

PeopleSoft®

Enterprise PeopleTools 8.46 PeopleBook: Supported Integration Technologies

February 2005

Enterprise PeopleTools 8.46 PeopleBook: Supported Integration Technologies
SKU PT846SIS-B 0205

Copyright © 1988-2005 PeopleSoft, Inc. All rights reserved.

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

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

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

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

Open Source Disclosure

PeopleSoft takes no responsibility for its use or distribution of any open source or shareware software or documentation and disclaims any and all liability or damages resulting from use of said software or documentation. The following open source software may be used in PeopleSoft products and the following disclaimers are provided.

Apache Software Foundation

This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>). Copyright © 1999-2000 The Apache Software Foundation. All rights reserved.

THIS SOFTWARE IS PROVIDED “AS IS” AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

OpenSSL

Copyright © 1998-2003 The OpenSSL Project. All rights reserved.

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (<http://www.openssl.org/>).

THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT “AS IS” AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

SSLey

Copyright © 1995-1998 Eric Young. All rights reserved.

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com). This product includes software written by Tim Hudson (tjh@cryptsoft.com). Copyright © 1995-1998 Eric Young. All rights reserved. THIS SOFTWARE IS PROVIDED BY ERIC YOUNG “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Loki Library

Copyright © 2001 by Andrei Alexandrescu. This code accompanies the book:

Alexandrescu, Andrei. “Modern C++ Design: Generic Programming and Design Patterns Applied.” Copyright © 2001 Addison-Wesley. Permission to use, copy, modify, distribute and sell this software for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation.

Helma Project

Copyright © 1999-2004 Helma Project. All rights reserved.

THIS SOFTWARE IS PROVIDED “AS IS” AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE HELMA PROJECT OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Helma includes third party software released under different specific license terms. See the licenses directory in the Helma distribution for a list of these license.

Sarissa

Copyright © 2004 Manos Batsis.

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA.

Contents

General Preface

- About This PeopleBookxi**
- PeopleSoft Application Prerequisites.....xi
- PeopleSoft Application Fundamentals.....xi
- Documentation Updates and Printed Documentation.....xii
 - Obtaining Documentation Updates.....xii
 - Ordering Printed Documentation.....xii
- Additional Resources.....xiii
- Typographical Conventions and Visual Cues.....xiv
 - Typographical Conventions.....xiv
 - Visual Cues.....xv
 - Country, Region, and Industry Identifiers.....xvi
 - Currency Codes.....xvi
- Comments and Suggestions.....xvi
- Common Elements Used in PeopleBooks.....xvii

Preface

- Supported Integration Technologies Preface.....xix**
- Supported Integration Technologies.....xix

Chapter 1

- Getting Started with Supported Integration Technologies.....1**
- Supported Integration Technologies Overview.....1
- Supported Integration Technologies Implementation.....1
- Other Sources of Information.....2

Chapter 2

- Using PeopleSoft EDI Manager.....3**
- Understanding EDI.....3
 - EDI Overview.....3
 - EDI Codes and PeopleSoft Codes.....5
- Defining Event Codes and Action Codes.....5

Understanding Event Codes and Action Codes.....5

Pages Used to Define Event Codes and Action Codes.....6

Defining Event Codes.....6

Defining Action Codes.....6

Specifying Data Conversion Values.....6

Understanding Specifying Data Conversion Values.....7

Pages Used to Specify Data Conversion Values.....7

Assigning Trading Partner Conversion IDs.....7

Creating Conversion Data Profiles.....8

Defining EDI Transactions.....10

Understanding Defining EDI Transactions.....11

Pages Used to Define EDI Transactions.....11

Creating Transaction IDs.....11

Defining Partner Profiles.....13

Setting Profile Defaults.....14

Setting Up Trading Partners.....15

Understanding Trading Partners.....15

Pages Used to Set Up Trading Partners.....16

Defining PeopleSoft Entity Codes.....16

Creating Internal Partner IDs.....17

Assigning External Trading Partner IDs.....18

Defining Business Entities in External Companies.....19

Deleting PeopleSoft EDI Manager Objects.....20

Understanding Deleting PeopleSoft EDI Manager Objects.....20

Page Used to Delete PeopleSoft EDI Manager Objects.....20

Deleting EDI Manager Objects.....20

Chapter 3

Mapping EDI Transactions.....23

Understanding EDI Transaction Mapping.....23

EDI Transaction Mapping.....23

New EDI Transactions.....23

PeopleSoft Business Document Format.....24

Defining Inbound Maps.....26

Pages Used to Define Inbound Maps.....27

Creating Staging Tables and Work Records.....27

Creating Inbound Maps.....28

Defining the Inbound PeopleSoft Business Document Format.....29

Specifying Staging Tables.....31

Defining Outbound Maps.....35

 Pages Used to Define Outbound Maps.....36

 Creating Outbound Maps.....36

 Specifying Source Records.....37

 Defining the Outbound PeopleSoft Business Document Format.....38

Creating Data Mapping Profiles.....42

 Understanding Data Mapping Profiles.....42

 Page Used to Create Data Mapping Profiles.....42

 Creating Data Mapping Profiles.....42

Using PeopleSoft EDI Manager for General Data Extraction.....43

Chapter 4

Monitoring EDI Processing.....47

Understanding EDI Agents.....47

Preparing Outbound Maps for EDI Agents.....47

 Understanding Preparing Outbound Maps for EDI Agents.....48

 Page Used to Prepare Outbound Maps for EDI Agents.....48

 Preparing Outbound Maps.....48

Scheduling Inbound EDI Agents.....49

Scheduling Outbound EDI Agents.....51

Reviewing Errors and Summary Data.....53

 Reviewing and Correcting Errors.....54

 Reviewing Packages and Transaction Groups.....54

Chapter 5

Using Forms Routing and the Outgoing Forms API.....57

Understanding Forms Routings.....57

 PSFORMS.DLL.....57

Session-Level Operations.....58

 PsfCloseSession.....58

 PsfGetAPIInfo.....58

 PsfOpenSession.....59

Query Operations.....60

 PsfGetFieldCount.....60

 PsfGetFieldList.....61

 PsfGetFormCount.....62

 PsfGetFormList.....63

 PsfGetLastError.....64

Send Operations.....65
 PsfSendForm.....65

Chapter 6

Using the Open Query ODBC Driver and API.....67

Understanding PeopleSoft Open Query.....67
 Architecture.....68
 PeopleSoft Open Query ODBC Driver.....70
 PeopleSoft Open Query API.....70
 External Reporting Tools.....71
 Internet.....71
 Supported ODBC Version 2.5 Functions.....71
 PeopleTools Initialization Procedures.....74
 SQLAllocEnv.....74
 SQLAllocConnect.....74
 SQLFreeEnv.....75
 Connection Procedures.....75
 SQLConnect.....76
 SQLDriverConnect.....77
 Information Procedures.....79
 SQLGetInfo.....79
 SQLFunctions.....80
 SQLGetTypeInfo.....81
 Catalog Procedures (Metadata).....82
 SQLProcedures.....82
 SQLProcedureColumns.....83
 Using SQL Execution Procedures.....85
 SQL Execution Procedures.....86
 SQLAllocStmt.....86
 SQLExecDirect.....87
 SQLPrepare.....88
 SQLExecute.....88
 SQLColAttributes.....89
 SQLDescribeCol.....90
 SQLDescribeParam.....91
 SQLGetRowCount.....91
 SQLNumParams.....92
 SQLNumResultCols.....92
 SQLBindCol.....93

- SQLBindParameter.....94
- Execution Models.....95
- Using Retrieval Procedures.....96
- Retrieval Procedures.....96
 - SQLFetch.....96
 - SQLGetData.....97
- Status and Error Retrieval Procedures.....98
 - SQLError.....98
- Transaction and Connection Termination Procedures.....99
 - SQLTransact.....99
 - SQLDisconnect.....100
 - SQLFreeConnect.....100
 - SQLFreeEnv.....101
- ODBC Compliance.....101
- ODBC to RDM Data Types.....102
- PeopleSoft Open Query ODBC API Example.....103

- Glossary of PeopleSoft Terms.....105**

- Index127**

About This PeopleBook

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

This preface discusses:

- PeopleSoft application prerequisites.
- PeopleSoft application fundamentals.
- Documentation updates and printed documentation.
- Additional resources.
- Typographical conventions and visual cues.
- Comments and suggestions.
- Common elements in PeopleBooks.

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

PeopleSoft Application Prerequisites

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

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

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

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

PeopleSoft Application Fundamentals

Each application PeopleBook provides implementation and processing information for your PeopleSoft applications.

Note. Application fundamentals PeopleBooks are not applicable to the PeopleTools product.

For some applications, additional, essential information describing the setup and design of your system appears in a companion volume of documentation called the application fundamentals PeopleBook. Most PeopleSoft product lines have a version of the application fundamentals PeopleBook. The preface of each PeopleBook identifies the application fundamentals PeopleBooks that are associated with that PeopleBook.

The application fundamentals PeopleBook consists of important topics that apply to many or all PeopleSoft applications across one or more product lines. Whether you are implementing a single application, some combination of applications within the product line, or the entire product line, you should be familiar with the contents of the appropriate application fundamentals PeopleBooks. They provide the starting points for fundamental implementation tasks.

Documentation Updates and Printed Documentation

This section discusses how to:

- Obtain documentation updates.
- Order printed documentation.

Obtaining Documentation Updates

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

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

See Also

PeopleSoft Customer Connection, <https://www.peoplesoft.com/corp/en/login.jsp>

Ordering Printed Documentation

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

- Web
- Telephone
- Email

Web

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

Telephone

Contact MMA Partners at 877 588 2525.

Email

Send email to MMA Partners at peoplebookspres@mmapartner.com.

See Also

PeopleSoft Customer Connection, <https://www.peoplesoft.com/corp/en/login.jsp>

Additional Resources

The following resources are located on the PeopleSoft Customer Connection website:

Resource	Navigation
Application maintenance information	Updates + Fixes
Business process diagrams	Support, Documentation, Business Process Maps
Interactive Services Repository	Interactive Services Repository
Hardware and software requirements	Implement, Optimize + Upgrade, Implementation Guide, Implementation Documentation & Software, Hardware and Software Requirements
Installation guides	Implement, Optimize + Upgrade, Implementation Guide, Implementation Documentation & Software, Installation Guides and Notes
Integration information	Implement, Optimize + Upgrade, Implementation Guide, Implementation Documentation and Software, Pre-built Integrations for PeopleSoft Enterprise and PeopleSoft EnterpriseOne Applications
Minimum technical requirements (MTRs) (EnterpriseOne only)	Implement, Optimize + Upgrade, Implementation Guide, Supported Platforms
PeopleBook documentation updates	Support, Documentation, Documentation Updates
PeopleSoft support policy	Support, Support Policy
Prerelease notes	Support, Documentation, Documentation Updates, Category, Prerelease Notes
Product release roadmap	Support, Roadmaps + Schedules
Release notes	Support, Documentation, Documentation Updates, Category, Release Notes

Resource	Navigation
Release value proposition	Support, Documentation, Documentation Updates, Category, Release Value Proposition
Statement of direction	Support, Documentation, Documentation Updates, Category, Statement of Direction
Troubleshooting information	Support, Troubleshooting
Upgrade documentation	Support, Documentation, Upgrade Documentation and Scripts

Typographical Conventions and Visual Cues

This section discusses:

- Typographical conventions.
- Visual cues.
- Country, region, and industry identifiers.
- Currency codes.

Typographical Conventions

This table contains the typographical conventions that are used in PeopleBooks:

Typographical Convention or Visual Cue	Description
Bold	Indicates PeopleCode function names, business function names, event names, system function names, method names, language constructs, and PeopleCode reserved words that must be included literally in the function call.
<i>Italics</i>	Indicates field values, emphasis, and PeopleSoft or other book-length publication titles. In PeopleCode syntax, italic items are placeholders for arguments that your program must supply. We also use italics when we refer to words as words or letters as letters, as in the following: Enter the letter <i>O</i> .
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 the W key.
Monospace font	Indicates a PeopleCode program or other code example.

Typographical Convention or Visual Cue	Description
“ ” (quotation marks)	Indicate chapter titles in cross-references and words that are used differently from their intended meanings.
... (ellipses)	Indicate that the preceding item or series can be repeated any number of times in PeopleCode syntax.
{ } (curly braces)	Indicate a choice between two options in PeopleCode syntax. Options are separated by a pipe ().
[] (square brackets)	Indicate optional items in PeopleCode syntax.
& (ampersand)	<p>When placed before a parameter in PeopleCode syntax, an ampersand indicates that the parameter is an already instantiated object.</p> <p>Ampersands also precede all PeopleCode variables.</p>

Visual Cues

PeopleBooks contain the following visual cues.

Notes

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

Note. Example of a note.

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

Important! Example of an important note.

Warnings

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

Warning! Example of a warning.

Cross-References

PeopleBooks provide cross-references either under the heading “See Also” or on a separate line preceded by the word *See*. Cross-references lead to other documentation that is pertinent to the immediately preceding documentation.

Country, Region, and Industry Identifiers

Information that applies only to a specific country, region, or industry is preceded by a standard identifier in parentheses. This identifier typically appears at the beginning of a section heading, but it may also appear at the beginning of a note or other text.

Example of a country-specific heading: “(FRA) Hiring an Employee”

Example of a region-specific heading: “(Latin America) Setting Up Depreciation”

Country Identifiers

Countries are identified with the International Organization for Standardization (ISO) country code.

Region Identifiers

Regions are identified by the region name. The following region identifiers may appear in PeopleBooks:

- Asia Pacific
- Europe
- Latin America
- North America

Industry Identifiers

Industries are identified by the industry name or by an abbreviation for that industry. The following industry identifiers may appear in PeopleBooks:

- USF (U.S. Federal)
- E&G (Education and Government)

Currency Codes

Monetary amounts are identified by the ISO currency code.

Comments and Suggestions

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

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

Or send email comments to doc@peoplesoft.com.

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

Common Elements Used in PeopleBooks

As of Date	The last date for which a report or process includes data.
Business Unit	An ID that represents a high-level organization of business information. You can use a business unit to define regional or departmental units within a larger organization.
Description	Enter up to 30 characters of text.
Effective Date	The date on which a table row becomes effective; the date that an action begins. For example, to close out a ledger on June 30, the effective date for the ledger closing would be July 1. This date also determines when you can view and change the information. Pages or panels and batch processes that use the information use the current row.
Once, Always, and Don't Run	Select Once to run the request the next time the batch process runs. After the batch process runs, the process frequency is automatically set to Don't Run. Select Always to run the request every time the batch process runs. Select Don't Run to ignore the request when the batch process runs.
Process Monitor	Click to access the Process List page, where you can view the status of submitted process requests.
Report Manager	Click to access the Report List page, where you can view report content, check the status of a report, and see content detail messages (which show you a description of the report and the distribution list).
Request ID	An ID that represents a set of selection criteria for a report or process.
Run	Click to access the Process Scheduler request page, where you can specify the location where a process or job runs and the process output format.
SetID	An ID that represents a set of control table information, or TableSets. TableSets enable you to share control table information and processing options among business units. The goal is to minimize redundant data and system maintenance tasks. When you assign a setID to a record group in a business unit, you indicate that all of the tables in the record group are shared between that business unit and any other business unit that also assigns that setID to that record group. For example, you can define a group of common job codes that are shared between several business units. Each business unit that shares the job codes is assigned the same setID for that record group.
Short Description	Enter up to 15 characters of text.
User ID	An ID that represents the person who generates a transaction.

See Also

Enterprise PeopleTools 8.46 PeopleBook: PeopleSoft Process Scheduler

Enterprise PeopleTools 8.46 PeopleBook: Using PeopleSoft Applications

Supported Integration Technologies Preface

This preface provides an overview of supported integration technologies.

Supported Integration Technologies

Prior to PeopleSoft 8, PeopleTools delivered a variety of tools and technologies for integrating PeopleSoft applications with each other and with third-party products and services. Typically, these technologies were narrowly focused on industry-specific transactions and data formats. Although we recommend the more robust and adaptable tools delivered with the current release, PeopleSoft still supports the older technologies documented in this PeopleBook.

CHAPTER 1

Getting Started with Supported Integration Technologies

This chapter discusses:

- Supported integration technologies.
- Supported integration technologies implementation.

Supported Integration Technologies Overview

This PeopleBook describes several integration technologies that PeopleSoft delivered prior to PeopleSoft 8.

Note. PeopleSoft supports the older technologies documented in this PeopleBook; however, you should use the more robust and adaptable tools delivered with the current release.

EDI Manager

Electronic Data Interchange (EDI) is a standard means of exchanging data between companies so that they can transact business electronically. PeopleSoft EDI Manager enables you to define event and action codes, define EDI transactions, set up trading partners, map transactions, and monitor transaction processing.

Outgoing Forms API

Forms routing enables the system to take data from a PeopleSoft page on which a user is working, enter it onto a third-party form, and mail the completed form to designated users by means of the forms product's mail capabilities. PeopleSoft provides a PSFORMS dynamic link library (PSFORMS.DLL) that PeopleSoft applications use to communicate with forms software, which includes session-level, query, and send operations.

Open Query ODBC Driver and API

The PeopleSoft Open Query ODBC driver and API enable third-party reporting tools or applications to access PeopleSoft data in conformance with the PeopleSoft Query access architecture (the embedded SQL access intelligence provided by PeopleSoft Query).

Supported Integration Technologies Implementation

The technologies discussed in this PeopleBook are automatically installed with PeopleTools. In the planning phase of your implementation, take advantage of all PeopleSoft sources of information, including the installation guides and release notes. Moreover, in addition to the information in this PeopleBook, refer to the following PeopleBooks.

Technology	PeopleBook
EDI Manager	<i>PeopleTools 8.46 PeopleBook: SQR for PeopleSoft Developers</i>
Outgoing Forms API	<i>PeopleTools 8.46 PeopleBook: PeopleSoft Application Designer</i>
Open Query ODBC Driver and API	<i>PeopleTools 8.46 PeopleBook: PeopleSoft Query and PeopleTools 8.46 PeopleBook: Crystal Reports for PeopleSoft</i>

Other Sources of Information

In addition to the implementation considerations presented in this section, take advantage of all PeopleSoft sources of information, including the installation guides, release notes, PeopleBooks, and curriculum.

See Also

[“Supported Integration Technologies Preface,” page xix](#)

Enterprise PeopleTools 8.46 PeopleBook: Getting Started with PeopleTools

CHAPTER 2

Using PeopleSoft EDI Manager

This chapter provides an overview of Electronic Data Interchange (EDI) and discusses how to:

- Define event and action codes.
- Specify data conversion values.
- Define EDI transactions.
- Set up trading partners.
- Delete PeopleSoft EDI Manager objects.

Note. EDI Manager should no longer be used to load or unload flat files that are not X.12 or EDIFACT EDI transactions. File layouts have stronger file access methods, with batch file import and export, interactive import data troubleshooting, and automatic generation of import PeopleCode.

Understanding EDI

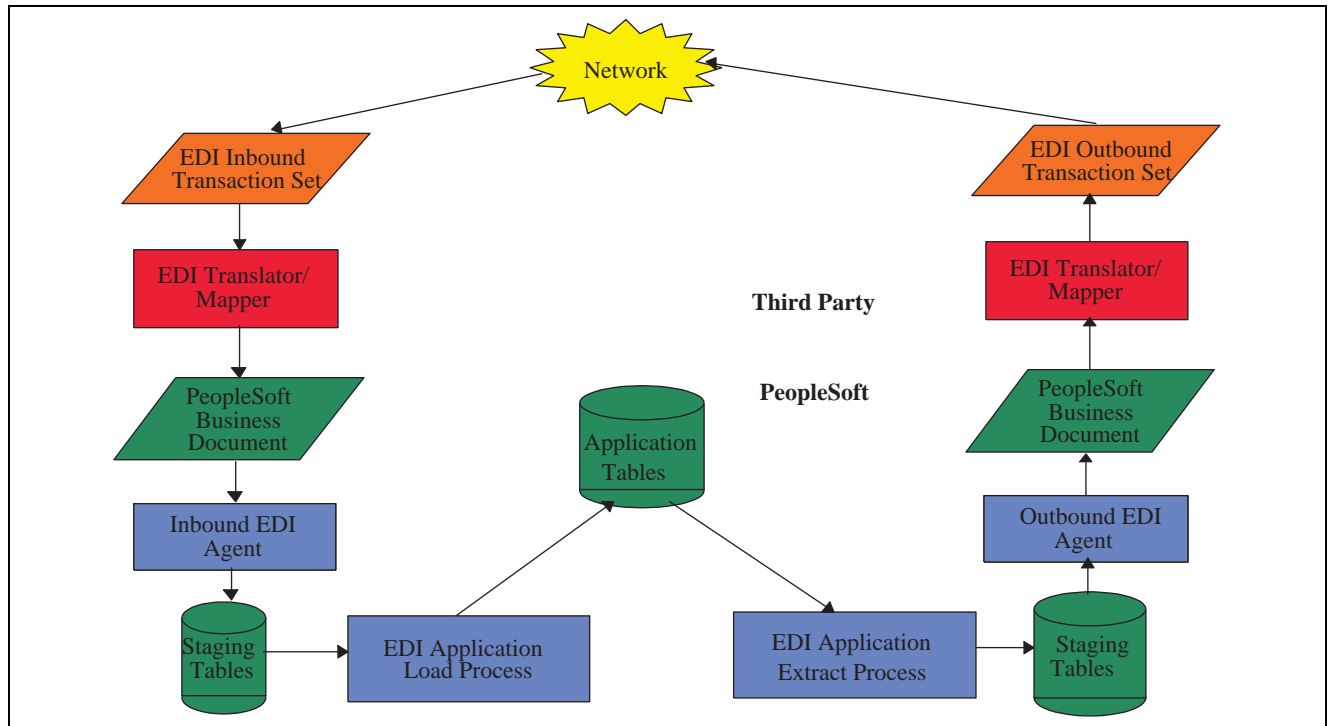
This section provides an overview of EDI and discusses EDI codes and PeopleSoft codes.

EDI Overview

EDI is a standard means of exchanging data between companies so that they can transact business electronically. For example, using EDI, a company can submit an order to a vendor, and the vendor can acknowledge and fulfill the order without any paper changing hands or any contact between company representatives.

EDI provides a standard format for transaction data, enabling trading partners to communicate in a common language. When one company needs to initiate a transaction with another, it extracts the transaction data from its database, translates it into the common EDI format, and transmits it over a network to the trading partner. The second company receives the EDI transmission and transfers its data into its transaction processing application.

This process involves several steps. Here is an overview of the architecture that enables PeopleSoft applications to complete EDI transactions:



PeopleSoft EDI architecture

The portion above the dotted line shows the services typically provided by a value-added network (VAN). A VAN provides services related to the exchange of EDI transactions—usually a private network for exchanging EDI transactions, although the network could also be the internet.

For incoming transactions, the transfer to PeopleSoft applications comes when the EDI translation software converts transactions from the standard EDI formats X.12 or EDIFACT into PeopleSoft Business Document format. A PeopleSoft-supplied EDI agent reads the PeopleSoft Business Document files and places their data in staging tables in the PeopleSoft database. An EDI application load Structured Query Report (SQR) transfers data from the staging tables to the appropriate application tables. From there, the data is processed like any other PeopleSoft transaction.

The process for outgoing transactions is similar. You periodically run an SQR that extracts the appropriate data from the application tables and copies it to a set of staging tables. An EDI agent creates PeopleSoft business documents from this data; then the EDI translation software converts the documents into X.12 or EDIFACT format and transmits them to the trading partner.

So the PeopleSoft portion of the process has two stages:

- Based on data mappings that you define, EDI agents copy data between PeopleSoft business documents and the incoming or outgoing staging tables.
- SQRs transfer data between the staging tables and the application tables, performing any necessary data validation.

You use PeopleSoft EDI Manager to define the mappings that EDI agents use for conversions and to maintain information about your vendors and other trading partners.

The staging tables are a temporary storage area for EDI transaction data. Later, a transaction-specific application load process transfers the data from the staging tables into the application tables.

EDI Codes and PeopleSoft Codes

When you receive an EDI transaction from a trading partner, much of its data is in the form of codes or identification numbers. For example, the trading partner who submitted the transaction is given as a trading partner ID, and the type of transaction is identified by a transaction ID.

Your PeopleSoft database also stores much of its data in the form of codes or IDs. You have business unit IDs, customer IDs, employee IDs, user IDs, and so on. Some of these codes and IDs represent the same data as the codes and IDs in the EDI transaction, even though the codes themselves might be different.

When the EDI agent processes a transaction, it copies the transaction data into the PeopleSoft database. As it does so, it can convert the external EDI codes into the internal PeopleSoft codes. Similarly, as it writes out the file for an outgoing EDI transaction, it can convert the PeopleSoft codes into codes that your trading partner will recognize. You specify how the EDI agent converts the codes using PeopleSoft EDI Manager.

