False
   End If
   '
End Function
```

GetTransactionByName, PsIoDriver_GetTransactionByName

Syntax: C++

```
virtual BOOL GetTransactionByName(const PSIOString& strTransName,
TransactionDef& transactions) = 0;
```

Syntax: Visual Basic

```
PsIoDriver_GetTransactionByName(strTransName As String, plstInput As
IPsEnumVarInfo, plstOutput As IPsEnumVarInfo) As Long
```

Class

C++: DLLBaseDriver

Visual Basic: PsIoDriver

Description

If there is no XML design-time plug-in for your runtime plug-in, you write the design-time functionality with the dynamic catalog methods. This is one of those methods. If you have an XML design-time plug-in, do not write this method.

Write this method to return a list of the names and data types of the input/output parameters for the transaction named *strTransName*. Since GetTransactionByName is a virtual method, you must write the code for this method.

Parameters

strTransName	A character string. GetTransactionByName finds all the transaction names containing this character string.
TransDef	The TransactionDef class. You will set an array of strings in that class that contains the input/output parameters for the transaction named *strTransName.*
PlstInput	

Return Value

Value	Meaning
TransDef	The array of strings that contains the input/output parameters for the transaction named *strTransName.*
BOOL, long	True if a transaction parameter list was returned, False otherwise.

Example

The following C++ example sets the input and output parameters for the transaction named add numbers and concatenate strings. For add numbers, the input parameters are number_1 and number_2, and the output parameter is sum. For concatenate strings, the input parameters are string_1 and string_2, and the output parameter is string. Use the AddInput method to add the input parameters, and use the method AddOutput to add the output parameters.

For more information about the AddInput and AddOutput methods, see AddInput and AddOutput.

```
BOOL SimpleDriver::GetTransactionByName(const PSIOString& strTransName,
TransactionDef& transDef)
{
   transDef.SetTransactionName(strTransName);

   if(strTransName == _T("add numbers"))
   {
      transDef.AddInput(VarInfo( _T("number_1"),
         DataType(IOC_DATATYPE_INT, _T("")), _T(""), _T(""), _T(""),
         eVAR_ATTR_REQUIRED));
      transDef.AddInput(VarInfo( _T("number_2"),
         DataType(IOC_DATATYPE_INT, _T("")), _T(""), _T(""), _T(""),
         eVAR_ATTR_REQUIRED));
      transDef.AddOutput(VarInfo( _T("sum"),
         DataType(IOC_DATATYPE_INT, _T("")), _T(""), _T(""), _T(""),
         eVAR_ATTR_REQUIRED));
      return TRUE;
    }
    else if(strTransName == _T("concatenate strings"))
    {
       transDef.AddInput(VarInfo( _T("string_1"),
          DataType(IOC_DATATYPE_STRING, _T("")), _T(""), _T(""),
          _T(""), eVAR_ATTR_REQUIRED));
       transDef.AddInput(VarInfo( _T("string_2"),
          DataType(IOC_DATATYPE_STRING, _T("")), _T(""), _T(""),
          _T(""), eVAR_ATTR_REQUIRED));
       transDef.AddOutput(VarInfo( _T("string"),
          DataType(IOC_DATATYPE_STRING, _T("")), _T(""), _T(""),
          _T(""), eVAR_ATTR_REQUIRED));
       return TRUE;
    }
    else
       return FALSE;
}
```

Visual Basic:

The following Visual Basic example sets the input and output parameters for the transaction named add numbers and concatenate strings. For add numbers, the input parameters are number_1 and number_2, and the output parameter is sum. For concatenate strings, the input parameters are string_1 and string_2, and the output parameter is string. Use the Add method to add the input and output parameter information.

For more information about the Add method, see Add.

IsSupported, PsIoDriver_IsSupported

Syntax: C++

```
virtual BOOL IsSupported(EIOCINTERFACETYPE type) = 0;
```

Syntax: Visual Basic

```
PsIoDriver_IsSupported(ByVal Type As
PSIODRIVERLib.ENUM_EIOCINTERFACESUPPORTED) As Long
```

Class

C++: DLLBaseDriver

Visual Basic: PsIoDriver

Description

Write this method to return true or false: true if the given type is a supported interface type, false otherwise. Since IsSupported is a virtual method, you must write the code for this method.

Parameters

type — the interface type to test. The allowed values are:

C++:
IOC_UNKNOWN (0), IOC_OBJQUERY (1),
IOC_TRANSACTION (2), IOC_OBJADD (3),
IOC_OBJUPDATE (4), IOC_OBJDELETE (5),
IOC_ORDERBYASC(6), IOC_ORDERBYDESC(7),
IOC_INPUT_CLASSEXPANSION(8),
IOC_STAICCATEGORY(9),
IOC_DYNAMICCATEGORY(10), IOC_ISWCHAR(11)

Visual Basic:
ENUM_IOC_UNKNOWN (0),
ENUM_IOC_OBJQUERY (1),
ENUM_IOC_TRANSACTION (2),
ENUM_IOC_OBJADD (3), ENUM_IOC_OBJUPDATE

(4), ENUM_IOC_OBJDELETE (5),
ENUM_IOC_ORDERBYASC(6),
ENUM_IOC_ORDERBYDESC(7),
ENUM_IOC_INPUT_CLASSEXPANSION(8),
ENUM_IOC_STATICCATEGORY(9),
ENUM_IOC_DYNAMICCATEGORY(10),
ENUM_IOC_ISWCHAR(11)

Return Value

Value	*Meaning*
BOOL, Long	True if the given type is a supported interface type, False otherwise.

Example

The following C++ example sets the supported types for the SimpleDriver Business Interlink to support transactions, and to set the catagories to static (the input/output parameters for the transaction are not changeable in data type or number of parameters).

```
BOOL SimpleDriver::IsSupported(EIOCINTERFACESUPPORTED option)
{
   switch(option)
   {
      case IOC_TRANSACTION:
      case IOC_STATICCATEGORY:
      return TRUE;

      case IOC_ISWCHAR:
      return  (sizeof(TCHAR) != sizeof(char));

      default:
      return FALSE;
   }
}
```

The following Visual Basic example sets the supported types for the SimpleDriver Business Interlink to support transactions, and to set the catagories to static (the input/output parameters for the transaction are not changeable in data type or number of parameters).

```
Private Function PsIoDriver_IsSupported(ByVal eType As
PSIODRIVERLib.ENUM_EIOCINTERFACESUPPORTED) As Long
    '
    Select Case eType
        Case ENUM_IOC_TRANSACTION
            PsIoDriver_IsSupported = True
        Case ENUM_IOC_STATICCATEGORY
```

```
            PsIoDriver_IsSupported = True
        Case ENUM_IOC_ISWCHAR
            PsIoDriver_IsSupported = True
        Case Else
            PsIoDriver_IsSupported = False
    End Select
    '
End Function
```

Transaction/Class Methods: Adding Input/Output Parameters to Transactions, Data Members to Classes

You use the Transaction and Class method when you write the design-time functionality using the dynamic catalog class methods, instead of using an XML design-time plug-in. The class for these methods is the class for your Interlink Object. These methods allow you to set information when you design the Transactions and for Classes for your Business Interlink Plug-in.

For C++, you must include the header file ioutil.h for this class.

The following table shows which methods you use to operate on the input and output tables:

Action to perform:	***Use this method***
Add a parameter to a transaction or a data member to a class in Visual Basic	Add
Add a data member to a class in C++	AddDataMemberDef
Add an input parameter to a transaction in C++	AddInput
Add an output parameter to a transaction in C++	AddOutput
Set a class name in C++	SetClassName
Set a transaction name in C++	SetTransactionName

Add

Syntax: Visual Basic

```
Add()
```

Class

IPsEnumVarInfo

Description

If there is no XML design-time plug-in for your runtime plug-in, you write the design-time functionality with the dynamic catalog methods. This is one of those methods. If you have an XML design-time plug-in, do not write this method.

The Visual Basic **Add** method adds a configuration parameter, an input or output parameter to a transaction, a data member to a class, or a category name to the category list for the transactions or classes. When adding parameters or data members, you supply the following information:

- A string containing the parameter or data member name.
- The data type of the variable.

For more information about the parameter data structure and the data types, see Parameter Lists.

- A string containing a display name. This name shows in the Input Name and Output Name columns in the Application Designer for Business Interlinks.
- A string containing the default value, if any.
- An attribute: ENUM_VAR_ATTR_REQUIRED if this data member is required, ENUM_VAR_ATTR_NONE if not required.

Parameters

None.

Return Value

None.

Example

In the following method, the **Add** method adds the category names Transactions and All Transactions to the list of categories.

```
Private Function PsIoDriver_GetCategories(ByVal eCatType As
PSIODRIVERLib.ENUM_EIOCINTERFACESUPPORTED, ByVal plstCriterialNames As
PSIODRIVERLib.IPsEnumString) As Long
   '
   Dim objPsBstr As PsBstr
   Set objPsBstr = plstCriterialNames.Add
   objPsBstr.Data = "Transactions"
   Set objPsBstr = plstCriterialNames.Add
   objPsBstr.Data = "All Transactions"
   PsIoDriver_GetCategories = True
   '
End Function
```

In the following method, the **Add** method adds a configuration parameter named Output File Name.