PeopleSoft EDI Manager can perform two kinds of conversions:

- It can translate between EDI event codes and PeopleSoft action codes, which specify what action a transaction requires.
- It can convert data values from any field in the transaction.

Defining Event Codes and Action Codes

This section provides an overview of event codes and action codes and discusses how to:

- Define event codes.
- Define action codes.

Understanding Event Codes and Action Codes

EDI transactions use *event codes* to specify what action the transaction calls for. For example, event codes specify whether the current transaction is a completely new transaction, a resubmitted transaction, or an update to a previous transaction.

EDI event codes come in two types, both of which can be specified for the same transaction:

- *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.
- *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.

PeopleSoft applications, on the other hand, determine the action to take on a transaction using a single *action code*. So when an EDI agent processes an EDI transaction, it needs to convert its event codes into a PeopleSoft action code.

Using PeopleSoft EDI Manager, you specify which pairs of event codes get translated into which action codes. Because you may want to process transactions differently depending on which trading partner they come from, EDI Manager enables you to define different event code-to-action code translations for each trading partner.

After you complete the following activities, you must specify a profile for each trading partner.

See Also

Chapter 2, “Using PeopleSoft EDI Manager,” Setting Up Trading Partners, page 15

Pages Used to Define Event Codes and Action Codes

Page Name	Object Name	Navigation	Usage
EC Primary Event Table	EC_PRI_EVENT_TABLE	PeopleTools, EDI Manager, Convert EDI/PeopleSoft Code, Primary Event Codes	Define primary event codes.
EC Sec Event Table	EC_SEC_EVENT_TABLE	PeopleTools, EDI Manager, Convert EDI/PeopleSoft Code, Secondary Event Codes	Define secondary event codes.
EC Action Code Definition	EC_ACTION_CODES	PeopleTools, EDI Manager, Convert EDI/PeopleSoft Code, Action Codes	Define action codes.

Defining Event Codes

To define primary or secondary event codes:

1. Select PeopleTools, EDI Manager, Convert EDI/PeopleSoft Code.
2. Select Primary Event Codes or Secondary Event Codes.
You can search for an existing value or directly add a new value by clicking the link.
3. Enter the event code as it will appear in PeopleSoft business documents.
4. Enter a description of the event code and save the page.
This description is for your information only. It doesn't affect processing.

Defining Action Codes

PeopleSoft delivers EDI Manager with a complete set of codes for the actions that standard PeopleSoft applications support. Add new action codes only if you create new programs that handle new actions.

To define an action code:

1. Select Action Codes.
2. Enter the action code as you want the EDI agent to write it in the electronic commerce staging tables.
Click Search to look for an existing action code.
3. Enter a description of the action code and save the page.
This description is for your information only. It doesn't affect processing.

Specifying Data Conversion Values

This section provides an overview of specifying data conversion values and discusses how to:

1. Assign trading partner conversion IDs.
2. Create conversion data profiles.

See Also

[Chapter 2, “Using PeopleSoft EDI Manager,” Setting Up Trading Partners, page 15](#)

Understanding Specifying Data Conversion Values

In many cases, the set of possible values in a particular field of an EDI transaction doesn't match the corresponding set of values in the PeopleSoft database. For example, the EDI transaction might identify bank transaction codes using three-digit numbers, while the PeopleSoft database uses single letters to represent the same codes.

In such cases, PeopleSoft EDI Manager needs to translate the external values into the corresponding internal values.

Pages Used to Specify Data Conversion Values

Page Name	Object Name	Navigation	Usage
Conversion Type Definition	EC_ECTPCVT	PeopleTools, EDI Manager, Convert EDI/PeopleSoft Code, Conversion Types	Identify database tables whose values need to be converted.
Conversion Data Profile	EC_ECTPCVT_VALUES	PeopleTools, EDI Manager, Convert EDI/PeopleSoft Code, Conversion Data Profile	Specify how values from a PeopleSoft database table appears in PeopleSoft business documents.

Assigning Trading Partner Conversion IDs

Access the Conversion Type Definition page.

Conversion Type Definition

Conversion Type ID: TTT

Record: SDK_CURR_CD_TE

Description: Currency code table

Long Description:

Conversion Type Definition page

To identify a database table whose values need to be converted:

1. Select PeopleTools, EDI Manager, Convert EDI/PeopleSoft Code, Conversion Types.
2. Enter a conversion type ID or search for an existing one.
You can also search for the record name.
3. Enter the record name of the table whose values you want to convert and enter a description.
You can enter a description in the Description text box, the large edit box, or both.
4. Save the page.

Creating Conversion Data Profiles

Access the Conversion Data Profile page.

Conversion Data Profile

Cvt Pro ID: IDT

Description:

Source Profile:

Conversion Find | View All First 1 of 1 Last

Cvt TypeID: TTT Currency code table Setid Based SetID:

No-Value Pass Thru Default Value:

Conversion Values Find | View All First 1 of 1 Last

Internal Value	Int Deflt	External Value:	Ext Deflt
<input type="text"/>	<input checked="" type="checkbox"/>	<input type="text"/>	<input checked="" type="checkbox"/>

Conversion Data Profile page

A conversion data profile takes the values from a particular PeopleSoft database table (such as the table holding bank transaction codes) and specifies how the values appear in PeopleSoft business documents. You can create multiple conversion data profiles for the same table, because you might need to create different conversions for different trading partners. For example, different banks might use different transaction codes.

To create a conversion data profile:

1. Select PeopleTools, EDI Manager, Convert EDI/PeopleSoft Code, Conversion Data Profile.

Search for an existing Conversion data profile or enter a new value.

The Conversion Data Profile page appears. You use this page to specify translation values for one or more database tables, identified here by their conversion profile IDs.

The page includes two areas to enter data. The outer area enables you to add multiple conversion type IDs to the conversion profile. The inner area enables you to scroll among the values in the table identified by the conversion ID.

2. (Optional.) Copy the definition of a similar conversion data profile.

If the profile you're creating is similar to an existing profile, you can copy the details of the existing profile definition into this component. Then you can change just the parts that are different.

Select the existing *Source Profile*, then click the EC Copy Conversion Profile button. The system copies the conversion values from the selected profile.

3. Enter a description of the conversion data profile.

You can enter both a short description (in the Description text box) and a long description (in the long edit box). These descriptions are for your information only; the system doesn't use them.

4. Select a conversion type ID.

In the Cvt TypeID field, select the trading partner conversion type ID of the table that you want to provide external values for.

5. Specify whether the table whose values you are converting has a setID key.

Many tables in PeopleSoft Financials applications have a setID key. The setID key enables you to store data for multiple business units on the same table while making sure that each unit only accesses its own data. Selecting a setID controls the list of internal values available for mapping: if the table you are retrieving the internal values from has a SetID field, the Conversion Data Profile page only displays the values for the selected setID.

To select a setID, select the SetID Based check box, then select the appropriate SetID in the field that appears.

6. Select an internal value and enter the corresponding external value.

In the Internal Value drop-down list box, select a value from the table whose conversion ID you selected. Then, in the External Value text box, type the corresponding value as it will appear in PeopleSoft business documents.

7. Specify which internal and external values to use when there are multiple possibilities.

One potential complication of having many-to-one mapping occurs with outbound transactions: for a given internal value, what value should the outbound EDI agent write into the outbound file? In this situation, the outbound EDI agent uses the row marked as the internal default (Int Deflt field) value.

The same scenario holds in reverse when you have multiple internal database values that you need to map to the same external value. There is no problem with outbound transactions: the outbound EDI agent maps the internal value to the appropriate external value. For incoming transactions, however, you need to specify which of the multiple internal values to use. You do so by selecting one of the rows as the external default (Ext Defltfield) value.

When you need to convert multiple external values to the same internal value, you need to mark each of the external values as an external default, then mark *one* of the rows as the internal default. When you need to convert multiple internal values to the same external value, mark each of the internal values as an internal default, then mark one row as the external default.

Note. You need to perform this step even if you have all one-to-one mappings. Select the Int Deflt and Ext Deflt check boxes for every row.

8. Repeat the previous steps to add additional conversion IDs.

Defining EDI Transactions

This section provides an overview of defining EDI transactions and discusses how to:

- Create transaction IDs.
- Define partner profiles.
- Set profile defaults.

See Also

Chapter 2, “Using PeopleSoft EDI Manager,” Setting Up Trading Partners, page 15

Understanding Defining EDI Transactions

When you receive an EDI transaction from a trading partner, the first record of the transaction includes a *transaction ID* that identifies the transaction type. The EDI agent uses the transaction ID, in conjunction with the trading partner ID, to determine which inbound map to use to process the transaction data. Similarly, when you initiate an outbound transaction, the EDI agent puts the appropriate transaction ID in the first transaction record so that the recipient knows what kind of transaction you've sent.

In PeopleSoft EDI Manager, you need to identify the transactions that your system is prepared to process. You also need to specify, for each of your trading partners, the transactions that they are authorized to perform. After you complete these steps, you must assign the profiles to your trading partners.

Pages Used to Define EDI Transactions

Page Name	Object Name	Navigation	Usage
Transaction Definition	EC_TRANS_PNL	PeopleTools, EDI Manager, Define EDI Transactions, Transactions	Create EDI transaction IDs.
Partner Profile - Profile Definition	EC_TP_PROFILE_1	PeopleTools, EDI Manager, Define EDI Transactions, Partner Profile Click the Profile Definition tab.	Define partner profiles.
Partner Profile - Profile Defaults	EC_TP_PROFILE_2	PeopleTools, EDI Manager, Define EDI Transactions, Partner Profile Click the Profile Defaults tab.	Specify the transactions a partner with a specified profile can perform and how the EDI agent converts event codes into action codes for each transaction.

Creating Transaction IDs

Access the Transaction Definitions page.

The screenshot shows the 'Transaction Definition' page. It features a 'Trans ID' field with 'TNT' and 'Inbound' entered. Below is a 'Descr:' field. The main section is 'Transaction Option Definition', which has a header bar with 'Find | View All' and navigation buttons 'First', '1 of 1', and 'Last'. It contains a table with columns 'EC Option' and 'Description', and 'Add' and 'Delete' buttons. Below this is another section 'Transaction Option Values', also with a header bar and navigation buttons, containing a table with columns 'Op Value' and 'Description', and 'Add' and 'Delete' buttons.

Transaction Definition page

To define a transaction:

1. Add a new value and enter a unique EC transaction ID as it will appear in PeopleSoft business documents.
2. Specify whether you're setting up an ID for inbound or outbound transactions.

Note. If you support both inbound and outbound transactions of the same transaction type, you have to define two transaction IDs.

3. Click the Add button.

The Transaction Definition page appears. You use this page to define the transaction and specify any special transaction parameters.

4. Enter a description of the transaction in the Descr (Description) text box.

The description appears in list boxes when you need to select a transaction ID.

5. Enter the name of a transaction option that the application load or application extract process uses.

Use the EC Option text box to specify a transaction parameter that the system will use when it copies data between the electronic commerce staging tables into the transactional application tables.

6. Enter the valid transaction option values for the transaction option.

List the valid values in the Op Value field for the transaction option that you specified in the previous step. Add a row for each option value. When you define a trading partner, you'll specify which of these values to use for that partner. In this way, the application load or application extract process can handle the transaction differently depending on the trading partner that it comes from.

Repeat steps 5 and 6 to define additional transaction options.

Defining Partner Profiles

Access the Partner Profile - Profile Definition page.

Partner Profile - Profile Definition page

To define a partner profile:

1. (Optional) Copy the definition of a similar partner profile.

If the profile that you are creating is similar to an existing profile, you can copy the details of the existing profile definition into this component. Then you can change just the parts that are different.

Select the existing Source TPID, and then click the EC Copy Profile Function button. The system copies the values from the selected profile.

2. Specify the directory where you want the EDI agent to write outbound transaction files for partners with this profile.

Enter the directory path in the EC Outbound File List Path text box.

Note. If PeopleSoft EDI Manager runs on a server under MVS, it ignores this setting. Instead, it looks in the JCL for a DDNAME matching the file name that you specify in the next step.

3. Enter a name for the file that lists all pending outbound transaction files.

As the EDI agent creates transaction files, it adds the names of those files to a list file, the name of which you enter in the EC Outbound File List Name text box.

The transaction files themselves are automatically named 1.DAT, 2.DAT, and so on, except on MVS servers. On MVS, the files are named using the first eight characters of the external trading partner ID.

4. (Optional) Select New List File Per Run if you want a new list generated every run.
5. (Optional) Click the Message button to insert a standard message header or footer in the file.

Setting Profile Defaults

Access the Partner Profile - Profile Defaults page.

The screenshot shows the 'Partner Profile - Profile Defaults' page. It features three main sections:

- Available Transactions:** Includes a search bar for 'EC Transaction ID' (containing 'TNT') and 'In/Out' (containing 'I'). There are 'Add' and 'Delete' buttons.
- Profile Defaults:** Includes a search bar for 'EC Option' and 'Op Value'. There are '+' and '-' buttons.
- Action Assignment:** Includes search bars for 'EC Action Code', 'Pri Event', and 'Sec Event'. There are '+' and '-' buttons.

Partner Profile - Profile Defaults page

Use this page to specify which transactions a partner with this profile can perform and how the EDI agent converts event codes into action codes for each transaction. If a transaction has transaction options associated with it, you also specify which value to assign to that option for partners with this profile.

To set profile defaults:

1. Select a transaction type that partners with this profile can perform.

Select the transaction ID in the EC Transaction ID field. Specify whether it's an inbound or outbound transaction by selecting *I* or *O* in the In/Outfield.

2. If the transaction type has transaction options associated with it, specify which value to use for each one in the Profile Defaults group box.

When you define a transaction type, you can specify transaction options that the system uses when it copies data between the electronic commerce staging tables into the transactional application tables. If the transaction type that you are adding includes options, specify which of the available values for that option the EDI agent should use for partners with this profile.

Note. Most transaction types do not have any options, and you can bypass this step.

Select the option in the EC Option field. Then select one of its valid values in the Op Value field. Repeat this step for each defined transaction option.

3. Specify which action code to use for each pair of primary and secondary event codes.

The controls in the Action Assignment group box enable you to specify how the EDI agent translates between the event codes in a PeopleSoft business document and the action codes in the electronic commerce staging tables.

Select one of the defined action codes in the EC Action Code field, and then select the primary and secondary event codes that map to this action code. If this transaction type uses only a primary event code (Pri Event field), you can leave the Sec Event (secondary event) box empty.

Repeat this step to define each possible action code and combination of primary and secondary event codes for this transaction type.

4. Repeat steps 1 through 3 for each transaction type that partners with this profile can perform.

See Also

[Chapter 2, “Using PeopleSoft EDI Manager,” EDI Codes and PeopleSoft Codes, page 5](#)

[Chapter 2, “Using PeopleSoft EDI Manager,” Creating Transaction IDs, page 11](#)

Setting Up Trading Partners

This section provides an overview of trading partners and discusses how to:

- Define PeopleSoft entity codes.
- Create internal partner IDs.
- Assign external trading partner IDs.
- Define business entities in external companies.

Understanding Trading Partners

In PeopleSoft EDI Manager, a trading partner is a business entity in an external company with which a business entity in your company does business.

When you set up trading partners in PeopleSoft EDI Manager, you need to define the internal structure of your company, and possibly of the companies with which you trade.

PeopleSoft Financials applications enable you to organize your company into multiple business units, each of which tracks its data somewhat independently. Each business unit likely does business with a different set of trading partners. When a trading partner submits an EDI transaction, it needs to address it to a specific business unit, or the EDI agent needs to know which business unit to forward the transaction to.

Conversely, the external companies that you do business with are not monolithic entities. They have different divisions, departments, or business units, and you might need to address EDI transactions to a specific entity inside that company.

Pages Used to Set Up Trading Partners

Page Name	Object Name	Navigation	Usage
Ec Entity Code Tbl (Entity Code Table)	EC_ENTITY_CODE_TBL	PeopleTools, EDI Manager, Setup Trading Partners, Entity Codes	Define PeopleSoft entity codes.
Ec Int Partner Def (Internal Partner Definition)	EC_EXT_PARTNER_DEF	PeopleTools, EDI Manager, Setup Trading Partners, Internal Partners	Create internal partner IDs.
Ec Ext Partner Def (External Partner Definition)	EC_EXT_PARTNER_DEF	PeopleTools, EDI Manager, Setup Trading Partners, External Partners	Create external trading partner IDs.
Ec Bus Entity Def (Business Entity Definition)	EC_BUS_ENTITY_DEF	PeopleTools, EDI Manager, Setup Trading Partners, Business Entities	Define business entities in external companies.

Defining PeopleSoft Entity Codes

Access the Ec Entity Code Tbl (Entity Code Table) page.

The screenshot shows the 'Ec Entity Code Tbl' page with the following fields:

- PeopleSoft Entity Code:** APBU
- Description:** Accounts Payable Business Unit
- Record (Table) Name:** BUS_UNIT_TBL_QE (with a search icon)
- External Entity:**

EC Entity Code Table page

You define entity codes for the types of business entities that serve as trading partners. Internally, the entities are usually business units. Use this page to specify the type of business entity to which the code applies and which table in the PeopleSoft database lists valid entities of this type.

To define a PeopleSoft entity code:

1. Select the record definition for the table that lists valid entities of this type.

Select the prompt table for this type of entity from the Record (Table) Name field. For example, if you're defining an entity code for business units, specify the table that holds business unit IDs.

2. Specify whether this entity code applies to external trading partners.

You need to identify entity codes both for your internal business entities and for the external companies that you trade with. For example, you'll probably want to create entity codes based on the Vendor table and the Customer table.

If the entity code that you're defining is for external companies, select the External Entity check box.

Creating Internal Partner IDs

Access the Ec Int Partner Def (Internal Partner Definition) page.

The screenshot shows the 'Ec Int Partner Def' page with the following details:

- Int TPID:** GENERALINT
- Descr:** General Internal Partner
- Business Unit Assignment:** A table with two columns: '*PS Code' (value: GENR) and '*Unit' (value: GENR). It includes search icons and 'Add' and 'Delete' buttons.
- TP ID Alias Definition:** Fields for '*Ext TPID' and '*Alias TPID' with search icons and 'Add' and 'Delete' buttons.

Ec Internal Partner Definition page

You assign trading partner IDs to the internal entities (business units) that external trading partners need to be able to submit transactions to. Because of the way each PeopleSoft Financials application tracks its own set of business units, a single trading partner ID can refer to multiple business unit IDs, all of which actually refer to the same physical business unit. Use this page to specify which internal business entities share this partner ID and which external trading partners do business with this partner.

To create an internal partner ID:

1. Specify to which business entity or entities this trading partner ID applies.

In the PS Code field, select the entity code for the type of business entity, and then select the specific entity in the Unit field.

Because the same business unit can be referred to by different business unit IDs in different PeopleSoft Financials applications, you can add multiple units to the same trading partner ID.

2. Specify which external trading partners work with this internal trading partner, and what name they use to refer to the internal partner.

In the Ext TPID field, select the name of an external trading partner that does business with the internal partner that you are defining. Then, in the Alias TPID field, enter the name by which the selected external partner refers to the internal partner, that is, the text that will appear in the Internal Trading Partner field in PeopleSoft business documents.

Note. If you haven't defined your external trading partners yet, you'll need to return to this page after you add them.

Repeat this procedure for each external trading partner.

Assigning External Trading Partner IDs

Access the Ec Ext Partner Def (External Partner Definition) page.

Ec External Partner Definition page

You'll need to find out from the trading partner or from the VAN what trading partner ID to use.

To add an external trading partner:

1. Select the map profile that includes the mappings that this trading partner uses.
Select the appropriate map ID from the list of valid map profile IDs.
2. Select the partner profile that identifies the transactions that this partner can perform.
Select the appropriate partner profile ID from the list of valid IDs.
3. (Optional.) Select the conversion data profile that specifies how to translate this partner's codes into PeopleSoft codes.
Select the appropriate conversion data profile ID from the Cvt Pro ID field.
4. Specify the customer ID, vendor ID, or other ID for this trading partner.

Note. If you exchange transactions with multiple business entities inside this trading partner, you'll need to define the partner's external business entities. If you define external business entities, each one needs to have its own unique customer ID, vendor ID, or other ID. The ID that you assign here should be the ID for the corporate office.

In the PS Code field, select the entity code for the type of entity this partner is: customer, vendor, and so on. If your system uses table set processing, select the setID for the Customer or Vendor table in the SetID field. In the PS Customer/Vendor Number field, select the specific value from the Customer or Vendor table that corresponds to this trading partner.

If different tables in the system refer to this trading partner by different names, you can list all the IDs as part of the trading partner definition.

5. Identify the internal trading partners who do business with this external trading partner.

In the Int TPID field, select the name of an internal trading partner that does business with the external partner you are defining. Then, in the Alias TPID field, enter the name by which the new external partner refers to the selected internal partner, that is, the text that will appear in the Internal Trading Partner field in the PeopleSoft business documents.

Repeat this step for each internal trading partner.

Defining Business Entities in External Companies

Access the Ec Bus Entity Def (Business Entity Definition) page.

Ec Bus Entity Def

External Business Entity ID: TTT

External Business Entity Definition

Description:

Parent Trading Partner ID:

Customer / Vendor Assignment Find | View All First 1 of 1 Last

*PS Code	SetID	*PS Customer/Vendor Number	Add	Delete
<input type="text"/>	<input type="text"/>	<input type="text"/>		

Associated Internal Entity Assignment Find | View All First 1 of 1 Last

*Internal Entity ID	*Alias Internal Entity ID	Add	Delete
<input type="text"/>	<input type="text"/>		

Ec Business Entity Definition page

Perform this procedure only if you need to be able to address transactions to multiple business entities that are part of the same external trading partner. This is similar to creating an external partner definition, except that business entities are subordinate to external trading partners and acquire some of their properties. Use this page to define the groups or entities that you deal with inside an external trading partner.

To define business entities in an external trading partner company:

1. Specify which trading partner company this business entity is a part of.

Select the appropriate trading partner ID from the Parent Trading Partner ID field.

When the EDI agent processes a transaction that involves the business entity that you are defining, it will use the map ID, profile ID, and conversion data ID assigned to the parent trading partner.

2. Specify the customer ID or vendor ID for this business entity.

If you want to establish a business entity as a valid trading partner, it must appear as a separate entry in your PeopleSoft database as a customer or a vendor. In the Customer/Vendor Assignment group box, you identify which customer or vendor is referred to by the external business entity ID that you are creating. The system needs this information to determine the customer ID or vendor ID to assign to incoming EDI transactions, and the external business entity ID to assign to outgoing EDI transactions.

In the PS Code field, select the entity code for customers or vendors. If your system uses table set processing, select the setID for the Customer or Vendor table in the SetID field. In the PS Customer/Vendor Number field, select the specific value from the Customer or Vendor table that corresponds to this business entity.

If different tables in the system refer to this business entity by different names, you can list all the IDs as part of the external business entity definition.

3. Identify the internal trading partners who do business with this external business entity, and then save the page.

In the Internal Entity ID field, select the name of an internal trading partner that does business with the external entity you are defining. Then, in the Alias Internal Entity ID field, enter the name by which the new external entity refers to the selected internal partner, that is, the text that will appear in the Internal Trading Partner field in PeopleSoft business documents.

Repeat this step for each internal trading partner.

Deleting PeopleSoft EDI Manager Objects

This section provides an overview of deleting PeopleSoft EDI Manager Objects and discusses how to delete PeopleSoft EDI Manager objects.

Understanding Deleting PeopleSoft EDI Manager Objects

You might want to delete business entities if they are no longer valid. PeopleSoft enables you to delete maps, trading partner profiles, data conversion profiles, internal trading partners, external trading partners, and so on.

Page Used to Delete PeopleSoft EDI Manager Objects

Page Name	Object Name	Navigation	Usage
Ec Delete	EC_DELETE	PeopleTools, EDI Manager, Setup Trading Partners, Remove EDI Manager Objects	Delete PeopleSoft EDI Manager objects.

Deleting EDI Manager Objects

Access the Ec Delete page.

Ec Delete

Objects To Be Deleted

Delete EC Map:	<input type="text"/>	<input type="button" value="Q"/>
Delete Trading Partner Profile:	<input type="text"/>	<input type="button" value="Q"/>
Delete Data Conversion Profile:	<input type="text"/>	<input type="button" value="Q"/>
Delete External TP ID:	<input type="text"/>	<input type="button" value="Q"/>
Delete Internal TP ID:	<input type="text"/>	<input type="button" value="Q"/>

Press SAVE To Commit Delete

Ec Delete page

To delete EDI Manager objects, select the objects to be deleted in the appropriate field list, and then click Save to delete the selected objects.

CHAPTER 3

Mapping EDI Transactions

This chapter provides an overview of Electronic Data Interchange (EDI) transaction mapping and discusses how to:

- Define inbound maps.
- Define outbound maps.
- Create data mapping profiles.
- Use PeopleSoft EDI Manager for general data extraction.

Understanding EDI Transaction Mapping

This section discusses:

- EDI transaction mapping.
- New EDI transactions.
- PeopleSoft business document format.

EDI Transaction Mapping

PeopleSoft applications store transaction data in tables in the PeopleSoft database. For an application to process an EDI transaction, it needs the transaction data in its tables. So the first step in processing an EDI transaction is transferring the data from an incoming EDI Transaction Set file into the application tables. Similarly, the first step in generating an EDI transaction for delivery to a trading partner is getting the transaction data out of the application tables into an EDI Transaction Set file.

You specify how the EDI agent transfers data between EDI Transaction set files and application tables by creating electronic commerce maps. Two kinds of electronic commerce maps are available:

- Inbound maps, which transfer incoming transaction data into your PeopleSoft database.
- Outbound maps, which create outgoing EDI documents from transaction data in the database.

Map profiles specify the maps that your trading partners have access to.

New EDI Transactions

We deliver PeopleSoft applications with a number of EDI transactions already defined. You can use PeopleTools to add support for additional transactions. Here are the major steps in the process.

To process new EDI transactions:

1. Use PeopleTools to develop the application that processes the transaction.

Create the database tables and the pages that users need to process the transaction.

2. Define a PeopleSoft Business Document format based on the EDI format for the transaction.

At this step, think about how the data from the EDI document needs to map to the PeopleSoft application tables.

3. Write a translation program that converts EDI documents into PeopleSoft Business Document format.

Usually, you will have an EDI translator program to do this step.

4. Create staging tables to serve as a temporary holding area between the PeopleSoft document and the application tables.

5. Create maps to copy data between the PeopleSoft business documents and the staging tables.

6. Write an application load procedure that transfers data from the staging tables to the application tables.

This chapter documents step 5 in this procedure.

PeopleSoft Business Document Format

A PeopleSoft business document is an ASCII file that contains the data for one or more EDI transactions. It contains transaction data and control information, which tells the system what type of transaction it is, who it came from, and how parts of the document relate to each other.

A PeopleSoft business document is divided into records, separated from each other with a carriage return/line feed (CR/LF) pair. Each record is divided into fixed-length fields. One of the fields in the record, usually the first one, is a record ID field that specifies the record type: a header, detail line, summary line, and so on. The remaining fields depend on the type of record.

The first record in a PeopleSoft business document is always a control record. The control record specifies the type of transaction contained in the following lines and the trading partners involved. Based on the control record, the EDI agent retrieves the mapping definition for the specified transaction and the data conversion options relevant to the trading partner.

The EDI agent reads the records in the document one at a time. For each record following the control record, the EDI agent reads the value in the record ID field and uses it to determine the PeopleSoft record definition to use to parse the rest of the record. The PeopleSoft record definition specifies the location, size, and data type of the remaining fields in the record. The EDI agent copies the data from the document into staging tables in the database, following the rules in the mapping definition.

When the EDI agent encounters a new control record, identified by the record ID *999* or *998*, it retrieves the mapping definition for the new transaction type and repeats the process. Within the flat file, the control records:

- Enable a third-party translator or trading partner, upon receiving the file, to determine what the data contains (*999* only).
- Enable PeopleSoft EDI Manager, upon receiving the file, to determine what the data contains (*999* and *998*).

Most transactions include a variety of record types: a header, detail lines, schedules, summary lines, and so on. The layout of the records within a PeopleSoft business document follows a logical pattern. All detail lines linked to a particular parent or header line must follow the parent record.

Note. The EDI agent processes transactions at the “unit of work” level, corresponding to the ST/SE level in X.12 format. Each transaction should be a unit of work that you want the EDI agent to commit or roll back as a unit, such as a single purchase order or invoice.

Let’s look at an example:

```

001 Header Line 1
002 Detail Line 1A
002 Detail Line 1B
002 Detail Line 1C001 Header Line 2
002 Detail Line 2A
002 Detail Line 2B

```

This file has two header lines and five detail lines. The EDI agent determines which detail lines go with which header lines based entirely on their order.

When the EDI agent processes this PeopleSoft business document, it works through the records in order. First, it processes header line 1 using the work record that's appropriate for record ID 001. Next, it processes detail line 1A using the work record for record ID 002.

If you ask the EDI agent to calculate a value, such as the sum of all the detail lines, it determines which detail lines to group together based on their position. It calculates the value for all detail lines *since the most recently encountered header line*.

999 Record Format

This reserved record identifier acts as a control record to switch map definitions within a data file. The layout looks like this :

Field	Description	Value
1-3	Record Identifier.	'999'
4-18	Transaction ID.	Char
19-22	External entity code.	Char / Values : CUST or VNDR
23-38	External trading partner ID.	Char
39-42	Internal alias entity code.	Char / Values : AP, AR, OM, ...
43-58	Internal trading partner alias ID.	Char

998 Record Format

This format allows the specification of a PeopleSoft map ID directly in the data contents. This type of map definition can be used when trading partner conversions are not needed or not available. All other data conversion functionality can be used.

Field	Description	Value
1–3	Record identifier.	'998'
4–13	PeopleSoft EC map ID.	Char
14–23	PeopleSoft trading partner profile ID.	Char
24–33	PeopleSoft data conversion profile ID.	Char

Defining Inbound Maps

This section describes how to:

1. Create staging tables and work records.
2. Create inbound maps.
3. Define the inbound PeopleSoft business document format.
4. Specify staging tables.

You create one map for each EDI transaction.

Note. Electronic commerce maps help transfer data between PeopleSoft business documents and staging tables in a PeopleSoft database. They don't apply to the application load or application extract processes, which copy data between the staging tables and the transactional tables.

Pages Used to Define Inbound Maps

Page Name	Object Name	Navigation	Usage
Inbound Maps - Description page	EC_MAP_00	PeopleTools, EDI Manager, Map EDI Transactions, Inbound Maps Select the Description tab.	Create inbound transaction maps.
Inbound Maps - Business Document Layout	EC_INBOUND_MAP_01	PeopleTools, EDI Manager, Map EDI Transactions, Inbound Maps Select the Business Document Layout tab.	Define business document layouts.
Inbound Maps - Target Records	EC_INBOUND_MAP_02	PeopleTools, EDI Manager, Map EDI Transactions, Inbound Maps Select the Target Records tab.	Specify staging tables.

Creating Staging Tables and Work Records

To create staging tables and work records:

1. In Application Designer, create the staging tables for the incoming data.

Save the record definitions with the extension `_EC` to identify them as staging tables. Use SQL Create to create tables using the record definitions.

The tables should be structured as an intermediate step between the PeopleSoft Business Document format that you want to translate and the structure of the transaction tables.

The record definition for a staging table has three kinds of fields:

- Required system fields, which must be key fields for the tables.

Field Name	Description
ECTRANSID	Transaction ID.
ECQUEUEINSTANCE	Instance ID.
ECTRANSINOUTSW	<i>I</i> or <i>O</i> , specifying an incoming or outgoing transaction.

- Fields corresponding to the data fields in the PeopleSoft business documents.
 - (Optional) Extra fields for calculated data needed in the application tables, such as summary values, date stamps, or action codes.
2. In PeopleSoft Application Designer, create work records that mimic the structure of the records in the PeopleSoft business document that you want to translate.

You need to create a work record for each physical record in the PeopleSoft business document file.

One field in the work record needs to be designated as the record ID for the document record that it mimics. For most records, it's the first field: the record ID in the PeopleSoft business document is usually the first data on the line.

The EDI agent uses the record ID in the work record to match the appropriate work record with each record in the PeopleSoft business document. You can add the appropriate record ID as a default value for the field; PeopleSoft EDI Manager will automatically supply the correct record ID when you describe the business document format on the Business Document Layout page.

The rest of the fields in the work record definition should match the corresponding fields in the PeopleSoft Business Document record. They should be in the same order and should have the same length. Add default values for any fields that have them.

Save the work record with the extension `_WD` to identify it as a work document.

See Also

[Chapter 3, "Mapping EDI Transactions," PeopleSoft Business Document Format, page 24](#)

Creating Inbound Maps

Access the Inbound Maps - Description page.

Inbound Maps - Description page

To create an inbound map:

1. In the EC Transaction ID field, select the transaction to which this map applies.
2. Enter a description of the map.

You can enter both a short description (in the Description text box) and a long description (in the larger box). These descriptions are for your information only; the system doesn't use them.

3. (Optional.) Copy the definition of a similar map.

You can copy the details of an existing map definition into this component and change just the parts that are different.

Select the existing map in the Source Map Definition field, and then click the Ec Copy Map DefnSwitch button. The system copies the information from the selected map into this page.

See Also

Chapter 2, “Using PeopleSoft EDI Manager,” page 3

Defining the Inbound PeopleSoft Business Document Format

Access the Inbound Maps - Business Document Layout page.

Inbound Maps - Business Document Layout page

You have already created work record definitions that mimic the structure of the PeopleSoft business documents. On this page, you specify which work record definition to use for each record in the PeopleSoft business documents, and what type of data appears in each field.

To define the inbound format:

1. Select a work record definition from the Business Doc Record field, and then click the EC Get File Fields button.

PeopleSoft EDI Manager copies data from the selected record definition into the Map group box, with one row for each field in the record.

2. Specify the record ID for this work record definition.

If the first field in the work record definition isn't the one for record IDs, browse the rows in the Map group box to find the record ID field. The EDI agent uses this field to match the work record definition to the appropriate records in PeopleSoft business documents.

- a. In the Map group box for the appropriate record ID, select the Record ID option in the Field Value Conversion group box. The EC File Row ID Value text box appears.
 - b. In the text box, enter the record ID of the records whose structure this work record definition mimics. When the EDI agent finds a record in a PeopleSoft business document with this record ID, it uses this work record definition to parse it.
3. Select the next field in the work record definition.
 4. Check the location of a field in the record.

If you defined the work record definition properly, the correct field information appears in the text boxes, copied from the work record. You can verify the existing information rather than entering it. If the format of the PeopleSoft business document is slightly different than the work record definition, you can modify the formatting on this page to match the document without affecting the work record definition.

The Seq (sequence) text box gives the number of the field in the record. The first field in the record is field 1, the second is field 2, and so on. The Name field is the field name from the work record, and the Field Type field indicates the data type of the field's data.

The Start and Length boxes specify the location and size of the field in the PeopleSoft business document. The start position is the number of characters from the beginning of the record to where the first character of this field's data appears. The length is the number of characters reserved for this field; it's also the number of characters that the system will transfer to the staging tables.

For numeric data, the Dec (decimal positions) field indicates the number of characters to the right of the decimal point. If an incoming PeopleSoft business document uses implicit decimal format, that is, if it doesn't include decimal points in this field (in any row), the EDI agent inserts them (in all rows) at the position specified in this box. If the data already includes decimal points, the EDI agent doesn't change them or insert any other decimal points.

For date information, you also need to enter a date format and delimiter. You use these text boxes to specify the format for dates in this field in the incoming file. Use standard date formatting conventions, as illustrated in the following table:

Date Value	Date Format	Delimiter
19961226	YYYYMMDD	N (for none)
1996/12/26	YYYYMMDD	/
DEC-96-26	MMYYDD	-
26-DEC-1996	DDMMYYYY	-
26.12.1996	DDMMYYYY	.

5. Select an option in the Special EDI Attribute group box to specify whether the data in the current field is transaction data or an EDI control code.

Options are:

<i>None</i>	Normal transaction data.
<i>EC Entity Code</i>	An entity code specifies the type of business entity that the trading partner ID for the current transaction refers to. The entity code tells the EDI agent that the table to use to find the list of valid IDs.
<i>EC Queue Control Number</i>	A unique identifying number that you want to store in the EDI agent queue and refer to when reviewing EDI audit history. For example, in a purchase order, you might identify the PO number as the EC queue control number.
<i>Pri Evt Cd (primary event code) and Sec Evt Cd (secondary event code)</i>	The primary event code, sometimes called the purpose code. The EDI agent uses event codes to determine the appropriate action code. If you identify a primary event code, you must also identify a secondary event code, sometimes called the transaction code, in the same record.

6. Specify how the EDI agent needs to convert the data as it copies data from the field into the staging tables. Field value conversion options are:

<i>None</i>	The EDI agent copies the data exactly as it appears in the PeopleSoft business document field.
<i>TP Convert</i>	The EDI agent converts data from this field according to the rules defined in a conversion type definition. When you select this option, a drop-down list box appears to select a conversion type definition.
Record ID	Use this option for the first field in the work record definition only. (See step 3.)
<i>Convert</i>	The EDI agent converts the data as specified in the From and To boxes. When the field in the PeopleSoft business document contains the value in the From box, the EDI agent converts it into the value in the To box. You need to enter a row for each possible value in the field.

Note. You can perform data conversion at several points during processing. Use the Convert option for conversions that apply to this map only and are valid for all trading partners.

- Repeat steps 4 through 6 for each field in the work record definition.
- Repeat steps 1 through 7 for each record in the PeopleSoft business document.

To associate another work record definition with this map, put the cursor in the Business Doc Record field and press the F7 key.

See Also

[Chapter 2, “Using PeopleSoft EDI Manager,” page 3](#)

Specifying Staging Tables

Access the Inbound Maps - Target Records page.

The screenshot displays the 'Inbound Maps - Target Records' configuration page. It features a navigation bar with tabs for 'Description', 'Business Document Layout', and 'Target Records'. The main area is divided into several sections: 'EC Map ID: AUDIT', 'Trans ID: TNT', and 'Descr:'. Below this is a 'File' section with 'File Row ID: 000' and 'File Layout Model: BUSPROC_BD_WW'. A 'Record' section shows '*Target Record: SDK_BUS_EXP_DT'. The 'Target Field Settings' section includes 'Sequence: 7', '*Field Name: SDK_EXPENSE_AMT', and 'Field Type: 3'. The 'Set Field Equal To' section has three radio buttons: 'File Field Value', 'Default Value', and 'EC Agent Calc'd EC Calc Cd: Total/Accumulate Summa'. The 'Related Record Info' section has 'Row:', 'Record:', and 'Field:' fields.

Inbound Maps - Target Records page

To specify staging tables:

1. Select a work record definition.

Browse the rows in the File group box to select a work record definition that you added to this mapping on the previous page. The record definition name appears in the File Layout Model field.

2. In the Target Record field, select the staging table to copy data into, and then click the EC Get Recfields SW button.

Field information is copied from the record definition into the page.

3. Select a field in the staging table record definition.

Browse the rows in the Target Field Settings group box to select the field.

The Sequence field gives the position of the field in the record. The field name is from the record definition, and the field type indicates the data type of the field's data. All of this information comes from the record definition that you selected.

4. In the Set Field Equal To group box, select the value that you want the EDI agent to copy into the selected field.

To copy the value from a field in the PeopleSoft business document, select the File Field Value option. Enter the name of the field whose value you want to copy in. Use the field name from the work record definition that appears as the file layout model at the top of the page.

To enter a default value into the table, select the Default Value option and enter the value in the text box that appears.

Note. You can specify both a file field to copy data from and a default value. The EDI agent enters the default value into the staging table only when the specified file field doesn't have a value.

To enter a value that the EDI agent calculates as it copies data into the staging table, select the EC Agent Calc'd (EC agent calculated) option, and then select a calculation option from the drop-down list box that appears. Options include:

Calculation Option	Value Written in the Field
<i>Action Code Conversion</i>	<p>An action code.</p> <p>The EDI agent gets the values from the fields in this record labeled Primary Event Code and Secondary Event Code, looks up the combination in the partner profile for the trading partner, and inserts the appropriate action code.</p>
<i>Apportion Parent Value</i>	<p>A percentage of the value from the parent line.</p> <p>The EDI agent takes the numeric value from the preceding parent line, divides it by the number of child lines, and puts the resulting value in this field on each child line.</p> <p>You have to complete out the Related Record Info group box to identify the parent line.</p>
<i>Average Summary Value</i>	<p>The average value from a particular field.</p> <p>The EDI agent adds up the values in this field from all occurrences of this record reporting to the same parent line, divides by the number of records, and enters the result in this field.</p> <p>You have to complete the Related Record Info group box to identify the field whose average you want.</p>
<i>Business Document Level External TPID (business document level external trading partner ID)</i>	<p>The trading partner ID of the trading partner that generated the EDI transaction.</p> <p>The EDI agent gets the trading partner ID from the first record in the PeopleSoft business document.</p>
<i>Business Document Level Internal TPID (business document level internal trading partner ID)</i>	<p>The trading partner ID of the internal group to which the EDI transaction is addressed.</p> <p>The EDI agent gets the trading partner ID from the first record in the PeopleSoft business document.</p>
<i>Current Date</i>	<p>The date on which the EDI agent processes the PeopleSoft business document.</p>
<i>Current Date and Time</i>	<p>The date and time when the EDI agent processes the PeopleSoft business document.</p>

Calculation Option	Value Written in the Field
<i>Current Time</i>	The time when the EDI agent processes the PeopleSoft business document.
<i>EC Queue Instance</i>	The system-generated queue instance ID given to this PeopleSoft business document. Note. To create a valid mapping, you must map this value to the ECQUEUEINSTANCE field.
<i>EC Transaction ID</i>	The transaction ID from the first record in the PeopleSoft business document. Note. To create a valid mapping, you must map this value to the EDTRANSID field.
<i>File Name/File ID</i>	The name of the PeopleSoft business document file.
<i>Incremented Key Sequence Number</i>	A system-generated sequence number. The EDI agent increments the number for each child row of a particular parent. It starts at 1 again when it reaches a new occurrence of the parent line. You have to complete the Related Record Info group box to identify the parent line.
<i>Inherit Parent Value</i>	A value copied directly from the parent line. You have to complete the Related Record Info group box to identify the parent line.
<i>Maximum Summary Value</i>	The maximum value of a field. The EDI agent checks all the occurrences of a specified record and field and copies the maximum value into this field. You have to complete the Related Record Info group box to identify the field whose maximum you want.
<i>Minimum Summary Value</i>	The minimum value of a field. The EDI agent checks all the occurrences of a specified record and field, and copies the minimum value into this field. You have to complete the Related Record Info group box to identify the field whose minimum you want.
<i>Operator ID</i>	The ID of the user who started the EDI agent.

Calculation Option	Value Written in the Field
<i>Process Instance</i>	The system-generated instance ID from PeopleSoft Process Scheduler.
<i>Run Control ID</i>	The run control ID of the run control used to start the EDI agent.
<i>Total/Accumulate Summary Value</i>	<p>The total of the values in a field.</p> <p>The EDI agent adds the values from a specified field in all occurrences of a record reporting to the same parent line and enters the result in this field.</p> <p>You have to complete the Related Record Info (related record information) group box to identify the field whose total you want.</p>
<i>Trading Partner Conversion</i>	The customer, vendor, or business unit that is associated with a trading partner ID.

5. Enter the related record information (for some calculated options only).

For some of the calculation options that you just selected, you identify the row, record, and field to use in the calculation.

For example, if you selected the Total/Accumulate Summary Value option, the EDI agent adds up the values from the specified row, record, and field.

Note. If you selected the Increment Key Sequence Number option, only the Row field appears. You don't need to select a record or field.

6. Repeat steps 3 through 5 for each field in the staging table.

To create a valid mapping, you must enter the appropriate values in the three required system fields:

- ECTRANSID: Use the EC Transaction ID option to have the EDI agent put the transaction ID in this field.
- ECQUEUEINSTANCE: Use the EC Queue Instance ID option to have the EDI agent put the instance ID in this field
- ECTRANSINOUTSW: Enter *I* to specify an incoming transaction.

7. Repeat steps 2 through 6 for each staging table that you want to copy data into.
8. Repeat steps 1 through 7 for each work record definition.

Defining Outbound Maps

This section discusses how to:

1. Create outbound maps.
2. Specify source records.

3. Define the outbound PeopleSoft Business Document format.

Pages Used to Define Outbound Maps

Page Name	Object Name	Navigation	Usage
Outbound Maps - Description	EC_OUTBOUND_MAP_01	PeopleTools, EDI Manager, Map EDI Transactions, Outbound Maps Select the Description tab.	Create an outbound transaction map.
Outbound Maps - Source Records	EC_OUTBOUND_MAP_01	PeopleTools, EDI Manager, Map EDI Transactions, Outbound Maps Select the Source Records tab.	Specify source records.
Outbound Maps - Target Business Doc Layout (Target Business Document Layout)	EC_OUTBOUND_MAP_02	Select the Target Business Layout Doc tab.	Define the outbound PeopleSoft Business Document format.

Creating Outbound Maps

Access the Outbound Maps - Description page.

Outbound Maps - Description page

The initial procedure for creating an outbound map is the same as that for creating an inbound map.

See [Chapter 3, “Mapping EDI Transactions,” Creating Inbound Maps, page 28.](#)

Specifying Source Records

Access the Outbound Maps - Source Records page.

Outbound Maps - Source Records page

To specify source records:

1. In the Source Record field, select the record definition that retrieves the data that you want from the staging tables.

You might select the record definition used to define the table or a view that retrieves just the data that you want.

Add lines to the PeopleSoft business document in a logical unit of work order: header information before detail line information, parent lines before child lines, and so on.

Note. You don't need to create a control record for the PeopleSoft business document (the 999 or 998 record). The EDI agent creates one automatically from information in the queue that stores pending outbound transactions.

2. Select a parent record (child lines only).

If the record definition that you selected in the previous step retrieves data for a record that is related to a preceding parent record, such as the detail line under a header, select the record definition for the parent record in the Parent Record field.

The parent record is typically the record definition that you used for the previous row in the outbound mapping.

3. Enter the SQL Where clause to specify which rows to retrieve.

When you're creating a header line, the Where clause needs to select the header information for a single transaction. You identify a single transaction using the transaction ID and queue instance ID. The Where clause should look like this:

```
WHERE ECTRANSID = $ECTRANSID$ AND ECQUEUEINSTANCE = $ECQUEUEINSTANCE$
```

The two values inside dollar signs are system variables that the EDI agent replaces with the transaction ID and queue instance ID, respectively, of the first transaction in the pending outbound queue. This Where clause retrieves only the data for one transaction.

If you're extracting the data for child lines, the Where clause must include a link to the parent record to tell the EDI agent how to join the tables. The Where clause needs to include a statement of this form:

```
WHERE fieldname = $record.fieldname$
```

The first *fieldname* is a field from the source record definition; *record.fieldname* is a field in the parent record definition. You must include the dollar signs around *record.fieldname*.

- Repeat steps 1 to 3 to specify the record definition for each record to go into the PeopleSoft business document.

Defining the Outbound PeopleSoft Business Document Format

Access the Outbound Maps - Target Business Doc Layout page.

Outbound Maps - Target Business Doc Layout page

For each of the source record definitions that you added on the previous page, you identify an associated work record definition that specifies how the EDI agent writes the staging table data into PeopleSoft business documents.

To define the outbound format:

1. Select the work record definition that specifies the format of the output record, and then click the Get File Fields SW button.

The File Row ID field identifies a source record definition from the previous page. In the Model File Layout field, select the work record definition that the EDI agent will use to write data from the source record into PeopleSoft business documents.

When you click the Get File Fields SW button, the system copies field information from the work record definition into the page. The *Target Record Field Info* group box displays information about where in the record it will write each field's data.

Depending on the data type of the record field, the Target Record Field Info group box offers additional formatting options so that you can format the data in the outgoing PeopleSoft business document.

- For any field, enter in the Strip Chars field any characters that you don't want the outbound EDI agent to include when it copies data into the outbound transaction file. For example, you may want to strip the hyphens from phone numbers and Social Security numbers, or the decimal from a currency amount. When the outbound EDI agent writes data from a field into an outbound transaction file, it first removes all characters in the Strip Chars field.

Note. The Strip Chars field contains a list of the individual characters to remove, not a string to remove. For example, if the field data is *455-67-8898* and the Strip Chars field has - (a hyphen) in it, the EDI Agent writes *455678898* to the file. If the Strip Chars field has the string *-5* in it, the EDI Agent writes the field data to the file as *4678898*.

- For character fields, select the Convert to Upper Case check box to tell the outbound EDI agent to write the contents of the field in all uppercase letters.
- For character fields or numeric fields, use the Pad Character field to specify a character for the outbound EDI agent to use to pad data so that it is the full length of the field. For character fields, the outbound EDI agent adds the specified character to the left of the existing field information. For numeric fields, the most typical use is to pad a number with initial zeros.
- For date fields, you can specify in the Date Fmt (date format) and Delimiter fields how to format dates in this field in the business document. Use standard date formatting conventions, as illustrated in the procedure for creating inbound maps.

See [Chapter 3, "Mapping EDI Transactions," Defining the Inbound PeopleSoft Business Document Format, page 29.](#)

2. Specify the data that you want the EDI agent to enter in the field.

In the Conversion Processing group box, select the value that you want the EDI agent to enter into the selected field.