```
Private Function PsIoDriver_GetParameterList(ByVal plstParams As
PSIODRIVERLib.IPsEnumVarInfo) As Long
    '
    Dim objVarInfo As VarInfo
    Set objVarInfo = plstParams.Add
    objVarInfo.Name = "Output_File_Name"
    objVarInfo.DisplayName = ""
    objVarInfo.Value = ""
    objVarInfo.DataType = ""
    objVarInfo.Attribute = ENUM_VAR_ATTR_NONE
    PsIoDriver_GetParameterList = True
    '
End Function
```

In the following method, the **Add** method adds two transaction names: add numbers and concatenate strings.

```
Private Function PsIoDriver_GetTransactionByCategory(ByVal bstrFilter As
String, ByVal bstrCriteria As String, ByVal plstTransNames As
PSIODRIVERLib.IPsEnumString) As Long
    '
    Dim objPsTransactCategory As PsBstr
    Set objPsTransactCategory = plstTransNames.Add
    objPsTransactCategory.Data = "add numbers"
    Set objPsTransactCategory = plstTransNames.Add
    objPsTransactCategory.Data = "concatenate strings"
    PsIoDriver_GetTransactionByCategory = True
    '
End Function
```

In the following method, the **Add** method adds the input parameters of number_1 and number_2 and the output parameter of sum to the add numbers transaction.

```
Private Function PsIoDriver_GetTransactionByName(ByVal bstrTransName As
String, ByVal plstInput As PSIODRIVERLib.IPsEnumVarInfo, ByVal plstOutput As
PSIODRIVERLib.IPsEnumVarInfo) As Long
   '
   Dim objVarInfo As VarInfo

   If bstrTransName = m_bstrAddNumber Then
      Set objVarInfo = plstInput.Add
      objVarInfo.Name = "number_1"
      objVarInfo.DisplayName = ""
      objVarInfo.Value = ""
```

```
        objVarInfo.DataType = 0
        objVarInfo.Attribute = ENUM_VAR_ATTR_REQUIRED

        Set objVarInfo = plstInput.Add
        objVarInfo.Name = "number_2"
        objVarInfo.DisplayName = ""
        objVarInfo.Value = ""
        objVarInfo.DataType = 0
        objVarInfo.Attribute = ENUM_VAR_ATTR_NONE

        Set objVarInfo = plstOutput.Add
        objVarInfo.Name = "sum"
        objVarInfo.DisplayName = ""
        objVarInfo.Value = ""
        objVarInfo.DataType = 0
        objVarInfo.Attribute = ENUM_VAR_ATTR_REQUIRED
        PsIoDriver_GetTransactionByName = True
    End If
End Function
```

AddInput

Syntax: C++

Class: TransactionDef

```
BOOL AddInput(const VarInfo& var);
```

Description

If there is no XML design-time plug-in for your runtime plug-in, you write the design-time functionality with the dynamic catalog methods. This is one of those methods. If you have an XML design-time plug-in, do not write this method.

The C++ **AddInput** method adds an input parameter to a transaction. The input parameter is of type VarInfo. VarInfo consists of the following structure:

- PSIOString name: the name of the parameter.
- PSIOString default: any default value for this parameter.
- DataType datatype: the data type of this parameter.
- BOOL required: indicates if the parameter is required.

For more information about the parameter data structure and the data types, see Parameter Lists.

Parameters

var the variable that is added as an input parameter.

Return Value

Value	*Meaning*
BOOL, Long	True if the input parameter was added to the transaction, False otherwise.

Example

In the following method, the **AddInput** method adds the input parameters number_1 and number_2 to the add numbers transaction.

```
BOOL SimpleDriver::GetTransactionByName(const PSIOString& strTransName,
TransactionDef& transDef)
{
   transDef.SetTransactionName(strTransName);

   if(strTransName == _T("add numbers"))
   {
      transDef.AddInput(VarInfo( _T("number_1"),
         DataType(IOC_DATATYPE_INT, _T("")), _T(""), _T(""), _T(""),
         eVAR_ATTR_REQUIRED));
      transDef.AddInput(VarInfo( _T("number_2"),
         DataType(IOC_DATATYPE_INT, _T("")), _T(""), _T(""), _T(""),
         eVAR_ATTR_NONE));
      transDef.AddOutput(VarInfo( _T("sum"),
         DataType(IOC_DATATYPE_INT, _T("")), _T(""), _T(""), _T(""),
         eVAR_ATTR_REQUIRED));
      return TRUE;
   }
   else
      return FALSE;
}
```

AddDataMemberDef

Syntax: C++

Class: ClassDef

```
BOOL AddDataMemberDef(const VarInfo& var);
```

Description

If there is no XML design-time plug-in for your runtime plug-in, you write the design-time functionality with the dynamic catalog methods. This is one of those methods. If you have an XML design-time plug-in, do not write this method.

The C++ **AddDataMemberDef** method adds a data member to a class. The data member is of type VarInfo. VarInfo consists of the following structure:

- PSIOString name: the name of the data member.
- PSIOString default: any default value for this data member.
- DataType datatype: the data type of this data member.
- BOOL required: indicates if the data member is required.

For more information about the parameter data structure and the data types, see Parameter Lists.

Parameters

var the variable that is added as a data member.

Return Value

Value	Meaning
BOOL, Long	True if the data member was added to the class, False otherwise.

Example

In the following method, the **AddDataMemberDef** method adds the data members sender, cc, subject, date, and body to the Emessage class.

```
BOOL  EmailDriver::GetClassByName(const PSIOString& strClassName, ClassDef&
classes)
{
   if(strClassName == "Emessage")
   {
      classes.SetClassName(strClassName);
      classes.AddDataMemberDef(VarInfo(_T("sender"),
         DataType(IOC_DATATYPE_STRING, _T("")), _T(""), _T(""), _T(""),
         eVAR_ATTR_REQUIRED));
      classes.AddDataMemberDef(VarInfo(_T("cc"),
         DataType(IOC_DATATYPE_STRING, _T("")), _T(""), _T(""), _T(""),
         eVAR_ATTR_REQUIRED));
      classes.AddDataMemberDef(VarInfo(_T("subject"),
```

```
            DataType(IOC_DATATYPE_STRING, _T("")), _T(""), _T(""), _T(""),
            eVAR_ATTR_REQUIRED));
         classes.AddDataMemberDef(VarInfo(_T("date"),
            DataType(IOC_DATATYPE_STRING, _T("")), _T(""), _T(""), _T(""),
            eVAR_ATTR_REQUIRED));
         classes.AddDataMemberDef(VarInfo(_T("body"),
            DataType(IOC_DATATYPE_STRING, _T("")), _T(""), _T(""), _T(""),
            eVAR_ATTR_REQUIRED));
         return TRUE;
      }
      return FALSE;
   }
```

AddOutput

Syntax: C++

Class: TransactionDef

```
BOOL AddOutput(const VarInfo& var);
```

Description

If there is no XML design-time plug-in for your runtime plug-in, you write the design-time functionality with the dynamic catalog methods. This is one of those methods. If you have an XML design-time plug-in, do not write this method.

The C++ **AddOutput** method adds an output parameter to a transaction. The output parameter is of type VarInfo. VarInfo consists of the following structure:

- PSIOString name: the name of the parameter.
- PSIOString default: any default value for this parameter.
- DataType datatype: the data type of this parameter.
- BOOL required: indicates if the parameter is required.

For more information about the parameter data structure and the data types, see Parameter Lists.

Parameters

var	the variable that is added as an input parameter.

Return Value

Value	*Meaning*
BOOL, Long	True if the output parameter was added to the transaction, False otherwise.

Example

In the following method, the **AddOutput** method adds the output parameter sum to the add numbers transaction.

```
BOOL SimpleDriver::GetTransactionByName(const PSIOString& strTransName,
TransactionDef& transDef)
{
   transDef.SetTransactionName(strTransName);

   if(strTransName == _T("add numbers"))
   {
      transDef.AddInput(VarInfo( _T("number_1"),
         DataType(IOC_DATATYPE_INT, _T("")), _T(""), _T(""), _T(""),
         eVAR_ATTR_REQUIRED));
      transDef.AddInput(VarInfo( _T("number_2"),
         DataType(IOC_DATATYPE_INT, _T("")), _T(""), _T(""), _T(""),
         eVAR_ATTR_NONE));
      transDef.AddOutput(VarInfo( _T("sum"),
         DataType(IOC_DATATYPE_INT, _T("")), _T(""), _T(""), _T(""),
         eVAR_ATTR_REQUIRED));
      return TRUE;
   }
   else
      return FALSE;
}
```

SetClassName

Syntax: C++

Class: ClassDef

```
BOOL SetClassName(const PSIOString& strClassName);
```

Description

If there is no XML design-time plug-in for your runtime plug-in, you write the design-time functionality with the dynamic catalog methods. This is one of those methods. If you have an XML design-time plug-in, do not write this method.

The C++ **SetClassName** method set the name of a class.

Parameters

StrClassName the name of the class.

Return Value

Value	Meaning
BOOL, Long	True if the data member was added to the class, False otherwise.

Example

In the following method, the **SetClassName** method sets the class name to be the PSIOString that is passed into the GetClassByName method.