- To copy the value from a field in the source record, select the Source Field option. Enter the name of the field whose value you want to copy in.

Note. You need to specify a source field for all fields in the work record definition except those for which you provide a default value. You must select a source field even if you select another option as well.

- To enter a default value into the document, select the Default option and enter the value in the field that appears.

You can specify both a file field to copy data from and a default value. If you do, the EDI agent enters the default value into the document only when the specified field doesn't have a value.

- If you select TP Conversion,, the EDI agent converts data from the specified source field according to the rules defined in a conversion type definition. When you select this option, a drop-down list box appears to select a conversion type definition.
- If you select Convert, the EDI agent converts the data in the source field as specified in the From and To fields that appear. When the field in the PeopleSoft business document contains the value in the From field, the EDI agent converts it into the value in the To box. Enter a row for each possible value in the field.

- To enter a value that the EDI agent calculates as it copies data into the staging table, select the Agent Value option, and then select a calculation option from the drop-down list box that appears. The following table describes the available calculation options.

Calculation Option	Value Written in the Field
<i>Action Code/Primary Event Code</i>	An action code. The EDI agent gets the values from the fields in this record labeled Primary Event Code and Secondary Event Code, looks up the combination in the partner profile for the trading partner, and inserts the appropriate action code.
<i>Current Date</i>	The date on which the EDI agent creates the PeopleSoft business document.
<i>Current Datetime</i>	The date and time when the EDI agent creates the PeopleSoft business document.
<i>Current Time</i>	The time when the EDI agent creates the PeopleSoft business document.
<i>EC Entity Code Flag</i>	Flags this field as an entity code field to use in determining trading partner conversion logic. The values specify whether the entity is a customer, vendor, or business unit.
<i>EC Queue Instance</i>	The system-generated queue instance ID given to this PeopleSoft business document.
EC Trans ID (EC transaction ID)	The transaction ID.
<i>Incremented Sequence Nbr</i> (incremented sequence number)	A system-generated sequence number. The EDI agent increments the number for each child row of a particular parent. It starts at 1 again when it reaches a new occurrence of the parent line.
<i>Secondary Event Code</i>	The secondary event code from the action code conversion.
<i>Trading Partner Conversion</i>	The trading partner ID associated with the customer, vendor, or business unit.

- Move to the next field in the record definition and repeat step 2.
- Scroll to the next source record definition and repeat steps 1 through 3.
- Save the map.

6. Run the Outbound EDI Agent Preparer to create the Outbound EDI Agent SQC.

After you create or modify outbound maps, you need to run a special compilation process that prepares the maps for use by the EDI agent.

See [Chapter 4, “Monitoring EDI Processing,” page 47](#).

Note. Run the Outbound EDI Agent Preparer only after you’ve made all your changes to outbound maps. You only need to run it only once to prepare all outbound maps.

See Also

[Chapter 3, “Mapping EDI Transactions,” Defining the Inbound PeopleSoft Business Document Format, page 29](#)

Creating Data Mapping Profiles

This section provides an overview of data mapping profiles and discusses how to create data mapping profiles.

Understanding Data Mapping Profiles

When you add a trading partner, you assign to it a *map profile*, which lists the electronic commerce maps that the EDI agent can use to process transactions from the partner. The map profile serves two purposes:

- It restricts a trading partner’s access to transactions that they aren’t authorized to exchange with your company.
- By assigning different map profiles to different trading partners, the EDI agent processes the same transaction differently for different partners.

Page Used to Create Data Mapping Profiles

Page Name	Object Name	Navigation	Usage
Ec Map Profile Def (Map Profile Definition)	EC_MAP_PROFILE_DEF	PeopleTools, EDI Manager, Map EDI Transactions, Data Mapping Profiles	Create data mapping profiles.

Creating Data Mapping Profiles

Access the Ec Map Profile Def (Map Profile Definition) page.

Ec Map Profile Def

EC Map Profile ID: TTT

Descr:

Map Assignment Find | View All First 1 of 1 Last

*EC Map ID	Description	EC Transaction ID
------------	-------------	-------------------

Ec Map Profile Def page

To create a map profile, add the maps for all the transactions that you want partners with this map profile to use. Select a map ID in the EC Map ID field, and then add a new row to add another map. Don't forget to add both inbound and outbound maps.

Using PeopleSoft EDI Manager for General Data Extraction

The primary purpose of PeopleSoft EDI Manager is to create PeopleSoft business documents, which contain business transaction data and are subsequently translated into X.12 or EDIFACT format and transmitted to a trading partner. However, you can also use it to extract data from database tables into a text file.

To create a text file from PeopleSoft database data:

1. Using Application Designer, create a view that extracts the data that you want and includes three EDI control fields.

For example, to print a list of the countries in COUNTRY_TBL, create a view with the three EDI control fields described in the following table, plus the fields that you want from COUNTRY_TBL, perhaps Country and Description.

The EDI control fields must be the first three fields in the view. The fields are:

Field	Description
ECTRANSID	The transaction ID for the transaction that you want to perform. You'll probably want to define a special transaction ID for this data extraction transaction.
ECQUEUEINSTANCE	The instance ID.
ECTRANSINOUTSW	Enter a value of <i>O</i> to specify an outgoing transaction.

The Select statement for the view looks something like this:

```
SELECT
  'CNTRY' ,
  0 ,
  'O' ,
  COUNTRY ,
  DESCR
FROM
  COUNTRY_TBL
```

In this example, the transaction ID is *CNTRY*, the queue instance ID is always *0*, and the inbound/outbound switch is always *O* (outbound). Because the queue instance is the same for all records, the EDI agent processes them as a single transaction.

Note. For this example to work, you must use PeopleSoft EDI Manager to define the transaction ID *CNTRY*.

2. Create a record definition that specifies the layout of the PeopleSoft business document file.

This record definition typically consists of the ECFILEROWID field and all the fields from the table (for the example, COUNTRY and DESCR).

3. Define the outbound map.

In the example, you would specify *CNTRY* as the transaction ID on the first page in the Outbound Map Definition component. On the second page, the source record is the view that you created in step 1 and the Where clause is:

```
WHERE ECTRANSID = 'CNTRY'
```

On the third page, the model file layout is the record definition that you created in step 2. After you've selected the record definition from the list box, click the lightning bolt button to copy the field information into the page. You can use the default values for all the fields except ECFILEROWID. For ECFILEROWID, select Default in the Conversion Processing group box and enter whatever value you want to appear in the first column of the text file.

4. Run the Outbound EDI Agent Preparer process.
5. Add an entry to the ECQUEUE table.

The EDI agent uses the ECQUEUE table to determine the transactions that are ready to be processed. The fields in the table are.

ECTRANSID	The transaction ID, which tells the EDI agent which map definition to use. In the example, the transaction ID is <i>CNTRY</i> .
------------------	---

ECQUEUEINSTANCE	The instance ID. The example uses a single instance ID— <i>0</i> (zero)—for all the data.
ECTRANSINOUTSW	The inbound/outbound switch. Extracting data is an outbound transaction, so the value is <i>O</i> (the letter <i>O</i>).
ECBUSDOCID	The business document ID. Because you are not creating an EDI transaction, you can specify <i>0</i> (zero) in this field.
ECQUEUESTATUS	The status of the current record. The EDI agent processes records with the status <i>L</i> .

The remaining fields in the table—**BUSINESS_UNIT**, **ECENTITYCD_BU**, **ECCUSTVNDRVAL**, and **ECENTITYCD_EXT**—all relate to trading partner information. Because you are not creating an EDI transaction, you can use generic partner information. Use the default value **GENR** in all these fields.

- Schedule the outbound EDI agent to run.

The outbound EDI agent extracts your data and processes any other transactions on the **ECQUEUE** table with the status *L*.

The status field **ECQUEUESTATUS** changes to *P* (Processed). To repeat the extraction, change the status back to *L*.

CHAPTER 4

Monitoring EDI Processing

This chapter provides an overview of Electric Data Interchange (EDI) agents and discusses how to:

- Prepare outbound maps for EDI agents.
- Schedule inbound EDI agents.
- Schedule outbound EDI agents.
- Review errors and summary data.

Note. You must run the outbound map preparation process before you can process any outgoing EDI transactions.

See Also

[Chapter 3, “Mapping EDI Transactions,” Defining Outbound Maps, page 35](#)

Understanding EDI Agents

EDI agents copy EDI transaction data between PeopleSoft business documents and the PeopleSoft database, using the maps that you defined in the previous chapter. Two types of EDI agents are available:

- Inbound EDI agents, which process incoming transactions by copying data from PeopleSoft business documents into the database.
- Outbound EDI agents, which create PeopleSoft business documents from transaction data in the EDI staging tables.

If your system processes EDI transactions, you’ll usually want to have two EDI agents running on a regular schedule: one inbound agent and one outbound agent. You can also start other EDI agent instances for special purposes, such as processing the corrected version of a PeopleSoft business document that failed the first time.

Preparing Outbound Maps for EDI Agents

This section provides an overview of preparing outbound maps for EDI agents and discusses how to prepare outbound maps.

Understanding Preparing Outbound Maps for EDI Agents

When you create an outbound map definition, you define what data the EDI agent needs to extract from the staging tables. In order to select data from the tables, the EDI agent needs to run SQL statements whose details depend on the map definition.

Whenever you create or modify an outbound map definition, you need to run a preparation process that generates the appropriate SQL statements and compiles them into a format that the EDI agent can use. This process creates an SQC file.

Note. You must run the Outbound EDI Agent Preparer process before you can process any outbound EDI transactions. PeopleSoft doesn't deliver a compiled version of the drivers; you have to compile them for your system.

Page Used to Prepare Outbound Maps for EDI Agents

Page Name	Object Name	Navigation	Usage
Prepare Outbound Driver - Run Parameters	EC_RUN_OUTPREP_01	PeopleTools, EDI Manager, Monitor EDI Processing, Prepare Outbound Driver	Generates the appropriate SQL statements and compiles them into a format that the EDI agent can use to use outbound maps.

Preparing Outbound Maps

Access the Prepare Outbound Driver - Run Parameters page.

Run Parameters

Run Control ID: TTT [Report Manager](#) [Process Monitor](#)

ECOUTMAP.SQC Directory:

Map Selection Criteria

Trans ID:

Find | View All First 1 of 1 Last

Select	Transaction ID	Map ID	Description
<input type="checkbox"/>			

Prepare Outbound Driver - Run Parameters page

To prepare outbound maps for the EDI agent:

1. Specify where to put the SQC that contains the outbound map definitions.

In the ECOUTMAP.SQC Directory field, enter the directory where you want PeopleSoft Process Scheduler to put the SQC. The directory needs to be in the SQR search path of the workstation or server that will run the outbound EDI agent.

2. Click the Run button.

The Process Scheduler Request page appears.

3. Specify the server where you want the SQR compiler to run.
4. Specify when you want the process to run.

You'll probably want to run the preparation process just once (then run it again after changing or adding maps).

5. Click the OK button to run (or schedule to run) the outbound map preparation.

See Also

Enterprise PeopleTools 8.46 PeopleBook: PeopleSoft Process Scheduler, "Submitting and Scheduling Process Requests"

Scheduling Inbound EDI Agents

Select PeopleTools, EDI Manager, Monitor EDI Processing, Schedule Inbound EC Agent to access the Run Control Parameters page.

Run Control Parameters

Run Control ID: TTT [Report Manager](#) [Process Monitor](#)

Run Option <input checked="" type="radio"/> File List Driven <input type="radio"/> Single File <input type="radio"/> Single Instance	Inbound Agent Parameters File List Path: <input type="text"/> File List Name: <input type="text"/>	
Force Profile <input checked="" type="radio"/> Do Not Force (998 or 999 in file) <input type="radio"/> Force with Map Information (998) <input type="radio"/> Force with Partner Information (999)	Inbound Agent Forced Parameters <input type="text"/>	File Options <input type="checkbox"/> Suppress Rowid <input type="checkbox"/> Comma Separated Format

Run Control Parameters page

To schedule the inbound EDI agent to run:

1. Select the run option that the EDI agent uses to determine the documents or transactions to process.

File List Driven	Select to process all the PeopleSoft business documents whose filenames appear in a list file that the EDI translator software creates. The filenames must include their complete paths. Enter the directory that contains the PeopleSoft business document files and the list file in the File List Path field, and the filename of the list file in the File List Name field.
Single File	Select to process all the transactions in a specified PeopleSoft business document. Enter the directory that contains the file in the Single File Path field and the PeopleSoft business document file name in the Single File Name field.
Single Instance	Select to process a single transaction. Select the business document ID, transaction ID, and queue instance ID of the transaction that you want to process, in that order.

Note. The Single Instance option is available only for transactions that the EDI agent has already attempted to process. The transaction needs to have an EC queue instance ID.

Note. If the EDI agent runs on a server under MVS, it ignores the File List Path or Single File Path setting. Instead, it looks in the Job Control Language (JCL) for a DDNAME matching the specified file name. The File List Path option is not very useful for MVS platforms, because each file in the list has to have a DDNAME..

2. Specify how the inbound agent determines which map definition to use.

By default, the inbound agent looks for records with the row ID 999 to determine what type of transaction it's processing, and therefore which map definition to use. However, you can also specify a transaction map or trading partner for all transactions that this inbound EDI agent processes, so the inbound EDI agent can process files that don't include control records.

If you select the Do Not Force option, the inbound EDI agent looks for a control record at the beginning of each transaction in the incoming data file. You don't need to complete any other parameters.

If you select Force with Map Information, the Inbound Agent Forced Parameters box displays list boxes to select the transaction map to use for all transactions in the incoming file.

If you select Force with Partner Information, you can select the partner information to use for all transactions in the incoming file.

3. Select the file option that the EDI agent uses to display file data.

Suppress Rowid: Select to map homogeneous files to the same table. A row ID of 000 must be specified in the map definition; however, the flat file need not have a record identifier.

Comma Separated Format: Select to import files of variable length fields into PeopleSoft tables. This option is only available when the profile is forced with either 999 or 998 information.

Note. The separator should appear between each field in the file. The delimiter surrounds each field.

4. Click the Run button.
The Process Scheduler Request page appears.
5. Specify on what server you want the agent to run.

Note. In a production workflow, you'll usually want to run the EDI agent on a dedicated server.

6. Specify when, and how often, you want the agent to run.
If you're running the EDI agent once, enter the date and time when you want to run it.
If you're running the agents on a server, PeopleSoft Process Scheduler can run them once or periodically on a specified schedule.

Note. The Run Output option is not relevant for the EDI agent.

7. Click the OK button to run (or schedule to run) the EDI agent.

See Also

Enterprise PeopleTools 8.46 PeopleBook: PeopleSoft Process Scheduler, "Submitting and Scheduling Process Requests"

Scheduling Outbound EDI Agents

Select PeopleTools, EDI Manager, Monitor EDI Processing, Schedule Outbound EC Agent to access the Outbound EC Agent - Run Parameters page.

Run Parameters

Run Control ID: TTT [Report Manager](#) [Process Monitor](#) Run

Outbound Agent Run Parameters

EC Trans ID

Business Unit

Vendor/Customer

Outbound File Options

<input type="checkbox"/> Suppress EC 999 Record	<input type="checkbox"/> Suppress Rowid	<input type="checkbox"/> Single Document File
<input type="checkbox"/> Keep Queue Status	<input type="checkbox"/> Message Header/Footer	<input type="checkbox"/> Separate Output Flag
<input type="checkbox"/> Comma Separated Format		

Outbound EC Agent - Run Parameters page

To schedule the outbound EDI agent to run:

1. Specify the transactions that you want the EDI agent to process.

To have the EDI agent process all pending transactions, leave all three check boxes cleared. A pending transaction is one that an application extract process has added to the staging tables, and that has a status of *L* (Loaded).

To process all transactions of a particular type, select the EC Trans ID check box, then select the transaction ID from the drop-down list box that appears.

To process all transactions from a particular business unit, select the Business Unit check box and enter the business unit ID in the drop-down list box that appears.

To process all transactions for a particular trading partner, select the Vendor/Customer check box, and then enter the vendor ID, customer ID, or other internal ID for the desired partner.

You can select multiple check boxes. For example, you could process all outgoing invoices for a particular vendor by selecting both the EC Trans ID and Vendor/Customer check boxes.

2. Specify how you want to format the outgoing transaction file.

By default, the outbound EDI agent adds a record with the row ID 999 at the beginning of each transaction. This record identifies the transaction type in a way that is recognized by the PeopleSoft inbound EDI agent. If you select the Suppress EC 999 Record check box, the outbound EDI agent does not include this record in the outgoing file.

If you don't want to include the row ID for each record in the outgoing file, select the Suppress Rowid check box.

By default, the outbound EDI agent creates outbound transaction files in which the data for each field appears in the fixed column position specified in the outbound map definition. If you select the Comma Separated Format check box, the outbound EDI agent instead creates transaction files in which the fields are separated by a comma or other character rather than always appearing in a specified column position. The comma-separated option enables you to select the file options to read in flat file data that is comma-delimited.

Note. The Comma Separated Format option is only available if you used forced agent mapping. Selecting the Suppress Rowid option enables you to load data without requiring a row ID. However, your map definition must contain a row ID even if you are suppressing it in the run control setup. The row ID must be *000*.

The CSV separator is the character that the EDI agent uses to separate the fields in a record. The CSV delimiter is the character that the EDI agent uses to mark the beginning and end of a data value. In the following example, the separator is a comma, and the delimiter is a single quotation mark.

'WOW', '23.23', 'Big Deal'

Note. When you use comma-separated format, the outbound EDI agent ignores the column positions specified in the outbound map definition. It adds data to the outbound file in the exact order in which the fields appear in the map definition. You must make sure that the order of rows map definition specifies the fields in the order that you want them to appear in the file.

Select Single Document File to allow the results of the EDI to be written to a single output file. The business document ID is incremented as though multiple documents were created. The file may contain multiple 999 records.

Select Keep Queue Status to not update the ECQUEUE.ECQUEUESTATUS field at the completion of processing. This allows the record in ECQUEUE to be processed multiple times.

Select Message Header/Footer to process header and footer information prior to and following each business document. Header and footer information is stored with the trading partner definition.

3. Click the Run button.
4. Specify on which server you want the agent to run.
5. Specify when, and how often, you want the agent to run.

If you're running the EDI agent once, enter the date and time when you want to run it.

If you're running the agents on a server, Process Scheduler can run them once or periodically on a specified schedule.

6. Click the OK button to run, or schedule to run, the EDI agent.

See Also

Enterprise PeopleTools 8.46 PeopleBook: PeopleSoft Process Scheduler, "Submitting and Scheduling Process Requests"

Reviewing Errors and Summary Data

This section discusses how to:

- Review and correct errors.
- Review packages and transaction groups.

Reviewing and Correcting Errors

PeopleSoft delivers a predefined business process named Manage EDI that routes information about any processing errors it encounters to the EDI Coordinator's worklist. The coordinator can select the item from the worklist to review the error.

To review and correct EDI processing errors:

1. Select PeopleTools, EDI Manager, View EDI Audit Trail, Business Document Errors.

The Business Document Detail page appears.

This page gives detailed information about the processing of each line in a PeopleSoft business document.

2. Review the processing details for the document.

When you first open the page, the first row shows the processing information for the first record in the file, which is a 999 record. Browse the rows to view the other file records. The status of each row appears in the Status field.

3. (Optional.) Fix the erroneous data in the PeopleSoft business document.

Continue with the rest of the steps only if you fix errors.

Note. Because this page enables you to change transaction data, you'll want to carefully control access to it. In Operator Security, give only the EDI coordinator rights to edit this page.

4. Save the page.
5. Select PeopleTools, EDI Manager, View EDI Audit Trail, Transaction Maintenance.
6. Reset the EC queue status for the corrected transactions to *L*.

When the EDI agent encounters an error while processing a transaction, it sets the status field in the pending transaction queue (ECQUEUE) to *E*. Resetting the status of a transaction to *L* makes the outbound EDI agent try to process it again.

Reviewing Packages and Transaction Groups

The EDI transaction set file that you receive over the network from an external trading partner might include any number of transactions bound for any number of internal trading partners. To manage all this transaction data, the file organizes it into several levels.

- 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.
- 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.
- Each transaction group includes one or more individual units of work. A unit of work is a single transaction that you commit or roll back as a whole.

When third-party-supplied EDI translation software creates PeopleSoft business documents, it divides up the transaction set file at approximately the transaction group level. A PeopleSoft business document typically includes multiple units of work, and may include multiple transaction groups depending on how the translation software is set up.

PeopleSoft business documents do not contain information about what packages or transaction groups their transactions come from. However, the EDI translation software can provide this information in a separate audit file, and the EDI agent can copy the information into audit tables in your database.

Note. For the auditing information to be available, the EDI translation software must write the package and transaction group information to an audit file, and the EDI agent must process this file using the predefined Audit inbound map.

To view summary information about a PeopleSoft business document:

1. Select PeopleTools, EDI Manager, View EDI Audit Trail, Business Document Summary.
2. Enter the business document ID of the document that you want to review.

This page displays information about how the EDI agent processed the specified PeopleSoft business document. If the document included multiple transactions, browse the rows on the lower part of the page to see the information about each individual transaction.

3. Review the packages and transaction groups that this document is part of.

To view summary information about an EDI package:

1. Select PeopleTools, EDI Manager, View EDI Audit Trail, Package Log Summary.
2. Enter the package control ID for the package whose status you want to review.

Browse the rows on the lower part of the page to view the transaction groups that were inside the package and the PeopleSoft business documents that were created from the data in the package.

To view summary information about an EDI transaction group:

1. Select PeopleTools, EDI Manager, View EDI Audit Trail, Transaction Group Log Summary.
2. Enter the transaction group control ID for the group whose status you want to review, or press ENTER to select from a list of the groups.

CHAPTER 5

Using Forms Routing and the Outgoing Forms API

This chapter provides an overview of forms routing and discusses:

- Session-level operations.
- Query operations.
- Send operations.

Understanding Forms Routings

With PeopleSoft Application Designer, you can implement *routings*, which transfer data from one step in a business process to another. One type of routing is a *forms routing*. With a forms routing, the system takes data from a PeopleSoft page that the user is working on, enters it onto a third-party form, and mails the completed form to designated users by means of the forms product's mail capabilities.

Note. You can also integrate electronic forms software so that users can send forms to PeopleSoft applications using PeopleSoft Component Interfaces.

To design a forms routing, you map the fields in a PeopleSoft page to the fields in the form that it wants to generate. When a user triggers the forms routing, the system copies the data from the page fields to the corresponding form fields, then sends the form.

See Also

Enterprise PeopleTools 8.46 PeopleBook: PeopleSoft Component Interfaces, “Introducing Component Interfaces”

PSFORMS.DLL

PeopleSoft applications communicate with the forms software using PSFORMS.DLL. To integrate a forms product with PeopleSoft applications, create a version of PSFORMS.DLL that supports the function calls that the PeopleSoft application uses to generate and send forms.

The function descriptions in this chapter describe the function calls that PeopleSoft applications make to PSFORMS.DLL and the responses that it expects to the calls.

Because PSFORMS.DLL should work at any installation, don't build in any site-specific information, such as the name of the forms server and database that the PeopleSoft applications access. For site-specific information, add entries to the Microsoft Windows registry. PSFORMS.DLL can read the entries to identify the forms server, database, and any other site-specific information that it needs.

Note. PeopleSoft supports any VIM- or MAPI-compliant email package. No further development is necessary.

Session-Level Operations

This section discusses the session-level operations in alphabetical order.

PsfCloseSession

Syntax

```
int FAR PASCAL PsfCloseSession(HSESSION hSession);
```

Description

Closes a session.

Parameters

Parameter	Description
hSession	The session handle assigned by PsfOpenSession when PeopleSoft connected to PSFORMS.DLL.

Returns

This operation returns the following values:

Value	Code	Meaning
PSF_OK	0	The session was successfully closed.
PSF_NOSESS	4	<i>hSession</i> doesn't identify a current session.

PsfGetAPIInfo

Syntax

```
int FAR PASCAL PsfGetAPIInfo(LPPSFDEFNKEY lpDefKey);
```

Description

Provides the name of forms product that this DLL supports and returns the API version.

PeopleSoft uses the same programming interface for all forms products. It uses the PsfGetAPIInfo function to get the product name of the forms software that your version of PSFORMS.DLL supports.

PsfGetAPIInfo puts the product name as a string in the name field of the structure.

Parameters

Parameter	Description
lpDefKey	<p>A pointer to a structure containing a character array for the function to which to write. The form of the structure is:</p> <pre>typedef struct { char name [PSF_FIELDNAMELEN + 1]; } PSFDEFNKEY;</pre>

Returns

An integer identifying the version of this API that the DLL implements. The value is set in the PeopleSoft forms interface header file.

PsfOpenSession

Syntax

```
int FAR PASCAL PsfOpenSession(LPSTR lpzUserName, LPSTR lpzPassword,
LPHSESSION lphSession);
```

Description

Connects the PeopleSoft system to the forms interface DLL using a specified login name and password.

PsfOpenSession logs in to the form software using the specified user name and password. It places a session handle in the buffer that *lphSession* points to. All subsequent API calls use this session handle.

PsfOpenSession always opens a new session, even when it could access a shared session.

Parameters

Parameter	Description
lpzUserName	A null-terminated string containing the user name to log in to the forms software with.
lpzPassword	A null-terminated string containing the password for the user identified by <i>lpzUserName</i> .
lphSession	A pointer to a buffer for the session handle that this function assigns.

Returns

This operation returns the following values:

Value	Code	Meaning
PSF_OK	0	The session was successfully opened.

Value	Code	Meaning
PSF_NOSESS	4	The function was unable to establish a session.
PSF_NOSPEC	6	The forms database or server was not specified.

Query Operations

This section discusses the query operations in alphabetical order.

PsfGetFieldCount

Syntax

```
int FAR PASCAL PsfGetFieldCount(HSESSION hSession, LPPSFFORMDEFN lpFormData);
```

Description

PsfGetFieldCount gives the number of fields in a form by entering an integer in the wNumFields field of the lpFormData structure. The name of the desired form is in the structure's formName field.

The PeopleSoft system uses the field count to allocate an array large enough to hold all the fields. It passes a pointer to this array to PsfGetFieldList.

Parameters

Parameter	Description
hSession	The session handle assigned by PsfOpenSession.
lpFormData	A pointer to a structure that contains form information. The form of the structure is: <pre>typedef struct { PSFDEFNKEY formName; // Name of the form WORD wNumFields; // Number of fields char cSep; // Separator character LPPSFFIELDVAL lpFields; // Pointer to field array } PSFFORMDEFN;</pre>

Returns

This operation returns the following values:

Value	Code	Meaning
PSF_OK	0	The function ran successfully.
PSF_NOMAIL	2	Mail system failure.
PSF_NOFORM	3	The specified form is not accessible.
PSF_NODB	5	The forms database is not accessible.

PsfGetFieldList

Syntax

```
int FAR PASCAL PsfGetFieldList(HSESSION hSession, LPPFSFORMDEFN lpFormData);
```

Description

Provides a list of the fields on a form.

PsfGetFieldList provides the PeopleSoft application with information about the fields on a form. The name of the desired form is in the structure's formName field. The structure's lpFields field points to the array of structures that PsfGetFieldList updates. Each item in the array has this structure:

```
typedef struct
{
    char szFldName[PSF_FORMNAMELEN + 1];    // Field name
    WORD wFldSize;    // Field size
    LPSTR lpszStrVal;    // Field value
} PSFFIELDVAL;
```

PsfGetFieldList enters data into the szFldName and wFldSize fields. The PeopleSoft system provides the field values before passing the structure to PsfSendForm.

PsfGetFieldList includes all fields required to send forms using the forms product, regardless of whether they appear on the user's form. For example, if a user's form contains a TO field but no CC field, PsfGetFieldList includes both fields, because a CC field is required to transmit a mail message. If the forms mail system supports and requires attributes such as sensitivity, return receipt, and priority, PsfGetFieldList also returns these items as fields.

Parameters

Parameter	Description
hSession	The session handle assigned by PsfOpenSession when PeopleSoft connected to PSFORMS.DLL.
lpFormData	A pointer to a structure that contains form information. The form is: <pre>typedef struct { PSFDEFNKEY formName; // Name of the form WORD wNumFields; // Number of fields char cSep; // Separator character LPPSFIELDVAL lpFields; // Pointer to array } PSFFORMDEFN;</pre>

Returns

This operation returns the following values:

Value	Code	Meaning
PSF_OK	0	The function ran successfully.
PSF_NOMAIL	2	Mail system failure.
PSF_NOFORM	3	The form is not accessible.
PSF_NOODB	5	The forms database is not accessible.

PsfGetFormCount

Syntax

```
int FAR PASCAL PsfGetFormCount(HSESSION hSession, LPINT lpnCount);
```

Description

Counts the available forms.

PsfGetFormCount sets *lpnCount* to a pointer to the number of forms available in the forms database. The PeopleSoft system uses the count to allocate an array large enough to hold all the forms. It passes a pointer to this array to PsfGetFormList.

Parameters

Parameter	Description
hSession	The session handle assigned by PsfOpenSession when PeopleSoft connected to PSFORMS.DLL.
lpnCount	A pointer to an integer that will receive the count.

Returns

This operation returns the following values:

Value	Code	Meaning
PSF_OK	0	The function ran successfully; <i>lpnCount</i> is set to the number of forms.
PSF_NOMAIL	2	Mail system failure.
PSF_NOODB	5	The forms database is not accessible.

PsfGetFormList

Syntax

```
int FAR PASCAL PsfGetFormList(HSESSION hSession, LPPFSFORMLIST lpFormList);
```

Description

Lists the available forms.

PsfGetFormList provides a list of the available forms in the forms database. The structure's lpForms field points to the array of structures that PsfGetFieldList updates. Each item in the form name array has this structure:

```
typedef struct
{
    char name[PSF_FIELDNAMELEN + 1];
} PSFDEFNKEY;
```

Parameters

Parameter	Description
hSession	The session handle assigned by PsfOpenSession when PeopleSoft connected to PSFORMS.DLL.
lpFormList	A pointer to a structure to hold the list of form names. The form of the structure is: <pre>typedef struct { UINT nNumForms; // The number of forms LPPSFDEFNKEY lpForms; // Form name array } PSFFORMLIST;</pre>

Returns

This operation returns the following values:

Value	Code	Meaning
PSF_OK	0	The function ran successfully.
PSF_NOMAIL	2	Mail system failure.
PSF_NOODB	5	The forms database is not accessible.

PsfGetLastError

Syntax

```
int FAR PASCAL PsfGetLastError(HSESSION hSession, LPSTR lpszErrText,
UINT uBufSize, UINT uErrCode);
```

Description

PsfGetLastError gets the text of an error message based on an error code. The PeopleSoft system passes it the error code that it received and a pointer to a text buffer. PsfGetLastError copies the corresponding error text into the buffer, up to the size specified by *uBufSize*.

Parameters

Parameter	Description
hSession	The session handle assigned by PsfOpenSession when PeopleSoft connected to PSFORMS.DLL.
lpszErrText	A pointer to a buffer where the function places the error text.
uBufSize	The size of the buffer to which <i>lpszErrText</i> points.
uErrCode	The error code for which to retrieve the error text.

Returns

This operation returns the following values:

Value	Code	Meaning
PSF_OK	0	The function ran successfully.
PSF_NOERR	9	No message was found for the specified error code.

Send Operations

This section discusses the send operation.

PsfSendForm

Syntax

```
int FAR PASCAL PsfSendForm(HSESSION hSession, LPPSFDEFN lpFormData);
```

Description

Sends a form through the mail system with the specified field values.

PsfSendForm sends a form using the specified open session. The *lpFormData* structure has values in all fields, including the data values in the field array that it points to.

Parameters

Parameter	Description
hSession	The session handle assigned by PsfOpenSession when PeopleSoft connected to PSFORMS.DLL.
lpFormData	A pointer to a structure that contains form information. The form of the structure is: <pre>typedef struct { PSFDEFNKEYformName; // Name of the form WORD wNumFields; // Number of fields charcSep; // Separator character LPPSFIELDVAL lpFields; // Pointer to array } PSFFORMDEFN;</pre>
	Each item in the field array has this structure: <pre>typedef struct { char szFldName[PSF_FORMNAMELEN + 1]; // Field name WORD wFldSize; // Field size LPSTR lpszStrVal; // Value } PSFFIELDVAL;</pre>

Returns

This operation returns the following values:

Value	Code	Meaning
PSF_OK	0	The function ran successfully.
PSF_NOFIELD	1	A field is missing or is the wrong size.
PSF_NOMAIL	2	Mail system failure.
PSF_NOFORM	3	The form is not accessible.
PSF_NOODB	5	The forms database is not accessible.

CHAPTER 6

Using the Open Query ODBC Driver and API

This chapter provides an overview of PeopleSoft Open Query and discusses:

- Supported Open Database Connectivity (ODBC) version 2.5 functions.
- PeopleTools initialization procedures.
- Connection procedures.
- Information procedures.
- Catalog procedures (metadata).
- Using SQL execution procedures.
- SQL execution procedures.
- Execution models.
- Using retrieval procedures.
- Retrieval procedures.
- Status and error retrieval procedures.
- Transaction and connection termination procedures.
- ODBC compliance.
- ODBC to RDM data types.
- PeopleSoft Open Query ODBC API example.

Understanding PeopleSoft Open Query

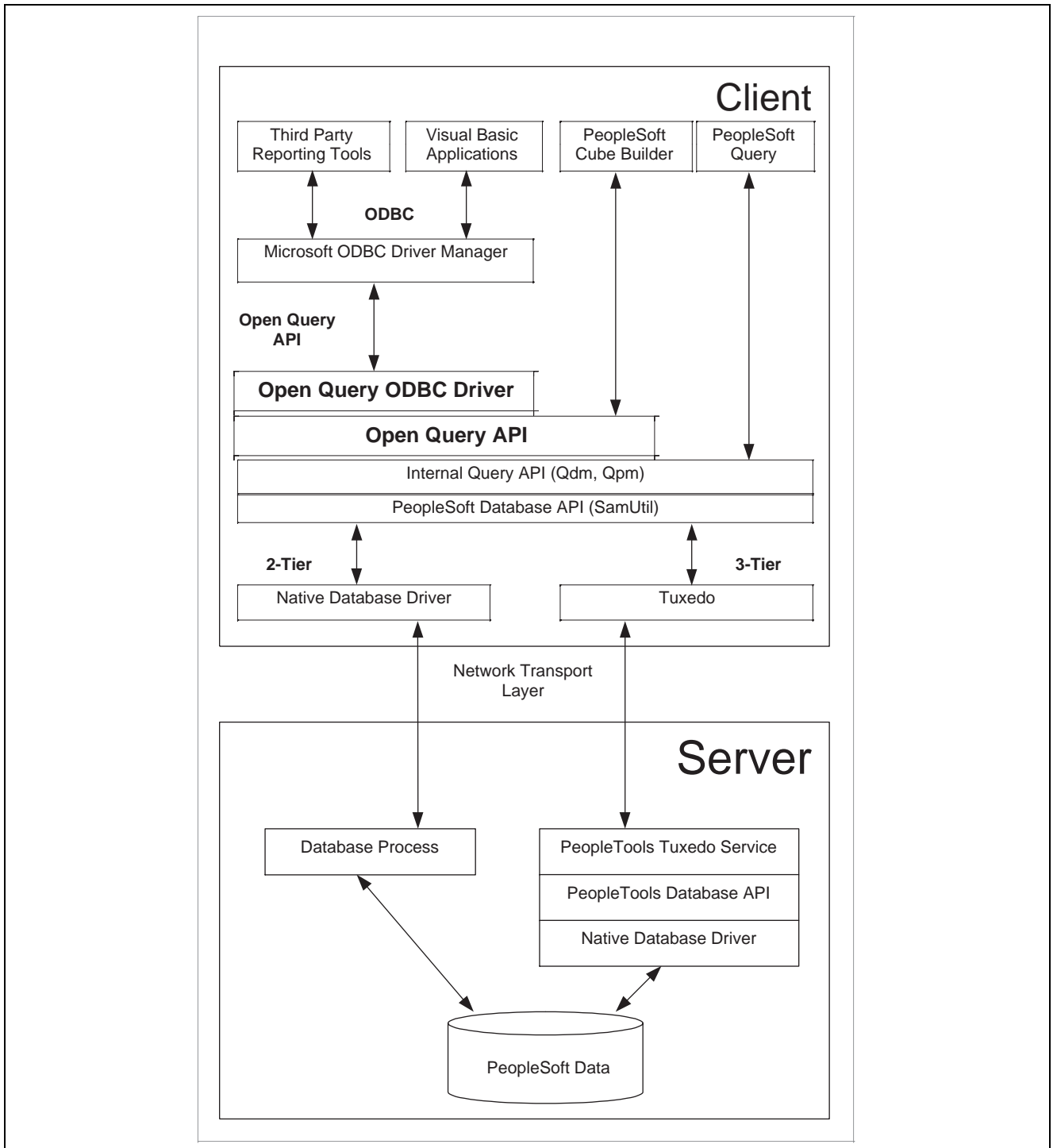
The PeopleSoft Open Query ODBC driver and API enable third-party reporting tools or applications to access PeopleSoft data in conformance with the PeopleSoft Query access architecture (the embedded SQL access intelligence provided by PeopleSoft Query). The PeopleSoft Query access architecture provides the following key features:

- Multiple levels of security.
 - Query authorization.
 - Operator security.
 - Operator profile.
 - Record level.
 - Access group.

- Row level.
- Security record.
- Standard query data access.
 - Access to all supported PeopleSoft databases.
 - Ability to run stored PeopleSoft queries.
 - Automatic use of table sets.
 - Effective-dated output.
 - Translate values.
 - International translations.

Architecture

This diagram illustrates the components involved in PeopleSoft Open Query architecture. The bold blocks represent PeopleSoft Open Query components.



PeopleSoft Open Query architecture

The following table describes PeopleSoft Open Query components:

Product	Audience	Purpose
Open Query ODBC	Third-party application vendors and implementation partners	Open, documented API providing access to PeopleSoft Query as a data source. Primary means of achieving programmatic interface to PeopleSoft data.
Open Query API	PeopleTools development.	PeopleSoft proprietary. Exposes Query definition and runtime functions for use within PeopleTools. API used by ODBC driver and online analytical processing (OLAP).

PeopleSoft Open Query ODBC Driver

The PeopleSoft Open Query ODBC driver is the external means to extract data from a PeopleSoft database.

Using ODBC as our external API has several advantages:

- Third-party tools do not need special information about PeopleSoft data.
- Most third-party reporting and query tools already support ODBC.
- Custom integration with PeopleTools is no longer required.
- Proprietary interface drivers (namely P2SPS.DLL) are eliminated.
- ODBC is supported by other application development tools, such as Visual Basic and PowerBuilder.
- Connectivity to PeopleSoft data is maintained by the PeopleSoft security architecture.
- Extension or modification of driver behavior is allowed by way of the ODBC standard.

The ODBC driver does not have any intrinsic knowledge of PeopleSoft data structures. Only the interface components exposed by way of PeopleSoft Open Query API are required to extract query data for a particular purpose.

This layer provides only the data modification and conversion code necessary to comply with the ODBC software development kit (SDK) standards. None of the PeopleTools structures are exposed at this level.

PeopleSoft Open Query API

The ODBC driver presents a stable interface to external applications. The PeopleSoft Open Query API is constantly under construction and therefore is not useful in this capacity. Because modifying behavior between releases presents a challenge when dealing with third-party applications, we created an intermediate layer between the ODBC driver and the underlying PeopleTools manager code. This wrapper code reduces the number of method calls to an acceptable level and allows the PeopleTools development team some leeway when new functionality is required or when the underlying code base is modified.

The PeopleSoft Open Query API enables the external driver (ODBC) and PeopleTools applications to focus solely on providing PeopleSoft data in the formats described by those products. It also abstracts the underlying connective architecture from the upper levels of the interface.

Because the architecture of this API is fluid, it is available only to PeopleTools development.

External Reporting Tools

The PeopleSoft Open Query ODBC driver enables third-party reporting tools or ODBC-enabled development applications to access a PeopleSoft database. The driver enforces user, table, and row-level security through internal PeopleTools APIs. A user can leverage PeopleSoft Query to easily create platform-independent queries that run against any supported PeopleSoft database platform.

Internet

The PeopleSoft Open Query API is a valuable link to all external data access mechanisms, including data over the internet.

Supported ODBC Version 2.5 Functions

This quick-reference summary lists the ODBC API calls supported by the ODBC driver. API calls that are not supported return SQL_ERROR. Each call is discussed in further detail in the following sections.

ODBC Call	Function
SQLAllocEnv	Allocates an environment handle for the ODBC connection.
SQLAllocConnect	Allocates a connection; returns a connection handle.
SQLAllocStmt	Allocates a statement handle for the specified connection.
SQLBindCol	Provides the buffer address for an answer column about to be fetched.
SQLBindParameter	Provides the value of a parameter (prompt variable) defined in the query.
SQLColAttributes	Returns result-column descriptor information for a result set.
SQLConnect	Connects to a PeopleSoft database.
SQLDescribeCol	Provides descriptors (data type and so on) for a result column.
SQLDescribeParam	Describes a parameter marker in a statement.
SQLDisconnect	Disconnects from the data source.

ODBC Call	Function
SQLDriverConnect	Connects to a PeopleSoft database, prompting the user for any login parameters not provided by the caller.
SQLError	Retrieves information about an error that occurred on a previous call.
SQLExecDirect	Prepares and executes a query. Note. Only the stored procedure syntax is supported in phase one.
SQLExecute	Executes a previously prepared query.
SQLFetch	Fetches a row of the answer set into the bound columns.
SQLFreeConnect	Closes the database connection and frees all resources that are associated with it.
SQLFreeStmt	Discards all resources that are associated with a previously prepared statement.
SQLGetData	Retrieves data for a specific column of the current fetched row. (Useful for long data, images, and so on.)
SQLGetFunctions	Tells applications which ODBC functions this driver supports.
SQLGetInfo	Retrieves information about the data source.
SQLGetRowCount	Returns the number of rows affected by the last execution.
SQLGetTypeInfo	Returns information about data types supported by the data source.
SQLNumParams	Returns the number of parameters in a statement.
SQLNumResultCols	Returns the number of result columns in the answer set of a prepared query.
SQLPrepare	Prepares a query for execution.

ODBC Call	Function
SQLProcedureColumns	Provides a list of queries and result columns available to the current operator and matching the specified qualifiers.
SQLProcedures	Retrieves a list of available stored procedures (queries).
SQLTransact	Commits or rolls back the current transaction.

The ODBC functions in the following table are supported calls with no underlying functionality. These functions exist to ensure compatibility with ODBC applications.

ODBC Call	Function
SQLColumns	Retrieves column information from the database.
SQLForeignKeys	Retrieves database information concerning foreign keys.
SQLGetConnectOption	Gets connection option information.
SQLGetCursorName	Gets the name of the cursor.
SQLGetStmtOption	Gets statement option information.
SQLMoreResults	Returns whether or not another result set is pending.
SQLPrimaryKeys	Retrieves database information on primary keys.
SQLSetConnectOption	Sets database connection options.
SQLSetCursorName	Sets the name of the cursor to be used with the statement.
SQLSetScrollOptions	Sets options to control cursor scrolling.
SQLSetStmtOption	Sets options for the statement.
SQLSpecialColumns	Retrieves information about optimal keys or auto-incremented columns.

ODBC Call	Function
SQLStatistics	Retrieves statistics on tables and indices from the database.
SQLTables	Retrieves a list of tables or views in the database.

PeopleTools Initialization Procedures

This section discusses the PeopleTools initialization procedures in alphabetical order.

SQLAllocEnv

Syntax

```
RETCODE SQLAllocEnv(phenv)
```

Description

SQLAllocEnv allocates memory for an environment handle and initializes the ODBC call level interface for use by an application.

Parameters

The parameters are:

Argument	Type	Use	Description
phenv	HENV FAR *	Output	Pointer to storage for the environment handle.

SQLAllocConnect

Syntax

```
RETCODE SQLAllocConnect(henv, phdbc)
```

Description

SQLAllocConnect allocates memory for a connection handle within the environment, identified by *henv*. This is called after SQLAllocEnv.

Parameters

The parameters are as follows:

Argument	Type	Use	Description
henv	HENV	Input	Environment handle.
phdbc	HDBC FAR *	Output	Pointer to storage for the connection handle.

SQLFreeEnv

Syntax

RETCODE `SQLFreeEnv` (*henv*)

Description

SQLFreeEnv releases an environment handle and frees all memory that is associated with the handle. This is called after SQLFreeConnect.

Parameters

The parameters are as follows:

Argument	Type	Use	Description
henv	HENV	Input	Environment handle.

Connection Procedures

This section discusses the PeopleTools connection procedures in alphabetical order.

ODBC uses an abstraction that maps a single name (called the data source name or DSN) to all the necessary underlying software components required to access the data. The data source name is chosen by an end user or a system administrator and should express the kind of data that the DSN represents. ODBC data source mapping information is maintained in the registry in Microsoft Windows NT.

In order to connect to a PeopleSoft data source, several pieces of information are needed. With the introduction in PeopleTools of multiple sign-on capabilities, users must be prompted for required log in information. By default, PeopleTools installs an ODBC data source with the name PeopleSoft PeopleTools. This DSN has no references to PeopleSoft connection information. It is in effect an empty data source. Using this data source forces the underlying PeopleSoft security mechanisms to present the user with a sign-on dialog box. The user completes this as for a normal PeopleSoft application.

As per the ODBC standard, the PeopleSoft driver enables the user to create a data source that provides the information necessary to log in. If the information matches a current session, the user is not prompted to log in again.

The connection environment is affected by the workstation settings for PeopleSoft Process Scheduler. The values for the DBBIN and TOOLBIN are searched for the necessary support files required to log in to a database. Hence these values need to be valid. In the case of a three-tier logon, the value for DBBIN should be set empty.

Note. ODBC supports three connection functions: `SQLConnect`, `SQLDriverConnect`, and `SQLBrowseConnect`. `SQLBrowseConnect` is not supported by the PeopleSoft Open Query ODBC driver.

SQLConnect

Syntax

```
RETCODE SQLConnect(hdbc, szDSN, cbDSN, szUID, cbUID, szAuthStr, cbAuthStr)
```

Description

`SQLConnect` loads a driver and establishes a connection to a data source. The connection handle references storage of all information about the connection, including status, transaction state, and error information.

This function assumes that a connection may be completed by supplying only a DSN, user, and password. It further assumes that the application will either prompt the end user for security information, the security information is hard-coded, or the security information can be obtained from a centralized security server such as like Kerberos.

Parameters

The parameters are:

Argument	Type	Use	Description
hdbc	HDBC	Input	Connection handle.
szDSN	UCHAR FAR*	Input	Data source name.
cbDSN	SWORD	Input	Length of <i>szDSN</i>
szUID	UCHAR FAR*	Input	User identifier.
cbUID	SWORD	Input	Length of <i>szUID</i> .
szAuthStr	UCHAR FAR*	Input	Authentication string (typically the password).
cbAuthStr	SWORD	Input	Length of <i>szAuthStr</i> .

SQLDriverConnect

Syntax

```
RETCODE SQLDriverConnect(hdbc, hwnd, szConnStrIn, cbConnStrIn, szConnStrOut,
cbConnStrOutMax, pcbConnStrOut, fDriverCompletion)
```

Description

SQLDriverConnect handles the entire connection process for an application, prompting the end user for any information to complete the connection. The programmer can specify as much or as little about the connection as necessary. The following example illustrates the simplest case, in which the application doesn't specify any information at all about the connection. It supplies the connection handle returned from SQLAllocConnect, a window handle, and option specification of SQL_DRIVER_COMPLETE, and zeros or NULLs for the rest of the arguments.

```
rc = SQLDriverConnect(hdbc, hwnd, NULL, 0, NULL, 0, 0, SQL_DRIVER_COMPLETE);
```

The following keywords are used and supported by the PeopleSoft ODBC driver:

- DSN: Data source name required by ODBC.
- APPNAME: Application server name used for three-tier logon only.
- DBTYPE: Database type of login can be any of the following values:
 - DB2: DB2 using Centura SQL network.
 - DB2400: DB2 on AS/400 using client access.
 - DB2ODBC: DB2 using the IBM ODBC driver.
 - DB2UNIX: DB2 UNIX driver.
 - ORACLE: Oracle using the OCI interface.
 - INFORMIX: Native Informix.
 - SYBASE: Native Sybase.
 - MICROSOFT: SQL Server using ODBC.
 - APPSRV: Used to indicate that the database name is actually an application server name.
- DBNAME: Name of the database or alias.
- DBQ: Used to combine values separated by a slash (/), such as APPNAME/DBTYPE/DBNAME. The APPNAME value and the following slash are dropped when not in three-tier.
- SERVER: Name of the database server, used by Sybase and Informix.
- UID: PeopleSoft user ID.
- PWD: Password associated with the PeopleSoft user.

The driver uses any information it retrieves from the ODBC.INI file or registry to augment the information passed to it in the connection string. If the information in the ODBC.INI file or registry duplicates information in the connection string, the driver uses the information in the connection string.

The existing PeopleSoft database connection dialog box prompts the user for any required connection information.