```
BOOL  EmailDriver::GetClassByName(const PSIOString& strClassName, ClassDef&
classes)
{
   if(strClassName == "Emessage")
   {
      classes.SetClassName(strClassName);
      classes.AddDataMemberDef(VarInfo(_T("sender"),
         DataType(IOC_DATATYPE_STRING, _T("")), _T(""), _T(""), _T(""),
         eVAR_ATTR_REQUIRED));
      classes.AddDataMemberDef(VarInfo(_T("cc"),
         DataType(IOC_DATATYPE_STRING, _T("")), _T(""), _T(""), _T(""),
         eVAR_ATTR_REQUIRED));
      classes.AddDataMemberDef(VarInfo(_T("subject"),
         DataType(IOC_DATATYPE_STRING, _T("")), _T(""), _T(""), _T(""),
         eVAR_ATTR_REQUIRED));
      classes.AddDataMemberDef(VarInfo(_T("date"),
         DataType(IOC_DATATYPE_STRING, _T("")), _T(""), _T(""), _T(""),
         eVAR_ATTR_REQUIRED));
      classes.AddDataMemberDef(VarInfo(_T("body"),
         DataType(IOC_DATATYPE_STRING, _T("")), _T(""), _T(""), _T(""),
         eVAR_ATTR_REQUIRED));
      return TRUE;
   }
   return FALSE;
}
```

SetTransactionName

Syntax: C++

Class: TransactionDef

```
BOOL SetTransactionName(const PSIOString& strTransactionName);
```

Description

If there is no XML design-time plug-in for your runtime plug-in, you write the design-time functionality with the dynamic catalog methods. This is one of those methods. If you have an XML design-time plug-in, do not write this method.

The C++ **SetTransactionName** method set the name of a transaction.

Parameters

StrTransactionName the name of the transaction.

Return Value

Value	*Meaning*
BOOL, Long	True if the input parameter was added to the transaction, False otherwise.

Example

In the following method, the **SetTransactionName** method set the transaction name to be the PSIOString that is passed into the GetTransactionByName method.

```
BOOL SimpleDriver::GetTransactionByName(const PSIOString& strTransName,
TransactionDef& transDef)
{
   transDef.SetTransactionName(strTransName);

   if(strTransName == _T("add numbers"))
   {
      transDef.AddInput(VarInfo( _T("number_1"),
         DataType(IOC_DATATYPE_INT, _T("")), _T(""), _T(""), _T(""),
         eVAR_ATTR_REQUIRED));
      transDef.AddInput(VarInfo( _T("number_2"),
         DataType(IOC_DATATYPE_INT, _T("")), _T(""), _T(""), _T(""),
         eVAR_ATTR_NONE));
      transDef.AddOutput(VarInfo( _T("sum"),
         DataType(IOC_DATATYPE_INT, _T("")), _T(""), _T(""), _T(""),
         eVAR_ATTR_REQUIRED));
```

```
        return TRUE;
    }
    else
        return FALSE;
}
```

push_back

Syntax: C++

Class: PSIOStringList

```
void push_back(PSIOString);

void push_back(char *);
```

Class: PSIOVarInfoList

```
void push_back(VarInfo *);
```

Description

Pushes data into a parameter.

Parameters

None.

Return Value

None.

Example

The following C++ code pushes a transaction category.

```
BOOL SimpleDriver::GetCategories(EIOCINTERFACESUPPORTED type, IOSTRINGLIST&
criteria1Names)
{
if(type == IOC_TRANSACTION)
{
   criteria1Names.push_back(_T("Transactions"));
   criteria1Names.push_back(_T("All Transactions"));
}
return TRUE;
}
```

The following C++ code pushes a configuration parameter.

```
BOOL  SimpleDriver::GetParameterList(VARINFOLIST& list)
```

```
{
   list.push_back(new VarInfo(_T("Output_File_Name"),
      DataType(IOC_DATATYPE_STRING, _T("")), _T(""), _T(""), _T(""),
      eVAR_ATTR_NONE));
   return TRUE;
}
```

CHAPTER 10

Writing The Execution Method for a Runtime Plug-In (Criteria Data)

Writing the execution method that uses criteria consists of writing one or more of the following methods:

- ExecuteObjectQuery for a Business Interlink Query. This method extracts criteria from the Interlink Object, and adds outputs to it.
- ExecuteObjectUpdate for a Business Interlink Update. This method extracts inputs and criteria from the Interlink Object.
- ExecuteObjectDelete for a Business Interlink Delete. This method extracts criteria from the Business Interlink Object.

For more information about the methods that are mentioned here, refer to Understanding The Business Interlink Methods.

When you write the execution method, you will use methods to perform the following tasks:

The execution method that you write performs the following tasks. You may or may not do these steps in the following order, depending upon how you wish to write your execution method.

1. Takes as input a Business Interlink Object.
2. Use the GetObjName method to extract the name of the Business Interlink Object. The name of the Business Interlink Object is the name of the transaction, and you use it to determine which transaction you are writing.
3. Use the GetConfigParameters method to extract the configuration parameter values for this Business Interlink Object, and performs whatever actions are needed with them (usually configuration parameters are needed to connect to the external system).
4. Use the GetOutputTable method to get the output table for the Business Interlink Object. Use the Clear method to clear the output table.
5. Use the GetOutputParams method to get the output parameters, and then use the AddColumn method to add one column to the table for each output parameter.

6. If you are writing ExecuteQuery or ExecuteUpdate, rather than extract input values from the input table, you extract the criteria from the criteria structure instead of performing steps 7 and 8.

7. If you are writing ExecuteTransaction, ExecuteObjectAdd, or ExecuteObjectDelete, use the GetInputTable method to get the input table for the Business Interlink Object, and then use the ResetCursor method to reset the cursor to the top of the input table.

8. Use the GetInputParams method to get the input parameters, and then use the GetNextRow method to extract one row of input parameters.

9. Pass your inputs (or the criteria) to the external system, calling functions in the external system to perform the actions that you want your Business Interlink Object to take, and receive the outputs from your external system.

10. Use the push_back method to insert the output values into the Business Interlink Object output table.

Understanding Business Interlink Object Criteria

Criteria are used with the object query Business Interlink Object and the object update Business Interlink Object; the method you write is ExecuteObjectQuery or ExecuteObjectUpdate. For example, a query may resemble the following:

```
Select output_list from class
   where criteria_rows
```

output_list	a list of the outputs for this query.
class	the class upon which this query is being performed.
criteria_rows	rows of criteria for this query.

A common way for you to build a query within the ExecuteObjectQuery method is to create a long character string that forms one query statement. To build that statement, you will need to get the following information from the Business Interlink Object that is the input for your ExecuteObjectQuery method:

- The criteria rows. Each row consists of a left-hand-side input, a relational operator, a right-hand-side input, and a logical operator. The rows can also be grouped, as in the following example. Get the rows and the grouping with the InterfaceObject criteria structures and methods.

  ```
  Select output from class where A > 5 AND (B < 4 OR C != 10)
  ```

  The rows B < 4 OR and C != 10 have been grouped.

- Outputs. Get the outputs, or the members of the class chosen for output, with the InterfaceObject method GetOutputParams.

- Class. Get the name of the class upon which the query is being performed with the InterfaceObject method GetObjName.

To understand the criteria rows, and how their data is stored, you should understand the CriteriaRow structure and the CriteriaNode structure.

Understanding the CriteriaRow Structure for Criteria Rows

A criteria row consists of the following:

```
left-hand-side relational-operator right-hand-side logical-operator
```

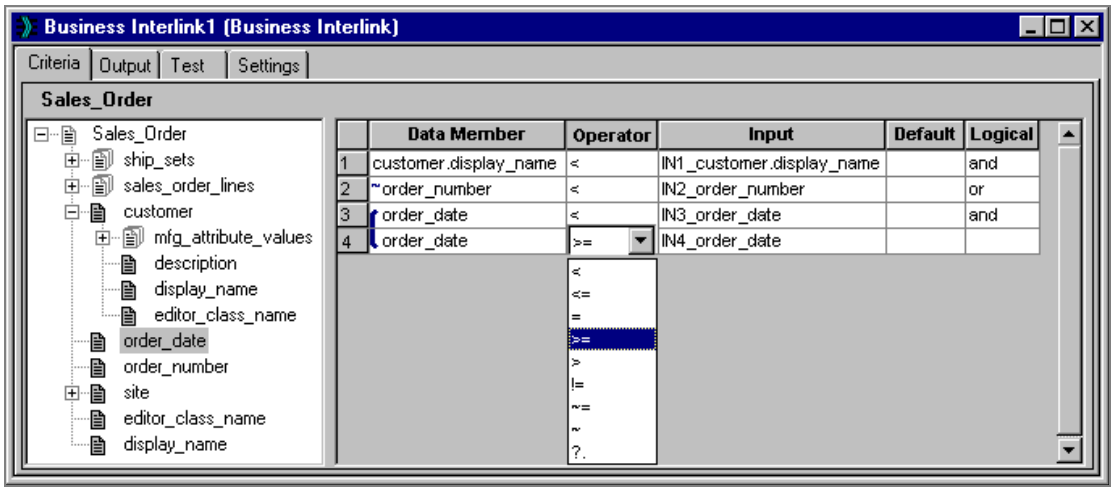
The CriteriaRowList structure consists of a list of CriteriaRow structures. The CriteriaRowList and CriteriaRow structures are included with the query Business Interlink Object that is input for the ExecuteObjectQuery method. The CriteriaRow structure consists of a criteria row, and has the following form.