Parameters

The parameters are:

Argument	Type	Use	Description
hdbc	HDBC	Input	Connection handle.
hwnd	HWND	Input	Window handle. The application can pass the handle of the parent window, if applicable, or a null pointer if either the window handle is not applicable or if SQLDriverConnect does not present any dialog boxes.
szConnStrIn	UCHAR FAR*	Input	A full connection string, a partial connection string, or an empty string.
cbConnStrIn	WORD	Input	Length of szConnStrIn.
szConnStrOut	UCHAR FAR*	Output	Pointer to storage for the completed connection string. Upon successful connection to the target data source, this buffer contains the completed connection string. Applications should allocate at least 255 bytes for this buffer.
cbConnStrOutMax	WORD	Input	Maximum length of the szConnStrOut buffer.

Argument	Type	Use	Description
pcbConnStrOut	SWORD FAR*	Output	Pointer to the total number of bytes returned in szConnStrOut. If the number of bytes is greater than or equal to cbConnStrOutMax, the completed connection string in szConnStrOut is truncated to cbConnStrOutMax -1.
fDriverCompletion	UWORD	Input	Flag that indicates whether the Driver Manager or the driver must prompt for more connection information: SQL_DRIVER_PROMPT, SQL_DRIVER_COMPLETE, SQL_DRIVER_COMPLETE_REQUIRED, or SQL_DRIVER_NOPROMPT.

Information Procedures

ODBC defines these functions as a means for the application to get information about the ODBC driver and data source. Typically, the application calls these functions passing a value of the particular type of information of interest. These values are numerous and are defined in the Microsoft ODBC SDK reference manual.

SQLGetInfo

Syntax

```
RETCODE SQLGetInfo(hdbc, fInfoType, rgbInfoValue, cbInfoValueMax, pcbInfoValue)
```

Description

SQLGetInfo returns general information about the driver and data source that is associated with a connection handle.

Parameters

The parameters are:

Argument	Type	Use	Description
hdbc	HDBC	Input	Connection handle.
fInfoType	UWORD	Input	Type of information. <i>fInfoType</i> must be a value representing the type of interest.
rgbInfoValue	PTR	Output	Pointer to storage for the information. Depends on the <i>fInfoType</i> requested.
cbInfoValueMax	SWORD	Input	Maximum length of the <i>rgbInfoValue</i> buffer.
pcbInfoValue	SWORD FAR *	Output	The total number of bytes (excluding the null termination byte for character data) available to return in <i>rgbInfoValue</i> .

SQLFunctions

Syntax

```
RETCODE SQLGetFunctions(hdbc, fFunction, pfExists)
```

Description

SQLGetFunctions returns information about whether a driver supports a specific ODBC function.

Parameters

The parameters are:

Argument	Type	Use	Description
hdbc	HDBC	Input	Connection handle.

Argument	Type	Use	Description
fFunction	UWORD	Input	SQL_API_ALL_FUNCTIONS or a #define value that identifies the ODBC function of interest.
pfExists	UWORD FAR *	Output	If <i>fFunction</i> is SQL_API_ALL_FUNCTIONS, <i>pfExists</i> points to a UWORD array with 100 elements. The array is indexed by #define values that are used by <i>fFunction</i> to identify each ODBC function; some elements of the array are unused and reserved for future use. An element has a value of <i>True</i> if it identifies an ODBC function that is supported by the driver. It has a value of <i>False</i> if it identifies an ODBC function that is not supported by the driver or does not identify an ODBC function.

SQLGetTypeInfo

Syntax

```
RETCODE SQLGetTypeInfo(hstmt, fSqlType)
```

Description

SQLGetTypeInfo returns information about data types that are supported by the data source. The driver returns the information in the form of a SQL result set.

Parameters

The parameters are:

Argument	Type	Use	Description
hstmt	HSTMT	Input	Statement handle for the result set.
fSqlType	SWORD	Input	The SQL data type.

Catalog Procedures (Metadata)

ODBC listing procedures supply the client with catalog table information. The PeopleSoft ODBC driver supports listings of queries and columns using PeopleSoft metadata.

SQLProcedures

Syntax

```
RETCODE SQLProcedures(hstmt, szProcQualifier, cbProcQualifier, szProcOwner,
cbProcOwner, szProcName, cbProcName)
```

Description

SQLProcedures returns the list of procedure names that are stored in a specific data source. *Procedure* is a generic term used to describe executable objects or named entities that can be invoked using input and output parameters and that can return result sets similar to the results returned by SQL Select statements.

This function is typically used before statement execution to retrieve information about procedures available from the data source's catalog.

Parameters

The parameters are:

Argument	Type	Use	Description
hstmt	HSTMT	Input	Statement handle.
szProcQualifier	UCHAR FAR *	Input	Procedure qualifier.
cbProcQualifier	SWORD	Input	Length of <i>szProcQualifier</i> .
szProcOwner	UCHAR FAR *	Input	String search pattern for procedure owner names.
cbProcOwner	SWORD	Input	Length of <i>szProcOwner</i> .

Argument	Type	Use	Description
szProcName	UCHAR FAR *	Input	String search pattern for procedure names.
cbProcName	SWORD	Input	Length of <i>szProcName</i> .

SQLProcedures returns the results as a standard result set (when SQLFetch is called), ordered by PROCEDURE_QUALIFIER, PROCEDURE_OWNER, PROCEDURE_NAME, PROCEDURE_REMARKS, and PROCEDURE_TYPE.

The following table lists the columns that are in the PeopleSoft result set.

Column Name	Data Type	Description
PROCEDURE_QUALIFIER	SQL_CHAR(128)	''
PROCEDURE_OWNER	SQL_CHAR(128)	'QUERY'
PROCEDURE_NAME	SQL_CHAR(128)	Query name with punctuation and spaces converted to underscore
REMARKS	SQL_CHAR(256)	Description of the Query, currently unused
PROCEDURE_TYPE	SQL_INT	SQL_PT_PROCEDURE

SQLProcedureColumns

Syntax

```
RETCODE SQLProcedureColumns(hstmt, szProcQualifier, cbProcQualifier, szProcOwner,  
cbProcOwner, szProcName, cbProcName, szColumnName, cbColumnName)
```

Description

SQLProcedureColumns returns a list of input and output parameters, as well as the columns that make up the result set for the specified procedures. The driver returns the information as a result set on the specified statement handle.

This function is typically used before statement execution to retrieve information about procedure parameters and columns from the data source's catalog.

The PeopleSoft driver returns information for the first query that is requested only. It does not return results for multiple queries. The driver will use the new query API functions QpmDescribeParm and QpmDescribeCol. QpmDescribeParm walks the query definition that is stored in hstmt and for each prompt variable returns a SQLProcedureColumns result row of COLUMN_TYPE equal to SQL_PARM_INPUT. QpmDescribeCol walks the same query definition and for each result column returns a SQLProcedureColumns result row of COLUMN_TYPE equal to SQL_RESULT_COL. The szProcQualifier and szProcOwner criteria are ignored. The result set returned is for the current user ID. The result set columns for Procedure Qualifier and Procedure Remarks do not apply and are set to NULL with a one-byte column length. The Procedure Owner column is set to either the user ID or Public.

Parameters

The parameters are as follows:

Argument	Type	Use	Description
hstmt	HSTMT	Input	Statement handle.
szProcQualifier	UCHAR FAR*	Input	Procedure qualifier name.
cbProcQualifier	SWORD	Input	Length of szProcQualifier.
szProcOwner	UCHAR FAR*	Input	String search pattern for procedure owner names.
cbProcOwner	SWORD	Input	Length of szProcOwner.
szProcName	UCHAR FAR*	Input	String search pattern for procedure names.
cbProcName	SWORD	Input	Length of szProcName.
szColumnName	UCHAR FAR*	Input	String search pattern for column names.
cbColumnName	SWORD	Input	Length of szColumnName.

SQLProcedureColumns returns the results as a standard result set (when SQLFetch is called), ordered by PROCEDURE_QUALIFIER, PROCEDURE_OWNER, PROCEDURE_NAME, and COLUMN_TYPE.

The following table lists the columns in the result set.

Column Name	Data Type	Description
PROCEDURE_QUALIFIER	SQL_CHAR(128)	N/A
PROCEDURE_OWNER	SQL_CHAR(128)	N/A
PROCEDURE_NAME	SQL_CHAR(128)	Procedure identifier.
COLUMN_NAME	SQL_CHAR(128)	Procedure column identifier.
COLUMN_TYPE	SQL_INT	SQL_PARAM_INPUT or SQL_RESULT_COL.
DATA_TYPE	SQL_INT	SQL data type.
TYPE_NAME	SQL_CHAR(128)	Data type name of procedure column.
PRECISION	SQL_INT	Precision of the procedure column.
LENGTH	SQL_INT	Length in bytes of data transferred on a SQLGetData or SQLFetch operation.
SCALE	SQL_INT	Scale of procedure column.
RADIX	SQL_INT	N/A
NULLABLE	SQL_INT	Whether the procedure column accepts a NULL value.
REMARKS	SQL_CHAR(256)	A description of the procedure column.

Using SQL Execution Procedures

The minimum ODBC SQL conformance level requires the driver to support the following:

- Data Definition Language (DDL): Create Table and Drop Table.
- Data Manipulation Language (DML): simple Select, Insert, Update Searched, and Delete Searched.
- Expressions: simple (such as $A > B + C$).

- Data types: Char, Varchar, or Long Varchar.

The PeopleSoft Open Query ODBC driver does not support the minimum SQL conformance level even though it reports supporting extended syntax. PeopleSoft Open Query supports only the ODBC extended SQL grammar for stored procedures. The stored procedure syntax is:

```
{[? = ] call procedure_name [ (param, ...)]}
```

The stored procedure execution model supports the independent creation of a SQL statement. Independent creation is done through PeopleSoft Query. However, instead of a stored procedure, the result is a PeopleSoft Query object.

A *statement handle* references statement information, such as network information, SQLSTATE values and error messages, cursor names, number of result set columns, and status information for SQL statement processing. Before an application can execute a SQL statement, it must allocate a statement handle for the statement. To allocate a statement handle, an application declares a variable of type `hstmt` and passes its address to `SQLAllocStmt`.

See Also

[Chapter 6, “Using the Open Query ODBC Driver and API,” Execution Models, page 95](#)

SQL Execution Procedures

This section discusses the SQL execution procedures in alphabetical order.

SQLAllocStmt

Syntax

```
RETCODE SQLAllocStmt(hdbc, phstmt)
```

Description

`SQLAllocStmt` allocates memory for a statement handle and associates the statement handle with the connection that is specified by the connection handle.

If the application calls `SQLAllocStmt` with a pointer to a valid statement handle, the driver overwrites the statement handle without regard to its previous contents.

Parameters

The parameters are:

Argument	Type	Use	Description
hdbc	HDBC	Input	Connection handle.
phstmt	HSTMT FAR*	Output	Pointer to storage for the statement handle.

SQLExecDirect

Syntax

```
RETCODE SQLExecDirect(hstmt, szSqlStr, cbSqlStr)
```

Description

SQLExecDirect executes a preparable statement, using the current values of the parameter marker variables if any parameters exist in the statement. The application calls SQLExecDirect to send a SQL statement to the data source. The driver modifies the statement to use the form of SQL used by the data source, then submits it to the data source. In particular, the driver modifies the escape clauses that are used to define ODBC-specific SQL grammar extensions.

The application can include one or more parameter markers in the SQL statement. To include a parameter marker, the application embeds a question mark (?) into the SQL statement at the appropriate position. It is unnecessary to use any parameter markers, as PeopleSoft Query objects know the exact number of prompt values. The PeopleSoft driver prompts the user for input values if no values were supplied through the input or the SQLBindParameter function.

Only stored procedures (predefined queries) are supported.

In addition to the ODBC error conditions, the PeopleSoft driver returns an error condition if the following conditions are true:

- A valid PeopleSoft query name can not be found or loaded.
- Prompt values cannot be satisfied using a prompting page.

Parameters

The parameters are:

Argument	Type	Use	Description
hstmt	HSTMT	Input	Statement handle
szSqlStr	UCHAR FAR*	Input	SQL statement to be executed.
cbSqlStr	SDWORD	Input	Length of <i>szSqlStr</i> .

SQLPrepare

Syntax

```
RETCODE SQLPrepare(hstmt, szSqlStr, cbSqlStr)
```

Description

SQLPrepare prepares a SQL string for execution. The application calls SQLPrepare to send a SQL statement to the data source for preparation. The application can include one or more parameter markers in the SQL statement. To include a parameter marker, the application embeds a question mark (?) into the SQL string at the appropriate position. After a statement is prepared, the application uses *hstmt* to refer to the statement in later function calls. The prepared statement that is associated with the *hstmt* may be reexecuted by calling SQLExecute until the application frees the *hstmt* with a call to SQLFreeStmt with the SQL_DROP option or until the *hstmt* is used in a call to SQLPrepare, SQLExecDirect, or one of the catalog functions (SQLColumns, SQLTables, and so on). After the application prepares a statement, it can request information about the format of the result set.

Only stored procedures (predefined queries) are supported.

Parameters

The parameters are:

Argument	Type	Use	Description
<i>hstmt</i>	HSTMT	Input	Statement handle.
<i>szSqlStr</i>	UCHAR FAR*	Input	SQL statement to be executed.
<i>cbSqlStr</i>	SDWORD	Input	Length of <i>szSqlStr</i> .

SQLExecute

Syntax

```
RETCODE SQLExecute(hstmt)
```

Description

SQLExecute executes a prepared statement, using the current values of the parameter marker variables if any parameter markers exist in the statement. SQLExecute executes a statement prepared by SQLPrepare. After the application processes or discards the results from a call to SQLExecute, the application can call SQLExecute again with new parameter values.

To execute a Select statement more than once, the application must call SQLFreeStmt with the SQL_CLOSE parameter before reissuing the Select statement.

As in the SQLExecDirect function, the PeopleSoft ODBC driver prompts the user for input values if they have not been supplied.

Note. For applications to be flexible enough for using SQL for a particular purpose, you must provide the application a means to query the ODBC driver for information pertaining to required storage and data types. This is done using descriptive functions defined by the ODBC specification. ODBC-enabled applications use these functions to dynamically query the driver for information about the result set and the input and output values.

Parameters

The parameters are:

Argument	Type	Use	Description
hstmt	HSTMT	Input	Statement handle.

SQLColAttributes

Syntax

```
RETCODE SQLColAttributes(hstmt, icol, fDescType, rbgDesc, cbValueMax, pcbValue)
```

Description

SQLColAttributes returns descriptor information for a column in a result set; it cannot be used to return information about the bookmark column (column 0). Descriptor information is returned as a character string, a 32-bit descriptor-dependent value, or an integer value.

Parameters

The parameters are:

Argument	Type	Use	Description
hstmt	HSTMT	Input	Statement handle.
icol	UWORD	Input	Column number of result data.
fDescType	UWORD	Input	Valid descriptor type.
rbgDesc	PTR	Output	Pointer to storage for descriptor information.
cbValueMax	SWORD	Input	Maximum buffer size.
pcbValue	SWORD FAR*	Output	Output length of data in buffer.

SQLDescribeCol

Syntax

```
RETCODE SQLDescribeCol(hstmt, icol, szColName, cbColNameMax, pcbColName,  
pfSqlType, pcbColDef, pibScale, pfNullable)
```

Description

SQLDescribeCol returns the result descriptor?column name, type, precision, scale, and nullability?for one column in the result set. An application typically calls SQLDescribeCol after a call to SQLPrepare and before or after the associated call to SQLExecute. An application can also call SQLDescribeCol after a call to SQLExecDirect.

Parameters

The parameters are:

Argument	Type	Use	Description
<i>hstmt</i>	HSTMT	Input	Statement handle.
<i>icol</i>	UWORD	Input	Column number of result data.
<i>szColName</i>	UCHAR FAR*	Output	Pointer to storage for the column name.
<i>cbColNameMax</i>	SWORD	Input	Maximum length of the <i>szColName</i> buffer.
<i>pcbColName</i>	SWORD FAR*	Output	Total number of bytes available to return in <i>szColName</i> .
<i>pfSqlType</i>	SWORD FAR*	Output	The SQL data type of the column.
<i>pcbColDef</i>	UDWORD FAR*	Output	The precision of the column on the data source.
<i>pibScale</i>	SWORD FAR*	Output	The scale of the column on the data source.
<i>pfNullable</i>	SWORD FAR*	Output	Indicates whether the column allows NULL values.

SQLDescribeParam

Syntax

```
RETCODE SQLDescribeParam(hstmt, ipar, pfSqlType, pcbColDef, pibScale, pfNullable)
```

Description

SQLDescribeParam returns the description of a parameter marker that is associated with a prepared SQL statement. In terms of PeopleSoft Query objects, this is the description of the prompt values required to fulfill the query keys.

Parameters

The parameters are:

Argument	Type	Use	Description
<i>hstmt</i>	HSTMT	Input	Statement handle.
<i>ipar</i>	UWORD	Input	Marker number.
<i>pfSqlType</i>	SWORD FAR*	Output	Pointer to storage for the SQL type.
<i>pcbColDef</i>	SWORD FAR*	Output	Pointer to storage for precision of value.
<i>pibScale</i>	SWORD FAR*	Output	Pointer to storage for scale of value.
<i>pfNullable</i>	UDWORD FAR*	Output	Pointer to storage for nullable flag.

SQLGetRowCount

Syntax

```
RETCODE SQLRowCount(hstmt, pcrow)
```

Description

SQLRowCount returns the number of rows affected by an Update, Insert, or Delete statement or by a SQL_UPDATE, SQL_ADD, or SQL_DELETE operation in SQLSetPos. If SQLRowCount is called during a fetch cycle, the value returned is the number of rows returned to the application at the current position.

Parameters

The parameters are:

Argument	Type	Use	Description
hstmt	HSTMT	Input	Statement handle.
pcrow	SDWORD FAR *	Input	Pointer to storage of the row counter.

SQLNumParams

Syntax

```
RETCODE SQLNumParams(hstmt, pccol)
```

Description

SQLNumParams returns the number of parameters in a SQL statement.

Parameters

The parameters are:

Argument	Type	Use	Description
hstmt	HSTMT	Input	Statement handle
pccol	SWORD FAR*	Output	Number of parameters in the statement.

SQLNumResultCols

Syntax

```
RETCODE SQLNumResultCols(hstmt, pccol)
```

Description

SQLNumResultCols returns the number of columns in the result set. SQLNumResultCols can be called successfully only when the statement handle is in the prepared or executed state. An application typically would use the value returned in pccol in a loop and call SQLDescribeCol for each column in the result set.

An application retrieves an entire row of values using a technique called binding. Binding associates the data from the data source with variables in the application program. There are two directions of binding: input and output. Input data must always be bound. On output, when a result column is bound, the variable receives the value for that column each time a new row is fetched. The following example shows how this technique differs from SQLGetData:

```
/* for all columns in the current result set */
for (i = 0; i < columns; i++)
    SQLBindCol(hstmt, ..., &value[i], ...)
while (SQL_SUCCESS == (rc = SQLFetch(hstmt)))
/* value[ i .. n] contains data for current row */
```

Parameters

The parameters are:

Argument	Type	Use	Description
hstmt	HSTMT	Input	Statement handle
pccol	SWORD FAR*	Output	Number of columns in the result set.

SQLBindCol

Syntax

```
RETCODE SQLBindCol(hstmt, icol, fCType, rgbValue, cbValueMax, pcbValue)
```

Description

SQLBindCol assigns the storage and data type for a column in a result set.

Parameters

The parameters are:

Argument	Type	Use	Description
hstmt	HSTMT	Input	Statement handle.
icol	UWORD	Input	Column number of result data.
fCType	SWORD	Input	The C data type of the result data.
rgbValue	PTR	Both	A pointer to storage for the result column.
cbValueMax	SDWORD	Input	Maximum length of the rgbValue buffer.
pcbValue	SDWORD FAR*	Both	A pointer to a buffer for the SQL_NULL_DATA or the number of bytes available to return in rgbValue prior to calling SQLFetch.

SQLBindParameter

Syntax

```
RETCODE SQLBindParameter(hstmt, ipar, fParamType, fCType, fSqlType, cbColDef,
ibScale, rbgValue, cbValueMax, pcbValue)
```

Description

An application calls SQLBindParameter to bind each parameter marker in a SQL statement. Bindings remain in effect until the application calls SQLBindParameter again or until the application calls SQLFreeStmt with the SQL_DROP or SQL_RESET_PARAMS option.

An application can use SQLBindParameter to supply the prompt values for a PeopleSoft query. SQLBindParameter calls the new function, ODBCBindParm. ODBCBindParm converts the ODBC C data type, fCType, to the ODBC SQL data type, fSqlType. It then maps the ODBC SQL data type to a supported PeopleSoft RDM data type and calls the appropriate internal bind function.

An ODBC driver is required to support conversions from all ODBC C data types to the ODBC SQL data types that they support.

Parameters

The parameters:

Argument	Type	Use	Description
hstmt	HSTMT	Input	Statement handle
ipar	UWORD	Input	Parameter number, ordered sequentially left to right, starting at 1.
fParamType	SWORD	Input	The type of the parameter.
fCType	SWORD	Input	The C data type of the parameter.
fSqlType	SWORD	Input	The SQL data type of the parameter.
cbColDef	UDWORD	Input	The precision of the column or expression of the corresponding parameter marker.
ibScale	SWORD	Input	The scale of the column or expression of the corresponding parameter marker

Argument	Type	Use	Description
rgbValue	PTR	Both	A pointer to a buffer for the parameter's data.
cbValueMax	SDWORD	Input	Maximum length of the rgbValue buffer.
pcbValue	SDWORD FAR*	Both	A pointer to a buffer for the parameter's length.

An application may also supply prompt values as literal strings embedded in the SQL statement string. For example:

```
SQLExecDirect(hstmt, "{call query.myquery(8001, NEWGN)}", SQL_NTS);
```

If prompt values are not provided, PeopleSoft Query prompts the user for each required value at the time of statement execution.

Execution Models

ODBC supports three execution models. Each accomplishes the same tasks, but differs from the others with regard to when and where (on the client or on the server) each step is performed.

ExecDirect

In this model, the SQL statement is specified, sent to the server, and executed all in one step. This model is best suited for SQL statements for a particular purpose or SQL statements that are executed only once. Parameters can be used, but they act merely as placeholders that the driver replaces with the parameter values before it sends the SQL statement to the server.

The DBMS discards the optimization information that is used to execute the SQL statement after execution is complete. If the same statement is specified again with `SQLExecDirect`, the entire process of parsing and optimizing happens again.

Prepare/Execute

In this model, the SQL statement is prepared (sent to the server, parsed, and optimized) first and executed later. When the statement is executed, what flows to the server is not the SQL statement itself, but a way of referencing the statement so that the access plan can be executed immediately. Parameters are often used in these SQL statements, so the only items that flow to the server are the reference to the access plan and the parameter values, not the entire SQL statement.

The Prepare/Execute model should be used when repeated execution of the same SQL statement is needed and when the SQL statement must be constructed dynamically during runtime. To use this model, the application calls `SQLPrepare` and then (presumably in a loop) calls `SQLExecute`.

Stored Procedures

The stored procedure model is like the Prepare/Execute model except that with stored procedures, the preparation step can be done independently from the application and the stored procedure persists beyond the runtime of the application. To use stored procedures in ODBC, the application calls `SQLExecDirect` but uses the SQL statement to specify the stored procedure name, as illustrated in the following example:

```
SQLExecDirect(hstmt, "{call query.proc1(?,?,?)}", SQL_NTS);
```

Using Retrieval Procedures

For row-returning statements, such as Select statements or stored procedures, ODBC provides three ways to retrieve data. Using a single function call, an application can retrieve a single value, an entire row of values, or multiple rows of values. The PeopleSoft driver supports only the first two methods: single value and entire row.

One way that an application can retrieve data is by using a function call (`SQLGetData`) for every column in every row. The application supplies function arguments that specify the column number and a variable in which to receive the data. After the function call has been successfully executed, the value for the given column is returned in the variable. The application uses two loops to retrieve an entire result set, as in this example:

```
/* For all rows */
while (SQL_SUCCESS == (rc = SQLFetch(hstmt)))
    /* for all columns in current results set */
    for (colnum = 1; colnum <= columns; colnum++)
        SQLGetData(hstmt, colnum, ..., &value, ..)
```

`SQLGetData` is also used for the piecemeal retrieval of large text and binary data (such as images). It is often difficult or impossible for an application to allocate a single piece of memory big enough to hold a large data object, such as a 50-page document or a high-density bitmap.

Retrieval Procedures

This section discusses the retrieval procedures in alphabetical order.

SQLFetch

Syntax

```
RETCODE SQLFetch(hstmt)
```

Description

`SQLFetch` fetches a row of data from a result set. The driver returns data for all columns that were bound to storage locations with `SQLBindCol`. `SQLFetch` positions the cursor on the next row of the result set. When the cursor is positioned to the last row of the result set, `SQLFetch` returns `SQL_NO_DATA_FOUND`.

If the application called `SQLBindCol` to bind columns, `SQLFetch` stores data in the locations specified by the calls to `SQLBindCol`. If the application does not call `SQLBindCol` to bind any columns, `SQLFetch` doesn't return any data; it just moves the cursor to the next row. An application can call `SQLGetData` to retrieve data not previously bound to a storage location.

Parameters

The parameters are:

Argument	Type	Use	Description
hstmt	HSTMT	Input	Statement handle

SQLGetData

Syntax

```
RETCODE SQLGetData(hstmt, icol, fCType, rgbValue, cbValueMax, pcbValue)
```

Description

SQLGetData returns result data for a single unbound column in the current row. The application must call SQLFetch to position the cursor on a row of data before it calls SQLGetData. You can use SQLBindCol for some columns and use SQLGetData for others within the same row. This function can be used to retrieve character or binary data values in parts from a column with a character, binary, or data-source-specific data type (for example, data from SQL_LONGVARIABLE or SQL_LONGVARCHAR columns).

Parameters

The parameters are:

Argument	Type	Use	Description
hstmt	HSTMT	Input	Statement handle.
icol	UWORD	Input	Column number of result data.
fCType	SWORD	Input	The C data type of the result data.
rgbValue	PTR	Both	A pointer to storage for the result column.
cbValueMax	SDWORD	Input	Maximum length of the <i>rgbValue</i> buffer.
pcbValue	SDWORD FAR*	Both	A pointer to a buffer for the SQL_NULL_DATA or the number of bytes available to return in <i>rgbValue</i> prior to calling SQLFetch.

Status and Error Retrieval Procedures

When any ODBC call fails, the driver retains error information until the next ODBC call. The error state and error text is retrieved from the driver with the `SQLError` function.

SQLError

Syntax

```
RETCODE SQLError(henv, hdbc, hstmt, szSqlState, pfNativeError,
szErrorMsg, cbErrorMsgMax, pcbErrorMsg)
```

Description

`SQLError` returns error or status information. An application typically calls `SQLError` when a previous call to an ODBC function returns `SQL_ERROR` or `SQL_SUCCESS_WITH_INFO`. The application can, however, call `SQLError` after any ODBC function call.

Parameters

The parameters are:

Argument	Type	Use	Description
<code>henv</code>	<code>HENV</code>	Input	Environment handle or <code>SQL_NULL_HENV</code> .
<code>hdbc</code>	<code>HDBC</code>	Input	Connection handle or <code>SQL_NULL_HDBC</code> .
<code>hstmt</code>	<code>HSTMT</code>	Input	Statement handle or <code>SQL_NULL_HSTMT</code> .
<code>szSqlState</code>	<code>UCHAR FAR *</code>	Output	<code>SQLSTATE</code> as null terminated string.
<code>pfNativeError</code>	<code>SDWORD FAR *</code>	Output	Native error code (specific to the data source).
<code>szErrorMsg</code>	<code>UCHAR FAR *</code>	Output	Pointer to storage for the error message text.

Argument	Type	Use	Description
cbErrorMsgMax	SWORD	Input	Maximum length of the <i>szErrorMsg</i> buffer. This must be less than or equal to <code>SQL_MAX_MESSAGE_LENGTH - 1</code> .
pcbErrorMsg	SWORD FAR *	Output	Pointer to the total number of bytes (excluding the null termination byte) available to return in <i>szErrorMsg</i> . If the number of bytes available to return is greater than or equal to <i>cbErrorMsgMax</i> , the error message text in <i>szErrorMsg</i> is truncated to <i>cbErrorMsgMax - 1</i> bytes.

Transaction and Connection Termination Procedures

Each query object that runs in ODBC creates a transaction. To ensure that all memory that is associated with a transaction is freed and locks are released, an application should call `SQLTransact`.

SQLTransact

Syntax

```
RETCODE SQLTransact(henv, hdbc, fType)
```

Description

`SQLTransact` requests a commit or rollback operation for all active operations on all statement handles that are associated with a connection. `SQLTransact` can also request that a commit or rollback operation be performed for all connections that are associated with the environment handle. In the case of query objects, the transaction is automatically closed upon termination of the statement handle.

Parameters

The parameters are:

Argument	Type	Use	Description
henv	HENV	Input	Environment handle or SQL_NULL_HENV.
hdbc	HDBC	Input	Connection handle or SQL_NULL_HDBC.
fType	UWORD	Input	Flag for SQL_COMMIT or SQL_ROLLBACK.

SQLDisconnect

Syntax

```
RETCODE SQLDisconnect(hdbc)
```

Description

SQLDisconnect closes the connection that is associated with a specific connection handle.

If an application calls SQLDisconnect before it has freed all statement handles that are associated with the connection, the driver frees those statement handles after it successfully disconnects from the data source. However, if one or more of the statement handles that are associated with the connection are still executing asynchronously, SQLDisconnect will return SQL_ERROR.

Parameters

The parameters are:

Argument	Type	Use	Description
hdbc	HDBC	Input	Connection handle.

SQLFreeConnect

Syntax

```
RETCODE SQLFreeConnect(hdbc)
```

Description

SQLFreeConnect releases a connection handle and frees all memory that is associated with the handle. This is called after SQLDisconnect.

Parameters

The parameters are:

Argument	Type	Use	Description
hdbc	HDBC	Input	Connection handle.

SQLFreeEnv

Syntax

RETCODE **SQLFreeEnv** (*henv*)

Description

SQLFreeEnv frees the environment handle and releases all memory that is associated with the environment handle.

Parameters

The parameters :

Argument	Type	Use	Description
henv	HENV	Input	Environment handle.

ODBC Compliance

The ODBC API defines a set of core functions that correspond to the functions in the X/Open and SQL Access Group Call Level Interface specification. ODBC also defines two extended sets of functionality, Level 1 and Level 2.

For the specific ODBC API descriptions and implementation details, refer to the Microsoft Open Database Connectivity Software Development Kit, Version 2.50.

The following lists summarize the functionality that is included in each conformance level.

Core API functions allow the following:

- Allocation and releasing of environment, connection, and statement handles.
- Conversion to data sources. Use of multiple statements on a connection.
- Preparation and immediate execution of SQL statements.
- Assignment of storage for parameters in a SQL statement and result columns.
- Retrieval of data from a result set. Retrieval of information about a result set.
- Commit or rollback transactions.
- Retrieval of error information.

The Level 1 API allows all the core functions, plus the following:

- Connection to data sources with driver-specific dialog boxes.

- Set and inquire values of statement and connection options.
- Transmission of part or all of a parameter value (useful for long data).
- Retrieval of part or all of a result column value (useful for long data).
- Retrieval of catalog information (columns, special columns, statistics, and tables).
- Retrieval information about driver and data source capabilities, such as supported data types, scalar functions, and ODBC functions.
- The Level 2 API allows all the core and Level 1 functions, plus the following:
 - Ability to browse connection information and list available data sources.
 - Transmission of arrays of parameter values. Retrieval of arrays of result columns values.
 - Retrieval of the number of parameters and description of individual parameters.
 - Scrollable cursor.
 - Retrieval of the native form of a SQL statement.
 - Retrieval of catalog information (privileges, keys, and procedures).
 - Translation DLL calls.

To claim that it conforms to a given API or SQL conformance level, a driver must support all the functionality in that conformance level, regardless of whether that functionality is supported by the DBMS that is associated with the driver. However, conformance levels do not restrict drivers to the functionality in the levels to which they conform. Drivers support as much functionality as they can; applications can determine the functionality that is supported by a driver by calling `SQLGetInfo`, `SQLGetFunctions`, and `SQLGetTypeInfo`.

ODBC to RDM Data Types

The following table shows the mapping from ODBC C data types to ODBC SQL and PeopleSoft RDM data types.

RDM Type	fSqlType	Type
RDM_CHAR	SQL_CHAR	Unsigned char FAR*
RDM_LONG_CHAR	SQL_VARCHAR	Unsigned char FAR*
RDM_NUMBER, RDM_SIGNED_NUMBER	SQL_NUMERIC	Unsigned char FAR*
RDM_DATE	SQL_DATE	Struct tag DATE_STRUCT { UWORD year; UWORD month; UWORD day; }

RDM Type	fSqlType	Type
RDM_TIME	SQL_TIME	Struct TIME_STRUCT { UWORD hour; UWORD minute; UWORD second; }
RDM_DATETIME	SQL_TIMESTAMP	Struct TIMESTAMP_STRUCT { SWORD year; UWORD month; UWORD day; UWORD hour; UWORD minute; UWORD second; UWORD fraction; }
RDM_IMAGE	SQL_LONGVARBINARY	Unsigned char FAR *

PeopleSoft Open Query ODBC API Example

The following example shows the ODBC API calls needed to execute a PeopleSoft query using the PeopleSoft Open Query ODBC driver. The following query requires two bind variables: business unit and department ID. The sample query returns an answer set of three columns: employee ID, name, and monthly rate.

```

/*****
* Function:      OpenQuerySample
*
* Description:  Sample program illustrating the usage of PeopleSoft
*              Open Query ODBC API.
*              Sample code uses basic PeopleSoft Query ODBC interface
*              functions. Most error checking is excluded to make
*              code easier to follow; in a typical application,
*              every return code would be checked.
*
* Returns:      TRUE if successful
*****/

BOOL WINAPI      OpenQuerySample(HWND hWnd)
{
HENV             hEnv;           // Environment handle for application
HDBC             hDbc;           // Connection handle
HSTMT            hStmt;         // Statement handle
RETCODE          retcode;       // Return code
char             szConnectString[] =
"DSN=PeopleSoft PeopleTools;DBTYPE=ORACLE;DBNAME=PTDMO7;UID=PTDMO;PWD=PTDMO;";
char             szConnStringOut[256]; // completed connection string
SWORD           nConnStringLength; // length of completed connect string

```

```

// Allocate environment, database connection
retcode = SQLAllocEnv(&hEnv);
if ((retcode = SQLAllocConnect(hEnv, &hDbc)) != SQL_SUCCESS)
    // error--this would normally abort program with message
    return(FALSE);

// Connect to the database
retcode = SQLDriverConnect(hDbc, hWnd, szConnectionString,
    strlen(szConnectionString), szConnStringOut, sizeof(szConnStringOut),
    &nConnStringLength, SQL_DRIVER_COMPLETE);

retcode = SQLAllocStmt(hDbc, &hStmt);

ProcessQuery(hStmt);

// Close the connection, release resources
retcode = SQLFreeStmt(hStmt);
retcode = SQLDisconnect(hDbc);
retcode = SQLFreeConnect(hDbc);
retcode = SQLFreeEnv(hEnv);

return(TRUE);
}

/*****
* Function:      ProcessQuery                                     *
*                                                       *
* Description:   Run a query and retrieve answer set.          *
*                                                       *
* Returns:      TRUE if successful, else FALSE                 *
*****/

BOOL          ProcessQuery(HSTMT hStmt)
{
    RETCODE    retcode;          // Return code
    char       szSelect[] = "{call query.myquery(?,?)";

    // binding of input variables must occur before statement execution
    for (i = 0; i < 2; i++)
        retcode = SQLBindCol(hStmt, i, datatype, &value, sizeof(value), &valueLen);

    retcode = SQLExecDirect(hStmt, szSelect, strlen(szSelect));

    while (retcode = SQLFetch(hStmt) == SQL_SUCCESS)
        // process data for a fetched row...

    return(retcode == SQL_NO_DATA_FOUND);
}

```

Glossary of PeopleSoft Terms

absence entitlement	This element defines rules for granting paid time off for valid absences, such as sick time, vacation, and maternity leave. An absence entitlement element defines the entitlement amount, frequency, and entitlement period.
absence take	This element defines the conditions that must be met before a payee is entitled to take paid time off.
academic career	In PeopleSoft Enterprise Campus Solutions, all course work that a student undertakes at an academic institution and that is grouped in a single student record. For example, a university that has an undergraduate school, a graduate school, and various professional schools might define several academic careers—an undergraduate career, a graduate career, and separate careers for each professional school (law school, medical school, dental school, and so on).
academic institution	In PeopleSoft Enterprise Campus Solutions, an entity (such as a university or college) that is independent of other similar entities and that has its own set of rules and business processes.
academic organization	In PeopleSoft Enterprise Campus Solutions, an entity that is part of the administrative structure within an academic institution. At the lowest level, an academic organization might be an academic department. At the highest level, an academic organization can represent a division.
academic plan	In PeopleSoft Enterprise Campus Solutions, an area of study—such as a major, minor, or specialization—that exists within an academic program or academic career.
academic program	In PeopleSoft Enterprise Campus Solutions, the entity to which a student applies and is admitted and from which the student graduates.
accounting class	In PeopleSoft Enterprise Performance Management, the accounting class defines how a resource is treated for generally accepted accounting practices. The Inventory class indicates whether a resource becomes part of a balance sheet account, such as inventory or fixed assets, while the Non-inventory class indicates that the resource is treated as an expense of the period during which it occurs.
accounting date	The accounting date indicates when a transaction is recognized, as opposed to the date the transaction actually occurred. The accounting date and transaction date can be the same. The accounting date determines the period in the general ledger to which the transaction is to be posted. You can only select an accounting date that falls within an open period in the ledger to which you are posting. The accounting date for an item is normally the invoice date.
accounting split	The accounting split method indicates how expenses are allocated or divided among one or more sets of accounting ChartFields.
accumulator	You use an accumulator to store cumulative values of defined items as they are processed. You can accumulate a single value over time or multiple values over time. For example, an accumulator could consist of all voluntary deductions, or all company deductions, enabling you to accumulate amounts. It allows total flexibility for time periods and values accumulated.
action reason	The reason an employee's job or employment information is updated. The action reason is entered in two parts: a personnel action, such as a promotion, termination, or change from one pay group to another—and a reason for that action. Action reasons are used by PeopleSoft Human Resources, PeopleSoft Benefits Administration,

PeopleSoft Stock Administration, and the COBRA Administration feature of the Base Benefits business process.

action template

In PeopleSoft Receivables, outlines a set of escalating actions that the system or user performs based on the period of time that a customer or item has been in an action plan for a specific condition.

activity

In PeopleSoft Enterprise Learning Management, an instance of a catalog item (sometimes called a class) that is available for enrollment. The activity defines such things as the costs that are associated with the offering, enrollment limits and deadlines, and waitlisting capacities.

In PeopleSoft Enterprise Performance Management, the work of an organization and the aggregation of actions that are used for activity-based costing.

In PeopleSoft Project Costing, the unit of work that provides a further breakdown of projects—usually into specific tasks.

In PeopleSoft Workflow, a specific transaction that you might need to perform in a business process. Because it consists of the steps that are used to perform a transaction, it is also known as a step map.

address usage

In PeopleSoft Enterprise Campus Solutions, a grouping of address types defining the order in which the address types are used. For example, you might define an address usage code to process addresses in the following order: billing address, dormitory address, home address, and then work address.

adjustment calendar

In PeopleSoft Enterprise Campus Solutions, the adjustment calendar controls how a particular charge is adjusted on a student's account when the student drops classes or withdraws from a term. The charge adjustment is based on how much time has elapsed from a predetermined date, and it is determined as a percentage of the original charge amount.

administrative function

In PeopleSoft Enterprise Campus Solutions, a particular functional area that processes checklists, communication, and comments. The administrative function identifies which variable data is added to a person's checklist or communication record when a specific checklist code, communication category, or comment is assigned to the student. This key data enables you to trace that checklist, communication, or comment back to a specific processing event in a functional area.

admit type

In PeopleSoft Enterprise Campus Solutions, a designation used to distinguish first-year applications from transfer applications.

agreement

In PeopleSoft eSettlements, provides a way to group and specify processing options, such as payment terms, pay from a bank, and notifications by a buyer and supplier location combination.

allocation rule

In PeopleSoft Enterprise Incentive Management, an expression within compensation plans that enables the system to assign transactions to nodes and participants. During transaction allocation, the allocation engine traverses the compensation structure from the current node to the root node, checking each node for plans that contain allocation rules.

alternate account

A feature in PeopleSoft General Ledger that enables you to create a statutory chart of accounts and enter statutory account transactions at the detail transaction level, as required for recording and reporting by some national governments.

analysis database

In PeopleSoft Enterprise Campus Solutions, database tables that store large amounts of student information that may not appear in standard report formats. The analysis database tables contain keys for all objects in a report that an application program can use to reference other student-record objects that are not contained in the printed report. For instance, the analysis database contains data on courses that are considered for satisfying a requirement but that are rejected. It also contains information on

	courses captured by global limits. An analysis database is used in PeopleSoft Enterprise Academic Advisement.
Application Messaging	PeopleSoft Application Messaging enables applications within the PeopleSoft Enterprise product family to communicate synchronously or asynchronously with other PeopleSoft and third-party applications. An application message defines the records and fields to be published or subscribed to.
AR specialist	Abbreviation for <i>receivables specialist</i> . In PeopleSoft Receivables, an individual in who tracks and resolves deductions and disputed items.
arbitration plan	In PeopleSoft Enterprise Pricer, defines how price rules are to be applied to the base price when the transaction is priced.
assessment rule	In PeopleSoft Receivables, a user-defined rule that the system uses to evaluate the condition of a customer's account or of individual items to determine whether to generate a follow-up action.
asset class	An asset group used for reporting purposes. It can be used in conjunction with the asset category to refine asset classification.
attribute/value pair	In PeopleSoft Directory Interface, relates the data that makes up an entry in the directory information tree.
audience	In PeopleSoft Enterprise Campus Solutions, a segment of the database that relates to an initiative, or a membership organization that is based on constituent attributes rather than a dues-paying structure. Examples of audiences include the Class of '65 and Undergraduate Arts & Sciences.
authentication server	A server that is set up to verify users of the system.
base time period	In PeopleSoft Business Planning, the lowest level time period in a calendar.
benchmark job	In PeopleSoft Workforce Analytics, a benchmark job is a job code for which there is corresponding salary survey data from published, third-party sources.
billing career	In PeopleSoft Enterprise Campus Solutions, the one career under which other careers are grouped for billing purposes if a student is active simultaneously in multiple careers.
bio bit or bio brief	In PeopleSoft Enterprise Campus Solutions, a report that summarizes information stored in the system about a particular constituent. You can generate standard or specialized reports.
book	In PeopleSoft Asset Management, used for storing financial and tax information, such as costs, depreciation attributes, and retirement information on assets.
branch	A tree node that rolls up to nodes above it in the hierarchy, as defined in PeopleSoft Tree Manager.
budgetary account only	An account used by the system only and not by users; this type of account does not accept transactions. You can only budget with this account. Formerly called "system-maintained account."
budget check	In commitment control, the processing of source transactions against control budget ledgers, to see if they pass, fail, or pass with a warning.
budget control	In commitment control, budget control ensures that commitments and expenditures don't exceed budgets. It enables you to track transactions against corresponding budgets and terminate a document's cycle if the defined budget conditions are not met. For example, you can prevent a purchase order from being dispatched to a vendor if there are insufficient funds in the related budget to support it.

budget period	The interval of time (such as 12 months or 4 quarters) into which a period is divided for budgetary and reporting purposes. The ChartField allows maximum flexibility to define operational accounting time periods without restriction to only one calendar.
business activity	The name of a subset of a detailed business process. This might be a specific transaction, task, or action that you perform in a business process.
business event	In PeopleSoft Receivables, defines the processing characteristics for the Receivable Update process for a draft activity. In PeopleSoft Sales Incentive Management, an original business transaction or activity that may justify the creation of a PeopleSoft Enterprise Incentive Management event (a sale, for example).
business process	A standard set of 17 business processes are defined and maintained by the PeopleSoft product families and are supported by Business Process Engineering group at PeopleSoft. An example of a business process is Order Fulfillment, which is a business process that manages sales orders and contracts, inventory, billing, and so forth. See also <i>detailed business process</i> .
business task	The name of the specific function depicted in one of the business processes.
business unit	A corporation or a subset of a corporation that is independent with regard to one or more operational or accounting functions.
buyer	In PeopleSoft eSettlements, an organization (or business unit, as opposed to an individual) that transacts with suppliers (vendors) within the system. A buyer creates payments for purchases that are made in the system.
campus	In PeopleSoft Enterprise Campus Solutions, an entity that is usually associated with a distinct physical administrative unit, that belongs to a single academic institution, that uses a unique course catalog, and that produces a common transcript for students within the same academic career.
catalog item	In PeopleSoft Enterprise Learning Management, a specific topic that a learner can study and have tracked. For example, "Introduction to Microsoft Word." A catalog item contains general information about the topic and includes a course code, description, categorization, keywords, and delivery methods. A catalog item can have one or more learning activities.
catalog map	In PeopleSoft Catalog Management, translates values from the catalog source data to the format of the company's catalog.
catalog partner	In PeopleSoft Catalog Management, shares responsibility with the enterprise catalog manager for maintaining catalog content.
categorization	Associates partner offerings with catalog offerings and groups them into enterprise catalog categories.
category	In PeopleSoft Enterprise Campus Solutions, a broad grouping to which specific comments or communications (contexts) are assigned. Category codes are also linked to 3C access groups so that you can assign data-entry or view-only privileges across functions.
channel	In PeopleSoft MultiChannel Framework, email, chat, voice (computer telephone integration [CTI]), or a generic event.
ChartField	A field that stores a chart of accounts, resources, and so on, depending on the PeopleSoft application. ChartField values represent individual account numbers, department codes, and so forth.
ChartField balancing	You can require specific ChartFields to match up (balance) on the debit and the credit side of a transaction.