```
int rowNum;
PSIOString lOper;      // logical operator
PSIOString lhsExpr;    // left hand side
PSIOString rOper;      // relational operator
PSIOString rhsExpr;    // right hand side
PSIOString rhsDefault; // default value of right hand side
PSIOString dataType;   // the data type
```

Each CriteriaRow structure in the CriteriaRowList is filled within the Application Designer, when a user designs a query Business Interlink Definition and creates the associated PeopleCode program. For example, the following query Interlink Definition designed in the Application Designer on a Sales Order class would produce a query Business Interlink Object that has a CriteriaRowList structure with four CriteriaRows.

The left-hand-side consists of members of the class upon which the query is being performed. An Application Developer, when creating a Business Interlink Definition within the Application Designer, selects these members from the class. The right-hand-side (IN1_customer.display_name, IN2_order_number, IN3_order_date, and IN4_order_date) is replaced within PeopleCode with constants, or with names of PeopleCode variables or Record Fields or other legitimate names. The Application Developer, when writing a PeopleCode program for this Business Interlink Definition, selects the constants, PeopleCode variables, etc.

```
Select Sales_Order
  where
    customer.display_name = IN1_customer.display_name AND
    NOT order_number < IN2_order_number OR
      (order_date < IN3_order_date AND
       order_date >= IN4_order_date)
```



Business Interlink Page: Criteria Tab

Understanding the CriteriaNode Structure for Negation and Grouping

The CriteriaNode structure is also included with the query Business Interlink Object that is input for the ExecuteObjectQuery method.

The CriteriaNode structure consists of pointers to CriteriaRows in a CriteriaRowList, and also can contain pointers to another CriteriaNode. The CriteriaNode structure has the following form:

```
int type;
BOOL negation;
Union UNION
{
   CriteriaGroup *pGroup;
   Int row;
} data;
```

*pGroup is a pointer to a CriteriaNodeList. row is a number pointing to a CriteriaRow in the CriteriaRowList structure. To tell you which of these is being used for this Criteria Node, the class CriteriaGroup contains an enum of the form:

```
enum { ELEMENT, GROUP};
```

ELEMENT says that this node uses the pointer to a CriteriaRow, and GROUP says that this node uses the pointer to a CriteriaNodeList.

Example of CriteriaRowList and CriteriaNodeList

There is a CriteriaRowList structure that contains all of the CriteriaRows. In addition, there is a CriteriaNode list structure that contains at least one CriteriaNodeList containing

CriteriaNodes that point to CriteriaRows, plus an additional CriteriaNodeList to point to the CriteriaRows (and CriteriaNodes) contained in each group.

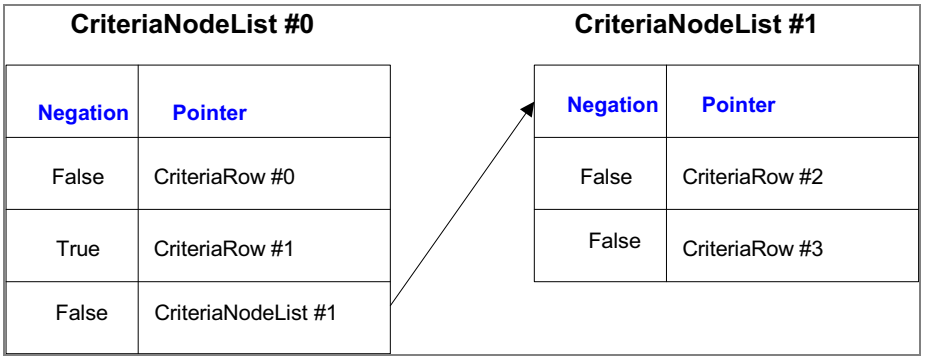
A Business Interlink Object for the following query has four CriteriaRows in its CriteriaRowList. The object also has two CriteriaNodeLists, because there is one grouping within this query.

```
Select Sales_Order
  where
    customer.display_name = "Harvey" AND
    NOT order_number < "101" AND
      (order_date < "1999-10-20" OR
       order_date >= "1999-08-20")
```

The query Business Interlink Object contains a CriteriaRowList with four CriteriaRows.

CriteriaRow #	*Left hand side lhsExpr*	*relational operator rOper*	*Right hand side rhsExpr*	*Logical operator lOper*
0	customer.display_name	=	"Harvey"	AND
1	order_number	<	"101"	AND
2	order_date	<	"1999-10-20"	OR
3	order_date	>=	"1999-08-20"	

The query Business Interlink Object contains two CriteriaNodeLists, each containing CriteriaNodes that point to CriteriaRows or to CriteriaNodeLists. The following diagram shows the structure of the CriteriaNode Lists for this Business Interlink Object. Each node in the CriteriaNodelists have a negation: the node pointing to CriteriaRow #1 has a negation. CriteriaNodeList #0 contains pointers to CriteriaRow #0 and CriteriaRow #1, and to CriteriaNodeList #1. CriteriaNodeList #1 contains the grouped CriteriaRows #2 and #3.



CriteriaNodeLists

Understanding m_CriteriaGroup

m_CriteriaGroup is a class of type CriteriaGroup. It contains all the CriteriaRow and the CriteriaNode List structures for this Business Interlink Object.

Building A Query String Using The Criteria Methods and Structures

If your execution method is either ExecuteObjectQuery or ExecuteObjectUpdate, you will use the criteria structures. The most common way to create a query or update is to create a long character string, fill the string with the criteria and other text for a query or update, and to send that string to your external system.

This section will use a character buffer of type TCHAR to contain this string:

```
TCHAR buf[512]
```

Adding the Outputs to the Query String

Use the InterfaceObject method GetOutputParams to return the output names for this query Business Interlink Object. Then retrieve the output names and use them to build the output names in your query.

For more information on GetOutputParams, see GetOutputParams, GetOutputParam, GetOutputParamCount.

The following example code places the string “select” into the output table, retrieves the output names, and then puts the output names into the output table, separating each output name with a comma.

```
_tcscpy(buf, _T("select "));

const VARINFOLIST& varListOutput = IntObj->GetOutputParams();

for(int i = 0; i < varListOutput.size() - 2; i++)
{
   _tcscat(buf, varListOutput[i]->m_strName.c_str());
   if(i != varListOutput.size()-3) _tcscat(buf, _T(", "));
}
```

_tcscpy is a function that copies a character string into a character string buffer. _tcscat is a function that concatenates two character strings.

VARINFOLIST is a list of varListOutputs; a varListOutput is a character string containing one output name. The size VARINFOLIST method gets the number of output names in this list of output names from GetOutputParams. The c_str PSIOBuf method gets a UNICODE string out of varListOutput; the c_astr PSIOBuf method gets an ASCII string.

For more information on the VARINFOLIST object, see Parameter Lists.

For more information about the size method, see size.

For more information about the c_str and c_astr methods, see c_str and c_astr.

Adding the Class Name to the Query String

Suppose you have already created, within a TCHAR buf[512] character buffer, the character string "select *output1, output2*", where *output1, output2* is a comma-delimited character string of all the output names for this query. For this query, you want to add to the buffer the string "from *class*", where *class* is the name of the class upon which this query will be performed. The name that you want to use is the name of this Business Interlink Object; use the GetObjName InterfaceObject method to get it. The following code will add "from *class*" to the buffer.

```
_tcscat(buf, _T(" from "));
_tcscat(buf, IntObj->GetObjName().c_str());
```

For more information on using GetObjName, refer to GetObjName, ObjName.

Adding the Criteria Rows and Groupings to the Query String

You will use the CriteriaGroup methods GetRows() and GetGroup() to get the CriteriaRows, and then you will use the CriteriaRow structure to extract the left-hand-side, relational-operator, right-hand-side, logical-operator) from the CriteriaRows.

Getting The Criteria Rows: GetRows

Call the GetRows method to get a criteria row.

```
CriteriaRowList* rowlist = IntObj->m_CriteriaGroup.GetRows()
```

IntObj	the InterfaceObject pointer that is input to this ExecuteTransaction method.
rowlist	a pointer of type CriteriaRowList. Points to a CriteriaRow within the CriteriaRowList structure.

For example, the following code calls a routine named GenerateCriteria, which has two inputs: the address of the first CriteriaNodeList structure, and a call to GetRows, which in this case, returns a pointer to the first CriteriaRowList structure.

```
GenerateCriteria(strCriteria, &IntObj->m_CriteriaGroup,
   IntObj->m_CriteriaGroup.GetRows());
```

Getting the Criteria GetGroup()

Call the GetGroup method to get a criteria node.

```
CriteriaNodeList* nodelist =
   &IntObj->m_CriteriaGroup.GetGroup()
```

IntObj	the InterfaceObject pointer that is input to this ExecuteTransaction method.
nodelist	a pointer of type CriteriaNodeList. Points to the first Criteria Node in the CriteriaNodeList.

For example, the following code shows a call to a routine named GenerateCriteria, which has two inputs: the address of the first CriteriaNodeList structure, and a call to GetRows, which in this case, returns a pointer to the first CriteriaRowList structure. Then the GenerateCriteria routine uses the address to the first CriteriaNodeList structure to call GetGroup, getting the first CriteriaNode within the CriteriaNodeList structure.

```
GenerateCriteria(strCriteria, &IntObj->m_CriteriaGroup,
   IntObj->m_CriteriaGroup.GetRows());
```

```
void RecordDriver::GenerateCriteria(PSIOString& strCriteria,
   CriteriaGroup* Group, CriteriaRowList *rowList)
{
   CriteriaNodeList *gList = Group->GetGroup();
```

Getting the Elements From The Criteria Rows

Each criteria row consists of a left-hand-side input, a relational operator, a right-hand-side input, and a logical operator (except for the last criteria row in the CriteriaRowList structure, which contains no logical operator).

The following example method, GenerateCriteria, returns a PSIOString character string, strCriteria. strCriteria contains all of the criteria for the query, the relational and logical operators, and the groupings.

GenerateCriteria has two inputs: a pointer to a CriteriaNodeList structure, and a pointer to a CriteriaRowList structure.

Comments within the code below explain how this example method parses the CriteriaNodeList structures, the CriteriaRow structures, and builds the character string from the elements in each CriteriaRow structure.

```
void RecordDriver::GenerateCriteria(PSIOString& strCriteria, CriteriaGroup*
Group, CriteriaRowList *rowList)
{
   // Get the pointer to a CriteriaNodeList.
   CriteriaNodeList *gList = Group->GetGroup();
   PSIOString op;
   // Loop for the number of CriteriaNodes in the CriteriaNodeList
   for(int i = 0; i < gList->size(); i++)
   {
      // If this CriteriaNode contains a pointer to a CriteriaRow,
      // process the CriteriaRow
      if((*gList)[i].type == CriteriaGroup::ELEMENT)
      {
         // Get the pointer to the CriteriaRow
         CriteriaRow* cRow = (*rowList)[(*gList)[i].data.row-1];
         // Add the logical operator for the previous CriteriaNode
         //  to strCriteria.
         strCriteria += op;
         // If this CriteriaNode has a negation,
         //  add NOT ( to strCriteria.
         if((*gList)[i].negation) strCriteria += _T(" NOT (");
         strCriteria += _T(" ");
         // Add the left-hand-side to strCriteria
         strCriteria += cRow->lhsExpr;
         strCriteria += _T(" ");
         // Add the relational operator to strCriteria
         strCriteria += cRow->rOper;
         strCriteria += _T(" ");
         // If the dataType of this CriteriaRow is string, add ' to
         strCriteria
         if(cRow->dataType == _T("string")) strCriteria += _T(" '");
         // Add the right-hand-side to strCriteria
         strCriteria += cRow->rhsDefault;
         // If the dataType of this CriteriaRow is string, add ' to
         strCriteria
         if(cRow->dataType == _T("string")) strCriteria += _T("'");
         // If this CriteriaNode has a negation,
         //  add ) to strCriteria; with the previous NOT(, this
         //  encloses the node within NOT().
         if((*gList)[i].negation) strCriteria += _T(") ");
         strCriteria += _T(" ");
         // Store the logical operator in the op buffer.
```

```
            op =  cRow->lOper;
         }
         // Else this CriteriaNode contains a pointer to a
         //  CriteriaNodeList; process the CriteriaNodeList
         else
         {
            // Add the logical operator for the previous CriteriaNode
            //  to strCriteria.
            strCriteria += op;
            // If this CriteriaNode has a negation,
            //  add NOT to strCriteria.
            if((*gList)[i].negation) strCriteria += _T(" NOT ");
            // Add ( to strCriteria to start this CriteriaNodeList
            strCriteria += _T(" (");
            // Call GenerateCriteria, using (*glist)[i].data.pGroup as
            // the address of the CriteriaGroup, and using the same
            // pointer to the CriteriaRowList structure that was
            // originally passed into GenerateCriteria.
            GenerateCriteria(strCriteria, (*gList)[i].data.pGroup,
               rowList);
            // Add ) to strCriteria to end this CriteriaNodeList
            strCriteria += _T(") ");
         }
      }
   }
```

CHAPTER 11

Configuring PSINTERLINKS as a WebApp on WebSphere

To configure Business Interlinks for WebLogic and WebSphere, update the PATH in C:\Apps\WebSphere\AppServer\bin\startServer.bat file. Add this to the PATH:

```
%WAS_HOME%/installedApps/peoplesoft/PSINTERLINKS/Web-inf/bin/winx86
```

The SET PATH then resembles:

```
SET PATH=%WAS_HOME%/installedApps/peoplesoft/PSINTERLINKS/Web-
inf/bin/winx86;%WAS_HOME%/bin;%JAVA_HOME%/bin;%JAVA_HOME%/jre/bin;%PATH%
```

You can backup the startServer.bat file into a PS-startServer.bat and then update the startServer.bat file with PATH changes.

To ensure that PSINTERLINKS is installed successfully and works

1. Cut and paste the HTML section (as below) into a file named test.html:

```
<html>
<head>
<title>Test BusInterlinkServLet</title>
</head>
<body bgcolor="#FFFFFF" text="#000000">
<center>
<br>
<blink><h1><b><i><font color="ff00ff">Test
BusInterlinkServLet</font></i></b></h1></blink>
<br>
</center>
<h1><b>Set a Home directory for all your plug-in's</h1><b>
<form method=get
action="http://your_web_server/PSINTERLINKS/BusInterlinkServLet">
<input type="text" size=30 name=homedir value="">
<br>
<br>
<input type="submit" value="Ok">
<input type="reset" value="Clear">
</form>
```

```
<h1><b>Test PsioloaderJ</h1><b>
<form method=post
action="http://your_web_server/PSINTERLINKS/BusInterlinkServLet">
<br>
<br>
<input type="text" size=30 name=xml value="test">
<input type="submit" value="Ok">
</form>
</body>
</html>
```

2. Modify *your_web_server* to be your web server name and port number, and save the test.html file.

3. Double click on test.html.

4. In the first Text box, put in the path containing the Business Interlink supporting libraries.

   For example, on WebSphere it is:

```
C:\APPS\WebSphere\AppServer\installedApps\peoplesoft\PSINTERLINKS\Web-
inf\bin\winx86
```

5. Click on the first "Ok" button.

   The first Text box and OK button tests the HTTP "GET" functionality of the servlet BusInterlinkServLet. If it works, means the BusInterlinkServLet servlet is installed correctly.

6. Click on the second "Ok" button.

   The second Text box and OK button tests the HTTP "POST" functionality of the servlet BusInterlinkServLet. If it works, it means the "POST" functionality works. It also loads up the BI supporting libraries successfully.

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 Manangement, 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 statutroy 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 "breadcumb 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 peformed using PeopleSoft Budget Planning. Budget activities include Line Item Budgeting, Line Item Mass Adjustments, Budget Allocations, and Position Budgeting. PeopleSoft Budget Planing-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 knows 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: (Incumbent Pay/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: range midpoint/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 it's 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 it's 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 aquisition 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 the 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 EE0 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 it's business enterprise. The term includes financial, manufacturing and supply chain, human resources, and payroll applications, among others.

Enterprise Warehouse (EW)

See PeopleSoft Enterprise Warehouse

Entry Authority

Authorization granted by employees to specific user IDs for entering expense data on their behalf.

Entry Currency

The currency used to enter budget data.

Entry Event

An automated process that generates multiple debits and credits resulting from single transactions, to produce standard supplemental accounting entries.

Entry Event Code

Designation of an Entry Event; an identifier or label.

Entry Event Generator

A mechanism that generates standard, supplemental accounting entries based on Entry Event codes.

Entry Event Process

An accounting transaction. Entry Event processes combine to form Entry Events. For example, requisition posting is a Purchasing process, and cash clearing is a Payables process. Each process can involve one or several Entry Event Steps.

Entry Event Step

Part of an accounting transaction. For example, the BUDG process includes these steps, among others: prepare allotment budgets, prepare organization budgets, and prepare revenue estimates. Entry Event steps combine to form Entry Event processes.

Entry On Duty Date (EOD)

Date that indicates when an employee started to work at his/her current agency.

Entry Type

Any activity that creates or updates an item.

EPM (Enterprise Performance Management)

See PeopleSoft Enterprise Performance Management

Equal Employment Opportunity Commission (EEOC)

In the United States the EEOC requires that most companies file one or more reports from a series named EEO-1 through EEO-9. These reports include counts by federal employment categories of male and female employees in certain ethnic groups.

Equitization

A process that enables parent companies to calculate the net income of subsidiaries on a monthly basis and adjust that amount to increase the investment amount and equity income amount before performing consolidations. For organizations with complicated parent/subsidiary business unit relationships, this automated process reduces the process time and reduces the possibility for errors.

Equity Increase

In PeopleSoft Workforce Analytics, Equity Increases are base pay increases granted to bring an employee's pay up to some internally specified standard for your organization.

Equivalent Standard Deviation

In PeopleSoft Demand Planning, a **Standard Deviation** developed during the Model Reset process that enables you to compare standard deviations from different models. The deviation is calculated by multiplying the **Model Equivalency Factors** defined on the **Control Group** by the standard deviation.

ERISA (Employee Retirement Income Security Act of 1974)

The U.S. Federal legislation enacted to prevent abuses of employee pension rights by employers.

Error Exception

A transaction that is stopped because the budget limits would be exceeded if it continued. For the transaction to proceed, action must be taken, such as canceling or reducing the transaction amount, increasing the budget amount, overriding the budget limits, or transferring available funds from another budget.

Error Ratio

The ratio of the statistical Standard Deviation to the base component that gives an indication of the accuracy of the forecast. The ratio is presented in the PeopleSoft Demand Planning Audit and Accuracy Review and is calculated during the **Forecast Calculation Process**.

Estimated Gross

Estimated labor cost associated with reported time (see Managing Time, Understanding time Reporting Codes).

Estimated Shipments

A group of shipment schedules used to manage sales order requested shipment data and actual shipment data. Used in conjunction with weight and volume pricing and freight charge calculations.

Ethnic Code

The Federal Office of Management and Budget (OMB) racial and ethnic census categories used for classifying individuals in U.S. Government reports.

ETL (Extract-Transform-Load)

See Extract-Transform-Load.

ETL maps

ETL maps provide rules for importing your source data to the data warehouse tables.

Evaluated Receipts Settlement (ERS)

A PeopleSoft Payables feature that matches receipts against purchase orders and generates vouchers without requiring an invoice.

Evaluations Periods

In PeopleSoft Demand Planning, indicates the number of future periods to use for the calculation of forecast errors. For example, if the number of evaluation periods is two, then the forecast error in April 2001 (after posting demand for April) would be determined by comparing the actual demand for April and the April forecast generated in February 2001. Evaluation periods are set separately for each view.

Event

Events are predefined points either in the Application Processor flow or in the program flow. As each point is encountered, the event fires on each component, triggering any PeopleCode program associated with that component and that event. Examples of events are FieldChange, SavePreChange, OnRouteSubscription, and so on.

Event

Occurrence or happening.

Event Class

An event or type of event that results in a change of benefits eligibility for an employee or dependent. Event classes are prominently used in COBRA and Benefits Administration processing.

Event Maintenance

The process that enables you to manage ongoing enrollments during a plan year. Changes involving maintenance include new hires and re-hires, terminations, family status changes, and changes to benefits eligibility.

Event Rule

Used by PeopleSoft Benefits Administration to determine how events are processed by the system. Event rules look at the benefit plan options an employee is eligible for and determine which options the employee can actually *choose.* Event rules are closely associated with eligibility rules but it is important to note that they are not the same. Event rules *should not* be used to determine eligibility.

Event Trigger

You use triggers to tell the system that when a change takes place to certain data (an event), it should perform an action automatically. When the event occurs, the system writes a line to a trigger table. Then when it's time for the action, the system reads the data from the trigger table and performs the appropriate action.

EW (Enterprise Warehouse)

See PeopleSoft Enterprise Warehouse.

See also PeopleSoft Enterprise Performance Management (EPM).

Expected Losses

In the financial services industry, the amount the institution predicts it will lose in portfolio value. Loan loss reserves are set aside to cover the expected losses.

Excepted Service

As defined by 5 USC 2103, the Excepted Service consists of those civil service positions that are not in the competitive service or Senior Executive Service.

Exception

User or system delivered, defined conditions applied to scheduled, reported or payable time that require audit or review (see Time Management)

Exception Rules

A rule (s) that is applied to scheduled, reported time, and payable time in order to determine conditions which require audit or review (see Time Administration).

Exception Severity

The degree of importance associated with an exception. For example, in exception which is a result of an employee clocking in late may have a Medium severity, while an exception which is a result of an employee not clocking in has a High severity (see Time Management).

Exception Time Reporting

A method of time reporting where only differences to the schedule are provided (see Time Reporting).

Excess Plan

A pension plan where the benefit formula provides an increased benefit for Final Average Earnings above a specified integration level. This compensates for the fact that Social Security benefits are based only on earnings up to a specified maximum.

Exchange Rate Variance

In PeopleSoft Cost Management, the change in currency exchange rate between the time the item is received into inventory and vouchered in Accounts Payable.

In PeopleSoft Payables, a matching feature that compares the exchange rate on the purchase order and the invoice and then copies any variance to PeopleSoft Inventory tables for analysis and accounting purposes.

Exclusive Pricing

Supersedes all pricing structures in effect for customers and products, except **Buying Agreement**, and enables you to drive pricing with a promotional structure. Exclusive pricing can be set up for a specific time frame and associated with particular orders.

Executive Schedule (EX)

Compensation and pay plan used by the Executive Branch of the federal government. Statutory pay limits are derived from several of the pay levels within this plan and imposed on the General Schedule and other existing pay plans throughout the Federal government.

Exercisable

The option shares that are available to the optionee to exercise.

Exercise

The transaction in which an individual purchases or "exercises" the right to purchase the option shares. The IRS refers to the purchase of company stock in an employee stock purchase plan as an exercise.

Exercise Date

The date on which an individual purchases underlying shares from and option grant or transacts a simultaneous purchase and sale of underlying option shares through a cashless exercise and collects option profit in cash or shares.

Exercise Price

The price per share required to exercise a stock option.

Exercise Proceeds

Cash, stock or other recognition received by a company as a result of option exercises, including cash or stock paid by individuals to exercise options and cash company tax savings from deducting non-statutory option profits at exercise.

Expense Location

Geographic area defined to enable the recording, tracking, and reporting of expense activity.

Expense Location Amount

Authorized spending defined for an expense type in a particular expense location and currency.

Expense Location Group

Collection of expense locations based on a common classification such as state, country, or continent.

Expense Type

Means of itemizing various kinds of business expenses. Examples are hotel, dinner, or ground transportation.

Expense Type Edit

User-defined requirement that mandates input of additional data—such as an airline ticket number or number of nights in a hotel—when an expense type is selected in an expense report.

Expense Type Group

Expense types that are classified together for reporting and tracking.

Expensed Item

Non-inventory item which may represent software, manuals, documentation, or any item for which no quantity on hand is maintained, but which can be specified on a bill of material (BOM). Expensed items can only exist as components on a BOM and cannot have a BOM, routing, or production area/item definition.

Expiration

The process by which the outstanding shares of an option cease to be exercisable, generally at the end of the option term. The length of the option term and the date of expiration are established in the Grant Agreement.

Expiration Date

In PeopleSoft Inventory, the date a lot exceeds its Shelf Life and is no longer acceptable for fulfillment or consumption. (Expiration Date = Creation Date + Shelf Life)

Expiration Date

The last day of an option term in which the option is canceled and no longer exercisable.

Expiration Grace Period

When you enter a stock action allows the exercise of the already vested shares as of the action date, the system will calculate the date these shares expire based on the grace period defined on the Stock Action Rules page for that stock action. The system will automatically cancel vested shares not exercised at the end of the expiration grace period.

Express Customer

A customer for whom the minimum necessary information is entered.

Expressions

Expressions enable you to create pseudo-columns made up of mathematical calculations based on actual fields on a table. Since expressions are resolved at run-time, duplicate information is not stored on the database.

Express Order

An order entry shortcut in PeopleSoft eStore and Mobile Order Management whereby the customer populates the shopping cart and goes directly to the order summary to checkout, bypassing any billing or shipment modification screens. Billing and shipping information defaults in as previously entered.

External Data

Data from external sources. For instance, in PeopleSoft Workforce Analytics, external data may include third party salary surveys and benchmark metric surveys.

External Scheme

In the United Kingdom an External Scheme is a vocational training, education and job placement program involving an employee, an employer and the government.

External System

Any system that is not directly compiled with the PeopleTools servers.

Extra Time

Any hours worked outside of an employee's normal (scheduled/shift) hours or days. Extra time may be scheduled in advance of when it is worked, and may be subject to special compensation rules. It may be treated differently than standard time for purposes of Benefit Entitlement and Administration.

Extraction

A reusable query that specifies what information should be retrieved from the Quality database.

Extract-Transform-Load (ETL)

The extraction and transport of data from one server to another remote server. In PeopleSoft budgeting ETL specifically refers to the process by which financial and human resource data is extracted from PeopleSoft Financials and HRMS and transferred to the PeopleSoft Enterprise Warehouse which PeopleSoft Budgeting uses to access and record data transactions. Within PeopleSoft Enterprise Warehouse, data migration typically refers to information moved from outside sources into the Operational Data Store tables.

Extrinsic Rewards

Tangible rewards that can be given to the individual. Typically categorized as financial and non-financial rewards. Financial rewards would include direct compensation, indirect compensation and deferred compensation. Non-financial rewards are provided to the individually and viewed as a benefit by the individual based on the culture of the organization such as the size or location of one’s office.

In PeopleSoft Workforce Analytics, tangible rewards given to an individual. Typically categorized as financial and non-financial rewards. Financial rewards would include direct compensation, indirect compensation and deferred compensation. Non-financial rewards are provided to the individually and viewed as a benefit by the individual based on the culture of the organization such as the size or location of one’s office.

F

Fact

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

Fact Table

A fact table is where facts are stored in the PeopleSoft Enterprise Warehouse.

Fair Labor Standards Act (FLSA)

A federal regulation governing several time and labor issues. *FLSA Overtime* requires that all nonexempt employees be paid at a rate of time-and-one-half for all hours over 40 physically worked during a workweek. This requirement may be superseded by state or local laws when the lesser law is to the greater benefit of the employee, or by union contract. An *FSLA Workweek* is a permanently established, regular workweek for a group of employees.

Fair Market Value (FMV)

The price of a company stock based on the current market value as determined by supply and demand, or a valuation method. The stock market sets the fair market value for a public company. For a private company the fair market value is more subjective, but typically determined by financial factors or set by an outside valuation company.

Fair Market Value Tracking Methods

Methods used to track and report trading activity on various exchanges (i.e. NYSE, AMEX, NASDAQ, etc…).

Family Medical Leave Act (FMLA)

A federal regulation that protects health benefits and job restoration for employees who must take a leave from work to care for themselves or family members. FMLA regulations contain provisions regarding employer coverage, employee eligibility and entitlement, notice and certification, continuation of health benefits, and job restoration. PeopleSoft Benefits applications offer FMLA Plans that help employers and employees determine FMLA eligibility and schedule and track FMLA leave requests.

Federal Employee Group Life Insurance Program (FEGLI)

Generally, if the employee has Federal retirement coverage or is on a temporary appointment exceeding one year, he/she is eligible to participate in the FEGLI program. Once eligible, he/she is covered automatically for Basic Life Insurance and premiums will be deducted from gross salary unless coverage is waived within the first period of eligibility. The program offers Basic Insurance coverage and three types of optional coverage: Option A (Standard), Option B (Additional), and Option C (Family).

Federal Employee Pay Comparability Act (FEPCA)

This law provides a structure and methodology to determine and authorize locality-based pay adjustments to Federal employees in order to elevate their basic pay to be commensurate with private sector employees working in the same occupations in the same geographic localities. It also includes a feature to authorize agencies to make advance salary payments to attract candidates for open positions which have consistently been hard-to-fill in certain geographic areas.

Federal Employees' Compensation Act (FECA)

This law provides compensation and medical benefits to civilian employees of the United States for disability due to personal injury or disease sustained while in the performance of duty. A feature of this law provides for the continuation of pay (COP) without charge to leave for up to 45 calendar days due to disability and/or medical treatment following a traumatic injury. Employees file claims with the U.S. Department of Labor, Office of Worker's Compensation, which adjudicates the claims and compensates the employing agencies for the employee's pay and benefits during the claim period.

Federal Employees Health Benefits (FEHB)

Generally, the employee is entitled to coverage by the FEHB program if appointed to a position with Federal retirement coverage or has been on the rolls on a temporary appointment for more than one year. The Federal employer shares the cost of the premium (about 75%); actual premiums depend on the plan selected. If under a temporary appointment, the employee pays both the employer and employee shares. If the position is part-time, the employee pays the employee share and a portion of the employer’s share.

Federal Employees Retirement System (FERS)

A retirement plan available to employees of the federal government. FERS covers all employees appointed to a position in the federal government after January 1, 1987. Coverage includes Social Security, a basic annuity plan, and a TSP.

Federal Employer Identification Number (EIN)

Used to identify the tax accounts of businesses. Businesses, which have employees or operate business as a partnership or corporation, must obtain an EIN.

Federal Insurance Compensation Act (FICA)

Employee and employer contributions to Social Security.

Federal Reserve Transit Number

A unique identifier for U.S.-based banks, allowing banks to transfer funds within the Federal Reserve system.

Feeder Line

A type of production line replenishment used in PeopleSoft Flow Production. If you are using feeder line replenishment, smaller production lines create subassemblies that feed directly to your production line.

FEGLI Living Benefits Act

Beginning 7/25/95, a Federal employee who is terminally ill may elect to receive a lump-sum payment equal to the full amount of basic life insurance only, or a limited portion designated in multiples of $1000. An election to receive this benefit is irrevocable; the individual is considered terminally ill if his /her life expectancy is 9 months or less.

FICA (Federal Insurance Contributions Act)

FICA consists of both a Social Security (retirement) payroll tax and a Medicare (hospital insurance) tax. The tax is levied on employers, employees, and certain self-employed individuals.

Fictitious Calculations

Fictitious calculation rules perform temporary calculations. A fictitious calculation is a sub-calculation run during a normal calculation to determine a net that would have been computed if certain parameters were used. This result is then used for further processing in the normal calculation. A fictitious calculation is always started from inside a normal calculation, run for one payee, and run for a specified set of periods.

FIFO (First In First Out)

Method used by companies to record Disqualifying Disposition Income. If a company uses this method they record the optionees disposition of shares by attributing the shares to the earliest exercise, purchase or release dates for which shares remain available for sale.

Fill-In Employment

Employment held by persons during the time period after leaving their regular occupation in anticipation of, but before entering, military service.

Filter

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

Final Average Earnings (FAE)

The PeopleSoft Pension Administration function that averages earnings from a specified period of an employee's career. The result is used as a component of the pension benefit formula.

Final Forecast

The final forecast is the prorated version of the adjusted forecast, summarized to all levels of the product hierarchy. This is the best-guess version of the forecast that is used to make all decisions dependent on the forecast.

Final Table Merge Engine

Final Table Merge Engine is used by the PeopleSoft Enterprise Warehouse; it moves enriched data from one table into another. When you run a job in a jobstream, the immediate results are stored in temporary tables. At the end of the jobstream, the Merge Engine runs and merges the output temporary tables into the final tables, where processing can continue.

Financial Accounting

The accounting for a business entity's assets, liabilities, revenues, and expenses to determine its net worth and to produce financial statements. Within Generally Accepted Accounting Principles, a business has some latitude as to when and how to record its financial transactions, as long as it continues to meet its legal and regulatory requirements. A business' financial accounting requirements are not necessarily the same as its cost accounting requirements. The one should not be mistaken for the other (i.e. the extent to which a company's financial accounting system meets its cost accounting needs depends on how it has chosen to describe its chart of accounts and the level at which it has chosen to record financial transactions.

Financial Instrument

In the financial services industry, a specific product or service sold by a financial institution to its customers. In terms of the reporting hierarchy, a product falls under a ledger account, while an instrument falls under a product. A product may be treated like a generic description or category, while an instrument is a specific instance of a category.

Financial Performance Measures (FPM)

For the financial services industry, the Financial Performance Measures program performs calculations on financial instruments based on the rules defined in the Financial Calculation Rules module, and using input from the Cash Flow Generator, Stratification engine, Product Pricing, and Curve Generator. Its calculations include: measures of duration, option-adjusted spread and option-adjusted cost for PeopleSoft Funds Transfer Pricing, and Monte Carlo simulation for PeopleSoft Asset Liability Management.

Financial Product

In the financial services industry, a product or service sold by a financial institution to its customers. In terms of the reporting hierarchy, a product falls under a ledger account, while an instrument falls under a product. A product may be treated like a generic description or category, while an instrument is a specific instance of a category.

Financial Services Instrument

In the financial services industry, products created by financial institutions and sold to retail customers. Product prices and interest rates are set by the financial institutions and take into account its customers' behavioral models.

Financial Statement Simulation

A facility within Planning & Simulation which establishes rules for simulating future period, or pro-forma, financial statements. The user defines corporate financial policies, such as corporate tax rates, dividend distribution frequency, and force balancing rules, which are then applied to cash flows for a given future accounting period. The Financial Statement Simulator engine drives costs and revenues to accounts on PF_LEDGER_F00 via a scenario.

Financials Warehouse

See Warehouses.

First Year Amount

See 1st Year Amount.

Fixed Basis

The basis option enables you to create the data for the Basis online, as part of the rule. Fixed Basis is used with the Allocation Manager only. It is available with all methods except when Period-Based Allocation is being used. The Fixed Basis is a predetermined table that can be populated online.

Fixed Offering

The offering type is fixed when the end date of each offering is the same for all employees regardless of the employee's grant dates.

Fixed Percentage

A fixed percentage value. The source pool amount will be split based on this percentage to get the target amount. Used with the Allocation Manager.

Fixed Period Requirements

In PeopleSoft Enterprise Planning and Production Planning, a lot-sizing technique that sets the order quantity to the demand for a given length of time.

Fixed Picking Bin

A dedicated picking location for an inventory item. Fixed picking bins are replenished from bulk locations when the available quantity falls below the optimal quantity.

Fixed Plan

A stock purchase offering period where the ending offering date will be the same as the purchase date. Eligible employees will always purchase stock on the specific purchase dates and by the purchase rules you define.

Fixed Quantity

An **Inventory Policy** method that defines a fixed amount of an item to be ordered to meet replenishment needs. This method can be selected as an inventory policy for order quantity, safety stock, **Reorder Point**, and minimum and maximum parameters.

Fixed Source

The fixed source option enables you to create the data for the Source online, as part of the rule. Fixed Source is used with the Allocation Manager only. It is available with all methods except when Period-Based Allocation is being used. The Fixed Source is a predetermined table that can be populated online.

Flexible Credit

Any credit associated with a given benefits program, plan, or type of coverage. Credits based on an entire program can be applied toward the benefit costs however the employee chooses.

Flexible Hours

Hours during the workday, workweek or pay period during which a time reporter covered by a flexible work schedule may choose to vary his times of arrival and departure from the worksite (see Scheduling)

Flexible Spending Account (FSA)

An account to which an employee and (optionally) an employer pledge an annual amount for a plan year. The employee then submits claims for authorized expenses.

Flexible TimeSpan

A user-defined period into which costs can be collected. Flexible TimeSpans can be as long or as short as you like—covering multiple years or a single day. The main purpose of Flexible TimeSpans is to assist you in analyzing costs.

Flexible Work Schedule

A method of scheduling a time reporter's time that is based on a range of flex hours of start and stop times and core work hours. This method can be used for scheduling clock and elapsed time reporters (see Scheduling)

FLSA Status

A PeopleSoft Human Resources term that is used to indicate whether a job is exempt or nonexempt according to the Fair Labor Standards Act. All employees associated with a

particular job will receive that job's FLSA Status. FLSA Status is an eligibility determination factor for PeopleSoft Benefits Administration.

Forecast Attribution

A FSI (financial services industry) transformation process through which forecasted product originations are pooled and run through the cash flow engine for future periods.

Forecast Calculation Process

In PeopleSoft Demand Planning, the process by which a **Statistical Forecast** is generated for each item at each level of the view. When a **Forecast Item** is set to recalculate, the system tries several forecast calculation methods and picks the one with the least amount of error. This process also makes adjustments for promotions and filters for abnormal demand.

Forecast Definition

Forecast definitions are a set of forecasting rules that generally govern multiple forecasts distinguished by key properties such as products, customers, channels, and so forth.

Forecast Element

Each forecast within a single definition is called a Forecast Element.

Forecast Fulfillment

In PeopleSoft Demand Planning, a process used to manage forecasted demand over a period of time. The process makes it possible to divide the total forecast demand into portions so that certain portions can be met, even if the total forecast cannot be met entirely.

Forecast Item

In PeopleSoft Demand Planning, a logical item used as the basis to forecast demand. The components of a forecast item key are defined for each level in a forecast view.

Forecast Level

See Level.

Forecast Period

A period in time as defined by the calendar for which data is processed through the PeopleSoft Demand Planning model.

Forecast Start Period/Year

Determines the most recent period for which demand data is available for a forecast view. This period can also be described as the last actual demand period to have had an impact on the forecast.

Forecast View

See **View**.

Foreign Education

Education acquired outside of any state of the U.S., the District of Columbia, the Commonwealth of Puerto Rico, a Trust Territory of the Pacific Islands, or any territory or possession of the U.S.

Form 10-K

A form used for annual reports pursuant to Section 13 or 15(d) of the Securities Exchange Act of 1934 for which no other form is prescribed.

Form 10-Q

A form used for quarterly reports under Section 13 or 15(d) of the Securities Exchange Act of 1934, filed pursuant to Rules 13a-13 or Rule 15d-13. This report, which public companies are required to file quarterly with the SEC, provides unaudited financial information and other selected material.

Form 5500 Participant Count Extract

A PeopleSoft Pension Administration data extract containing data that a plan administrator needs in order to complete IRS Form 5500, used to report on the number of plan participants.

Form S-8

A form used to register securities offered by a reporting company under its employee benefits plans, including stock option plans. Also called the Registration Statement under the Securities Act of 1933.

Form W-2

A form used by employers to provide workers with a statement of wages, tips and other compensation from the previous year. This form, distributed employees by January 31 of each year, reflects state and federal taxes, social security, Medicare wages, and tips withheld.

Formula

Element which enables you to define your own formulas for use—gives further flexibility to define complex organizational needs.

Formula Plan

This compensation distribution plan type is based on a pay out rule, as the pay out rule is defined. The pay out rule can be based on a flat amount, a percentage, or a data element. Whereas a Target Plan distributes pay out based on a comparison of a performance measure against a target, in a Formula Plan the pay out is based just on the pay out rule.

French Professional Elections

French companies employing a certain number of employees must hold elections for selecting personnel representatives (Délégués du personnel), and members of the Work Council (Comité d'Enterprise).

Frozen Rate

A rate that is applied to allocate resources to activities in place of the actual, budgeted and capacity rates calculated by the Activity-Based Management (ABM) Engine.

FTE (Full Time Equivalency)

FTE is the percent of full time the employee should normally work in this job. Full time is defined by the Standard Hours specified in either the Salary Plan Table or the Default Standard Hours specified in the Installation Table.

FTP (Funds Transfer Pricing) Adjustments

Adjustments made to the PeopleSoft Funds Transfer Pricing (FTP) base rate, for such factors as geographic premiums, liquidity premiums, embedded options, or incentive programs.

FTP (Funds Transfer Pricing) Base Rate

In PeopleSoft Funds Transfer Pricing (FTP), this refers to the basic charge or credit that is applied to a ledger account, a product, or an off-balance sheet position.

FTP (Funds Transfer Pricing)

See PeopleSoft Funds Transfer Pricing.

Full-Time Equivalent

See FTE.

Function

A category of pension calculation. PeopleSoft Pension Administration divides a pension calculation into nineteen “core functions” such as Service, Final Average Earnings, and Benefit Formula.

Function Result

The calculation rules for any of PeopleSoft Pension Administration’s nineteen core functions. These rules match Definitions—the specific parameters for the function—to the Groups of employees that use that particular definition. Function Result also refers to the value produced by the rules.

Fund ID

In the financial services industry, Fund ID is a lookup code used to track investment funds associated with a financial instrument or account. Provided primarily by the financial analytic applications to track investment funds for insurance policies.

Funds Transfer Pricing (FTP)

See PeopleSoft Funds Transfer Pricing.

Fungible

This term describes a resource used for multiple activities.

Future Period

Any pay period which is not current and whose close date hasn't passed (see Time Reporting).

Future Periods

The number of periods of future forecasts maintained by the PeopleSoft Demand Planning system.

G

Gang Reporting

See Crew Reporting.

General Deduction

Any non-benefit deduction. Examples include charitable deductions, union dues, parking, garnishments, and bonds. General Deductions are calculated from the General Deduction Table; Benefit Deductions draw on one of the benefits tables.

General Ledger Distribution

The process and guidelines by which accounting information is transferred from your PeopleSoft Receivables or Deduction Management system to a general ledger system.

General Schedule (GS)

Compensation and pay plan used by the Executive Branch of the federal government.

Generation Control

Generation control elements allow you to indicate to the system whether to process an element based upon criteria you define. There are six parameters that control this function and comprise the definition of the generation control element—HR Status, HR Action/Reason, Segment Status, Frequency, Formula, and Run Types.

Generic Conversion Factor

A conversion factor that applies universally between two units of measure. The factor is used in the conversions between levels of PeopleSoft Demand Planning **Forecast Item**s 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 f 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 andsecondary 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 manageaspects 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 (Minimum + Maximum)/2.

Midpoint Progression

In PeopleSoft Workforce Analytics, the percentage difference from one grade midpoint to the next higher-grade midpoint, calculated as (Midpoint2 – Midpoint1)/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 ona 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 of 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 MgrGetObj, SQLAccess, RemoteCall, and so on. When a client workstation sends a request to the application server, it sends a service name and a set of parameters, such as "MgrGetObject + parameters". The application server associates the service request with the appropriate server process to complete the transaction.

Service

The PeopleSoft Pension Administration function that determines how much service credit an employee has accrued.

Service Buy Back

The process by which an employee repays a pension plan in order to restore service credit that was forfeited when the employee withdrawal previous contributions. Typically, employees withdraw contributions upon termination and initiate service buy back processing upon 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 imbed 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 be 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=WEBLIB_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 differed. Only applicable if the Stock Action is LOA. Suspend Vesting must be selected for this rule to be applicable.

Vest Deferral Grace Period Service Rule

If the company provides a vest deferral grace period, they may stipulate that only certain individuals are eligible for the grace period based on service with the company. Only applicable if the Stock Action is LOA. Suspend Vesting must be selected for this rule to be applicable.

Vest Immediately

A stock option plan may provide that upon specific types of terminations, or upon a case by case scenario, all unvested shares held by an individual can be made immediately vested as of a specific date. Some companies' plans provide that under certain circumstances, such as retirement, the vesting of option shares accelerates upon termination of employment. When this occurs, you must modify the vesting schedule before you terminate the individual.

Vested Shares

Option shares that are free of any ownership restriction. Generally, vested exercised shares are fully owned by the optionee, free from restrictions and freely tradable.

Vested Termination

The termination of an employee who has a vested benefit. The benefit is deferred until the participant reaches retirement age. The employee is considered “Terminated Vested,” “Term Vested,” or simply “TV.”

Vesting

The method by which a granted option becomes free of all restrictions and the Optionee has full rights to the shares.

Vesting Schedule (Template)

A convenient way to set up the framework for a vesting schedule that can be uniformly applied to individual options. When you grant stock options, you define a vesting schedule to determine the default-vesting schedule for the option.

Vesting Service

The service used to determine an employee’s vesting percentage. Rules for accruing vesting service may be different from rules for accruing other plan service credits.

VETS–100 Federal Contractor Report

This report is required of employers in the United States. It lists federal job classifications, and the number of employees and new hires in the last 12 months who are special disabled military veterans or Vietnam era military veterans. It also provides totals for each job classification of both veterans and non-veterans who hold these jobs.

View

PeopleSoft Demand Planning,a multilevel forecast structure. Each view is associated with a unique view ID and includes information that defines the view and structure type. The three types of views are working, disbursement, and dynamic.

For PeopleSoft Budgeting see Budget View.

Virtual Tasks

In Time and Labor, Virtual task data is associated with a taskgroup profile that defines common characteristics for a given Taskgroup and Task Profile ID. A single row of data is linked to multiple Earnings records for multiple employees. By minimizing the physical storage of daily task data we provide enhanced performance without limiting its functionality.

Vision

In PeopleSoft Balanced Scorecard, the overall mission of an organization. Usually the highest level on a strategy tree. Vision is optional; you aren't required to have a vision component on each strategy tree.

See also Strategy Tree

Volume

Total share volume traded in a stock during market hours.

W

WA (Workforce Analytics)

See PeopleSoft Workforce Analytics

Waiver Of An OPM Qualification Standard

Involves setting aside requirements in a published standard to place an employee in a particular position, usually to avoid some kind of hardship to the employee, such as in cases of RIF or administrative error on part of the agency. Extra training and/or skills development may be needed to help the employee adjust to the new position. Waivers are granted by OPM or an agency, as appropriate, on a case-by-case basis, and do not directly affect other positions in the organization.

Warehouses

A warehouse reporting and analysis solution that supports the specific PeopleSoft business application that warehouse is using. It consists of predefined ETL maps, data warehouse tools, and Data Mart definitions. The warehouses are 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 it's 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

A

B

C

D

E

F

G

H

I

J

L

M

N

O