ChartField combination edit	The process of editing journal lines for valid ChartField combinations based on user-defined rules.
ChartKey	One or more fields that uniquely identify each row in a table. Some tables contain only one field as the key, while others require a combination.
checkbook	In PeopleSoft Promotions Management, enables you to view financial data (such as planned, incurred, and actual amounts) that is related to funds and trade promotions.
checklist code	In PeopleSoft Enterprise Campus Solutions, a code that represents a list of planned or completed action items that can be assigned to a staff member, volunteer, or unit. Checklists enable you to view all action assignments on one page.
class	In PeopleSoft Enterprise Campus Solutions, a specific offering of a course component within an academic term. See also <i>course</i> .
Class ChartField	A ChartField value that identifies a unique appropriation budget key when you combine it with a fund, department ID, and program code, as well as a budget period. Formerly called <i>sub-classification</i> .
clearance	In PeopleSoft Enterprise Campus Solutions, the period of time during which a constituent in PeopleSoft Contributor Relations is approved for involvement in an initiative or an action. Clearances are used to prevent development officers from making multiple requests to a constituent during the same time period.
clone	In PeopleCode, to make a unique copy. In contrast, to <i>copy</i> may mean making a new reference to an object, so if the underlying object is changed, both the copy and the original change.
cohort	In PeopleSoft Enterprise Campus Solutions, the highest level of the three-level classification structure that you define for enrollment management. You can define a cohort level, link it to other levels, and set enrollment target numbers for it. See also <i>population</i> and <i>division</i> .
collection	To make a set of documents available for searching in Verity, you must first create at least one collection. A collection is set of directories and files that allow search application users to use the Verity search engine to quickly find and display source documents that match search criteria. A collection is a set of statistics and pointers to the source documents, stored in a proprietary format on a file server. Because a collection can only store information for a single location, PeopleSoft maintains a set of collections (one per language code) for each search index object.
collection rule	In PeopleSoft Receivables, a user-defined rule that defines actions to take for a customer based on both the amount and the number of days past due for outstanding balances.
comm key	See <i>communication key</i> .
communication key	In PeopleSoft Enterprise Campus Solutions, a single code for entering a combination of communication category, communication context, communication method, communication direction, and standard letter code. Communication keys (also called <i>comm keys</i> or <i>speed keys</i>) can be created for background processes as well as for specific users.
compensation object	In PeopleSoft Enterprise Incentive Management, a node within a compensation structure. Compensation objects are the building blocks that make up a compensation structure's hierarchical representation.

compensation structure	In PeopleSoft Enterprise Incentive Management, a hierarchical relationship of compensation objects that represents the compensation-related relationship between the objects.
component interface	A component interface is a set of application programming interfaces (APIs) that you can use to access and modify PeopleSoft database information using a program instead of the PeopleSoft client.
condition	In PeopleSoft Receivables, occurs when there is a change of status for a customer's account, such as reaching a credit limit or exceeding a user-defined balance due.
configuration parameter catalog	Used to configure an external system with PeopleSoft. For example, a configuration parameter catalog might set up configuration and communication parameters for an external server.
configuration plan	In PeopleSoft Enterprise Incentive Management, configuration plans hold allocation information for common variables (not incentive rules) and are attached to a node without a participant. Configuration plans are not processed by transactions.
constituents	In PeopleSoft Enterprise Campus Solutions, friends, alumni, organizations, foundations, or other entities affiliated with the institution, and about which the institution maintains information. The constituent types delivered with PeopleSoft Enterprise Contributor Relations Solutions are based on those defined by the Council for the Advancement and Support of Education (CASE).
content reference	Content references are pointers to content registered in the portal registry. These are typically either URLs or iScripts. Content references fall into three categories: target content, templates, and template pagelets.
context	<p>In PeopleCode, determines which buffer fields can be contextually referenced and which is the current row of data on each scroll level when a PeopleCode program is running.</p> <p>In PeopleSoft Enterprise Campus Solutions, a specific instance of a comment or communication. One or more contexts are assigned to a category, which you link to 3C access groups so that you can assign data-entry or view-only privileges across functions.</p> <p>In PeopleSoft Enterprise Incentive Management, a mechanism that is used to determine the scope of a processing run. PeopleSoft Enterprise Incentive Management uses three types of context: plan, period, and run-level.</p>
control table	Stores information that controls the processing of an application. This type of processing might be consistent throughout an organization, or it might be used only by portions of the organization for more limited sharing of data.
cost profile	A combination of a receipt cost method, a cost flow, and a deplete cost method. A profile is associated with a cost book and determines how items in that book are valued, as well as how the material movement of the item is valued for the book.
cost row	A cost transaction and amount for a set of ChartFields.
course	<p>In PeopleSoft Enterprise Campus Solutions, a course that is offered by a school and that is typically described in a course catalog. A course has a standard syllabus and credit level; however, these may be modified at the class level. Courses can contain multiple components such as lecture, discussion, and lab.</p> <p>See also <i>class</i>.</p>
course share set	In PeopleSoft Enterprise Campus Solutions, a tag that defines a set of requirement groups that can share courses. Course share sets are used in PeopleSoft Enterprise Academic Advisement.

current learning	In PeopleSoft Enterprise Learning Management, a self-service repository for all of a learner's in-progress learning activities and programs.
data acquisition	In PeopleSoft Enterprise Incentive Management, the process during which raw business transactions are acquired from external source systems and fed into the operational data store (ODS).
data cube	In PeopleSoft Analytic Calculation Engine, a data cube is a container for one kind of data (such as Sales data) and works with in tandem with one or more dimensions. Dimensions and data cubes in PeopleSoft Analytic Calculation Engine are unrelated to dimensions and online analytical processing (OLAP) cubes in PeopleSoft Cube Manager.
data elements	Data elements, at their simplest level, define a subset of data and the rules by which to group them. For Workforce Analytics, data elements are rules that tell the system what measures to retrieve about your workforce groups.
dataset	A data grouping that enables role-based filtering and distribution of data. You can limit the range and quantity of data that is displayed for a user by associating dataset rules with user roles. The result of dataset rules is a set of data that is appropriate for the user's roles.
delivery method	In PeopleSoft Enterprise Learning Management, identifies the primary type of delivery method in which a particular learning activity is offered. Also provides default values for the learning activity, such as cost and language. This is primarily used to help learners search the catalog for the type of delivery from which they learn best. Because PeopleSoft Enterprise Learning Management is a blended learning system, it does not enforce the delivery method. In PeopleSoft Supply Chain Management, identifies the method by which goods are shipped to their destinations (such as truck, air, rail, and so on). The delivery method is specified when creating shipment schedules.
delivery method type	In PeopleSoft Enterprise Learning Management, identifies how learning activities can be delivered—for example, through online learning, classroom instruction, seminars, books, and so forth—in an organization. The type determines whether the delivery method includes scheduled components.
detailed business process	A subset of the business process. For example, the detailed business process named Determine Cash Position is a subset of the business process called Cash Management.
dimension	In PeopleSoft Analytic Calculation Engine, a dimension contains a list of one kind of data that can span various contexts, and it is a basic component of an analytic model. Within the analytic model, a dimension is attached to one or more data cubes. In PeopleSoft Cube Manager, a dimension is the most basic component of an OLAP cube and specifies the PeopleSoft metadata to be used to create the dimension's rollup structure. Dimensions and data cubes in PeopleSoft Analytic Calculation Engine are unrelated to dimensions and OLAP cubes in PeopleSoft Cube Manager.
directory information tree	In PeopleSoft Directory Interface, the representation of a directory's hierarchical structure.
division	In PeopleSoft Enterprise Campus Solutions, the lowest level of the three-level classification structure that you define in PeopleSoft Enterprise Recruiting and Admissions for enrollment management. You can define a division level, link it to other levels, and set enrollment target numbers for it. See also <i>population</i> and <i>cohort</i> .

document sequencing	A flexible method that sequentially numbers the financial transactions (for example, bills, purchase orders, invoices, and payments) in the system for statutory reporting and for tracking commercial transaction activity.
dynamic detail tree	A tree that takes its detail values—dynamic details—directly from a table in the database, rather than from a range of values that are entered by the user.
edit table	A table in the database that has its own record definition, such as the Department table. As fields are entered into a PeopleSoft application, they can be validated against an edit table to ensure data integrity throughout the system.
effective date	A method of dating information in PeopleSoft applications. You can predate information to add historical data to your system, or postdate information in order to enter it before it actually goes into effect. By using effective dates, you don't delete values; you enter a new value with a current effective date.
EIM ledger	Abbreviation for <i>Enterprise Incentive Management ledger</i> . In PeopleSoft Enterprise Incentive Management, an object to handle incremental result gathering within the scope of a participant. The ledger captures a result set with all of the appropriate traces to the data origin and to the processing steps of which it is a result.
elimination set	In PeopleSoft General Ledger, a related group of intercompany accounts that is processed during consolidations.
entry event	In PeopleSoft General Ledger, Receivables, Payables, Purchasing, and Billing, a business process that generates multiple debits and credits resulting from single transactions to produce standard, supplemental accounting entries.
equitization	In PeopleSoft General Ledger, a business 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.
equity item limit	In PeopleSoft Enterprise Campus Solutions, the amounts of funds set by the institution to be awarded with discretionary or gift funds. The limit could be reduced by amounts equal to such things as expected family contribution (EFC) or parent contribution. Students are packaged by Equity Item Type Groups and Related Equity Item Types. This limit can be used to assure that similar student populations are packaged equally.
event	A predefined point either in the Component Processor flow or in the program flow. As each point is encountered, the event activates each component, triggering any PeopleCode program that is associated with that component and that event. Examples of events are FieldChange, SavePreChange, and RowDelete. In PeopleSoft Human Resources, also refers to an incident that affects benefits eligibility.
event propagation process	In PeopleSoft Sales Incentive Management, a process that determines, through logic, the propagation of an original PeopleSoft Enterprise Incentive Management event and creates a derivative (duplicate) of the original event to be processed by other objects. Sales Incentive Management uses this mechanism to implement splits, roll-ups, and so on. Event propagation determines who receives the credit.
exception	In PeopleSoft Receivables, an item that either is a deduction or is in dispute.
exclusive pricing	In PeopleSoft Order Management, a type of arbitration plan that is associated with a price rule. Exclusive pricing is used to price sales order transactions.
fact	In PeopleSoft applications, facts are numeric data values from fields from a source database as well as an analytic application. A fact can be anything you want to measure your business by, for example, revenue, actual, budget data, or sales numbers. A fact is stored on a fact table.

financial aid term	In PeopleSoft Enterprise Campus Solutions, a combination of a period of time that the school determines as an instructional accounting period and an academic career. It is created and defined during the setup process. Only terms eligible for financial aid are set up for each financial aid career.
forecast item	A logical entity with a unique set of descriptive demand and forecast data that is used as the basis to forecast demand. You create forecast items for a wide range of uses, but they ultimately represent things that you buy, sell, or use in your organization and for which you require a predictable usage.
fund	In PeopleSoft Promotions Management, a budget that can be used to fund promotional activity. There are four funding methods: top down, fixed accrual, rolling accrual, and zero-based accrual.
gap	In PeopleSoft Enterprise Campus Solutions, an artificial figure that sets aside an amount of unmet financial aid need that is not funded with Title IV funds. A gap can be used to prevent fully funding any student to conserve funds, or it can be used to preserve unmet financial aid need so that institutional funds can be awarded.
generic process type	In PeopleSoft Process Scheduler, process types are identified by a generic process type. For example, the generic process type SQR includes all SQR process types, such as SQR process and SQR report.
gift table	In PeopleSoft Enterprise Campus Solutions, a table or so-called <i>donor pyramid</i> describing the number and size of gifts that you expect will be needed to successfully complete the campaign in PeopleSoft Contributor Relations. The gift table enables you to estimate the number of donors and prospects that you need at each gift level to reach the campaign goal.
GL business unit	Abbreviation for <i>general ledger business unit</i> . A unit in an organization that is an independent entity for accounting purposes. It maintains its own set of accounting books. See also <i>business unit</i> .
GL entry template	Abbreviation for <i>general ledger entry template</i> . In PeopleSoft Enterprise Campus Solutions, a template that defines how a particular item is sent to the general ledger. An item-type maps to the general ledger, and the GL entry template can involve multiple general ledger accounts. The entry to the general ledger is further controlled by high-level flags that control the summarization and the type of accounting—that is, accrual or cash.
GL Interface process	Abbreviation for <i>General Ledger Interface process</i> . In PeopleSoft Enterprise Campus Solutions, a process that is used to send transactions from PeopleSoft Enterprise Student Financials to the general ledger. Item types are mapped to specific general ledger accounts, enabling transactions to move to the general ledger when the GL Interface process is run.
group	In PeopleSoft Billing and Receivables, a posting entity that comprises one or more transactions (items, deposits, payments, transfers, matches, or write-offs). In PeopleSoft Human Resources Management and Supply Chain Management, any set of records that are associated under a single name or variable to run calculations in PeopleSoft business processes. In PeopleSoft Time and Labor, for example, employees are placed in groups for time reporting purposes.
incentive object	In PeopleSoft Enterprise Incentive Management, the incentive-related objects that define and support the PeopleSoft Enterprise Incentive Management calculation process and results, such as plan templates, plans, results data, user interaction objects, and so on.

incentive rule	In PeopleSoft Sales Incentive Management, the commands that act on transactions and turn them into compensation. A rule is one part in the process of turning a transaction into compensation.
incur	In PeopleSoft Promotions Management, to become liable for a promotional payment. In other words, you owe that amount to a customer for promotional activities.
initiative	In PeopleSoft Enterprise Campus Solutions, the basis from which all advancement plans are executed. It is an organized effort targeting a specific constituency, and it can occur over a specified period of time with specific purposes and goals. An initiative can be a campaign, an event, an organized volunteer effort, a membership drive, or any other type of effort defined by the institution. Initiatives can be multipart, and they can be related to other initiatives. This enables you to track individual parts of an initiative, as well as entire initiatives.
inquiry access	In PeopleSoft Enterprise Campus Solutions, a type of security access that permits the user only to view data. See also <i>update access</i> .
institution	In PeopleSoft Enterprise Campus Solutions, an entity (such as a university or college) that is independent of other similar entities and that has its own set of rules and business processes.
integration	A relationship between two compatible integration points that enables communication to take place between systems. Integrations enable PeopleSoft applications to work seamlessly with other PeopleSoft applications or with third-party systems or software.
integration point	An interface that a system uses to communicate with another PeopleSoft application or an external application.
integration set	A logical grouping of integrations that applications use for the same business purpose. For example, the integration set ADVANCED_SHIPPING_ORDER contains all of the integrations that notify a customer that an order has shipped.
item	In PeopleSoft Inventory, a tangible commodity that is stored in a business unit (shipped from a warehouse). In PeopleSoft Demand Planning, Inventory Policy Planning, and Supply Planning, a noninventory item that is 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 (BOM) or planning routing, and it can exist as a component on a planning BOM. A planning item cannot be specified on a production or engineering BOM or routing, and it cannot be used as a component in a production. The quantity on hand will never be maintained. In PeopleSoft Receivables, an individual receivable. An item can be an invoice, a credit memo, a debit memo, a write-off, or an adjustment.
item shuffle	In PeopleSoft Enterprise Campus Solutions, a process that enables you to change a payment allocation without having to reverse the payment.
joint communication	In PeopleSoft Enterprise Campus Solutions, one letter that is addressed jointly to two people. For example, a letter might be addressed to both Mr. Sudhir Awat and Ms. Samantha Mortelli. A relationship must be established between the two individuals in the database, and at least one of the individuals must have an ID in the database.
keyword	In PeopleSoft Enterprise Campus Solutions, a term that you link to particular elements within PeopleSoft Student Financials, Financial Aid, and Contributor Relations. You can use keywords as search criteria that enable you to locate specific records in a search dialog box.

KPI	An abbreviation for <i>key performance indicator</i> . A high-level measurement of how well an organization is doing in achieving critical success factors. This defines the data value or calculation upon which an assessment is determined.
LDIF file	Abbreviation for <i>Lightweight Directory Access Protocol (LDAP) Data Interchange Format file</i> . Contains discrepancies between PeopleSoft data and directory data.
learner group	In PeopleSoft Enterprise Learning Management, a group of learners who are linked to the same learning environment. Members of the learner group can share the same attributes, such as the same department or job code. Learner groups are used to control access to and enrollment in learning activities and programs. They are also used to perform group enrollments and mass enrollments in the back office.
learning components	In PeopleSoft Enterprise Learning Management, the foundational building blocks of learning activities. PeopleSoft Enterprise Learning Management supports six basic types of learning components: web-based, session, webcast, test, survey, and assignment. One or more of these learning component types compose a single learning activity.
learning environment	In PeopleSoft Enterprise Learning Management, identifies a set of categories and catalog items that can be made available to learner groups. Also defines the default values that are assigned to the learning activities and programs that are created within a particular learning environment. Learning environments provide a way to partition the catalog so that learners see only those items that are relevant to them.
learning history	In PeopleSoft Enterprise Learning Management, a self-service repository for all of a learner's completed learning activities and programs.
ledger mapping	You use ledger mapping to relate expense data from general ledger accounts to resource objects. Multiple ledger line items can be mapped to one or more resource IDs. You can also use ledger mapping to map dollar amounts (referred to as <i>rates</i>) to business units. You can map the amounts in two different ways: an actual amount that represents actual costs of the accounting period, or a budgeted amount that can be used to calculate the capacity rates as well as budgeted model results. In PeopleSoft Enterprise Warehouse, you can map general ledger accounts to the EW Ledger table.
library section	In PeopleSoft Enterprise Incentive Management, a section that is defined in a plan (or template) and that is available for other plans to share. Changes to a library section are reflected in all plans that use it.
linked section	In PeopleSoft Enterprise Incentive Management, a section that is defined in a plan template but appears in a plan. Changes to linked sections propagate to plans using that section.
linked variable	In PeopleSoft Enterprise Incentive Management, a variable that is defined and maintained in a plan template and that also appears in a plan. Changes to linked variables propagate to plans using that variable.
LMS	Abbreviation for <i>learning management system</i> . In PeopleSoft Enterprise Campus Solutions, LMS is a PeopleSoft Student Records feature that provides a common set of interoperability standards that enable the sharing of instructional content and data between learning and administrative environments.
load	In PeopleSoft Inventory, identifies a group of goods that are shipped together. Load management is a feature of PeopleSoft Inventory that is used to track the weight, the volume, and the destination of a shipment.
local functionality	In PeopleSoft HRMS, the set of information that is available for a specific country. You can access this information when you click the appropriate country flag in the global window, or when you access it by a local country menu.

location	Locations enable you to indicate the different types of addresses—for a company, for example, one address to receive bills, another for shipping, a third for postal deliveries, and a separate street address. Each address has a different location number. The primary location—indicated by a <i>1</i> —is the address you use most often and may be different from the main address.
logistical task	In PeopleSoft Services Procurement, an administrative task that is related to hiring a service provider. Logistical tasks are linked to the service type on the work order so that different types of services can have different logistical tasks. Logistical tasks include both preapproval tasks (such as assigning a new badge or ordering a new laptop) and postapproval tasks (such as scheduling orientation or setting up the service provider email). The logistical tasks can be mandatory or optional. Mandatory preapproval tasks must be completed before the work order is approved. Mandatory postapproval tasks, on the other hand, must be completed before a work order is released to a service provider.
market template	In PeopleSoft Enterprise Incentive Management, additional functionality that is specific to a given market or industry and is built on top of a product category.
mass change	In PeopleSoft Enterprise Campus Solutions, mass change is a SQL generator that can be used to create specialized functionality. Using mass change, you can set up a series of Insert, Update, or Delete SQL statements to perform business functions that are specific to the institution. See also <i>3C engine</i> .
match group	In PeopleSoft Receivables, a group of receivables items and matching offset items. The system creates match groups by using user-defined matching criteria for selected field values.
MCF server	Abbreviation for <i>PeopleSoft MultiChannel Framework server</i> . Comprises the universal queue server and the MCF log server. Both processes are started when <i>MCF Servers</i> is selected in an application server domain configuration.
merchandising activity	In PeopleSoft Promotions Management, a specific discount type that is associated with a trade promotion (such as off-invoice, billback or rebate, or lump-sum payment) that defines the performance that is required to receive the discount. In the industry, you may know this as an offer, a discount, a merchandising event, an event, or a tactic.
meta-SQL	Meta-SQL constructs expand into platform-specific Structured Query Language (SQL) substrings. They are used in functions that pass SQL strings, such as in SQL objects, the SQLExec function, and PeopleSoft Application Engine programs.
metastring	Metastrings are special expressions included in SQL string literals. The metastrings, prefixed with a percent (%) symbol, are included directly in the string literals. They expand at run time into an appropriate substring for the current database platform.
multibook	In PeopleSoft General Ledger, multiple ledgers having multiple-base currencies that are defined for a business unit, with the option to post a single transaction to all base currencies (all ledgers) or to only one of those base currencies (ledgers).
multicurrency	The ability to process transactions in a currency other than the business unit's base currency.
national allowance	In PeopleSoft Promotions Management, a promotion at the corporate level that is funded by nondiscretionary dollars. In the industry, you may know this as a national promotion, a corporate promotion, or a corporate discount.
need	In PeopleSoft Enterprise Campus Solutions, the difference between the cost of attendance (COA) and the expected family contribution (EFC). It is the gap between the cost of attending the school and the student's resources. The financial aid package

	is based on the amount of financial need. The process of determining a student's need is called <i>need analysis</i> .
node-oriented tree	A tree that is based on a detail structure, but the detail values are not used.
pagelet	Each block of content on the home page is called a pagelet. These pagelets display summary information within a small rectangular area on the page. The pagelet provides users with a snapshot of their most relevant PeopleSoft and non-PeopleSoft content.
participant	In PeopleSoft Enterprise Incentive Management, participants are recipients of the incentive compensation calculation process.
participant object	Each participant object may be related to one or more compensation objects. See also <i>compensation object</i> .
partner	A company that supplies products or services that are resold or purchased by the enterprise.
pay cycle	In PeopleSoft Payables, a set of rules that define the criteria by which it should select scheduled payments for payment creation.
payment shuffle	In PeopleSoft Enterprise Campus Solutions, a process allowing payments that have been previously posted to a student's account to be automatically reapplied when a higher priority payment is posted or the payment allocation definition is changed.
pending item	In PeopleSoft Receivables, an individual receivable (such as an invoice, a credit memo, or a write-off) that has been entered in or created by the system, but hasn't been posted.
PeopleCode	PeopleCode is a proprietary language, executed by the PeopleSoft component processor. PeopleCode generates results based on existing data or user actions. By using various tools provided with PeopleTools, external services are available to all PeopleSoft applications wherever PeopleCode can be executed.
PeopleCode event	See <i>event</i> .
PeopleSoft Pure Internet Architecture	The fundamental architecture on which PeopleSoft 8 applications are constructed, consisting of a relational database management system (RDBMS), an application server, a web server, and a browser.
performance measurement	In PeopleSoft Enterprise Incentive Management, a variable used to store data (similar to an aggregator, but without a predefined formula) within the scope of an incentive plan. Performance measures are associated with a plan calendar, territory, and participant. Performance measurements are used for quota calculation and reporting.
period context	In PeopleSoft Enterprise Incentive Management, because a participant typically uses the same compensation plan for multiple periods, the period context associates a plan context with a specific calendar period and fiscal year. The period context references the associated plan context, thus forming a chain. Each plan context has a corresponding set of period contexts.
person of interest	A person about whom the organization maintains information but who is not part of the workforce.
personal portfolio	In PeopleSoft Enterprise Campus Solutions, the user-accessible menu item that contains an individual's name, address, telephone number, and other personal information.
plan	In PeopleSoft Sales Incentive Management, a collection of allocation rules, variables, steps, sections, and incentive rules that instruct the PeopleSoft Enterprise Incentive Management engine in how to process transactions.

plan context	In PeopleSoft Enterprise Incentive Management, correlates a participant with the compensation plan and node to which the participant is assigned, enabling the PeopleSoft Enterprise Incentive Management system to find anything that is associated with the node and that is required to perform compensation processing. Each participant, node, and plan combination represents a unique plan context—if three participants are on a compensation structure, each has a different plan context. Configuration plans are identified by plan contexts and are associated with the participants that refer to them.
plan template	In PeopleSoft Enterprise Incentive Management, the base from which a plan is created. A plan template contains common sections and variables that are inherited by all plans that are created from the template. A template may contain steps and sections that are not visible in the plan definition.
planned learning	In PeopleSoft Enterprise Learning Management, a self-service repository for all of a learner’s planned learning activities and programs.
planning instance	In PeopleSoft Supply Planning, a set of data (business units, items, supplies, and demands) constituting the inputs and outputs of a supply plan.
population	In PeopleSoft Enterprise Campus Solutions, the middle level of the three-level classification structure that you define in PeopleSoft Enterprise Recruiting and Admissions for enrollment management. You can define a population level, link it to other levels, and set enrollment target numbers for it. See also <i>division</i> and <i>cohort</i> .
portal registry	In PeopleSoft applications, the portal registry is a tree-like structure in which content references are organized, classified, and registered. It is a central repository that defines both the structure and content of a portal through a hierarchical, tree-like structure of folders useful for organizing and securing content references.
price list	In PeopleSoft Enterprise Pricer, enables you to select products and conditions for which the price list applies to a transaction. During a transaction, the system either determines the product price based on the predefined search hierarchy for the transaction or uses the product’s lowest price on any associated, active price lists. This price is used as the basis for any further discounts and surcharges.
price rule	In PeopleSoft Enterprise Pricer, defines the conditions that must be met for adjustments to be applied to the base price. Multiple rules can apply when conditions of each rule are met.
price rule condition	In PeopleSoft Enterprise Pricer, selects the price-by fields, the values for the price-by fields, and the operator that determines how the price-by fields are related to the transaction.
price rule key	In PeopleSoft Enterprise Pricer, defines the fields that are available to define price rule conditions (which are used to match a transaction) on the price rule.
primacy number	In PeopleSoft Enterprise Campus Solutions, a number that the system uses to prioritize financial aid applications when students are enrolled in multiple academic careers and academic programs at the same time. The Consolidate Academic Statistics process uses the primacy number indicated for both the career and program at the institutional level to determine a student’s primary career and program. The system also uses the number to determine the primary student attribute value that is used when you extract data to report on cohorts. The lowest number takes precedence.
primary name type	In PeopleSoft Enterprise Campus Solutions, the name type that is used to link the name stored at the highest level within the system to the lower-level set of names that an individual provides.

process category	In PeopleSoft Process Scheduler, processes that are grouped for server load balancing and prioritization.
process group	In PeopleSoft Financials, a group of application processes (performed in a defined order) that users can initiate in real time, directly from a transaction entry page.
process definition	Process definitions define each run request.
process instance	A unique number that identifies each process request. This value is automatically incremented and assigned to each requested process when the process is submitted to run.
process job	You can link process definitions into a job request and process each request serially or in parallel. You can also initiate subsequent processes based on the return code from each prior request.
process request	A single run request, such as a Structured Query Report (SQR), a COBOL or Application Engine program, or a Crystal report that you run through PeopleSoft Process Scheduler.
process run control	A PeopleTools variable used to retain PeopleSoft Process Scheduler values needed at runtime for all requests that reference a run control ID. Do not confuse these with application run controls, which may be defined with the same run control ID, but only contain information specific to a given application process request.
product	A PeopleSoft or third-party product. PeopleSoft organizes its software products into product families and product lines. Interactive Services Repository contains information about every release of every product that PeopleSoft sells, as well as products from certified third-party companies. These products are displayed with the product name and release number.
product category	In PeopleSoft Enterprise Incentive Management, indicates an application in the Enterprise Incentive Management suite of products. Each transaction in the PeopleSoft Enterprise Incentive Management system is associated with a product category.
product family	A group of products that are related by common functionality. The family names that can be searched using Interactive Service Repository are PeopleSoft Enterprise, PeopleSoft EnterpriseOne, PeopleSoft World, and third-party, certified PeopleSoft partners.
product line	The name of a PeopleSoft product line or the company name of a third-party certified partner. Integration Services Repository enables you to search for integration points by product line.
programs	In PeopleSoft Enterprise Learning Management, a high-level grouping that guides the learner along a specific learning path through sections of catalog items. PeopleSoft Enterprise Learning Systems provides two types of programs—curricula and certifications.
progress log	In PeopleSoft Services Procurement, tracks deliverable-based projects. This is similar to the time sheet in function and process. The service provider contact uses the progress log to record and submit progress on deliverables. The progress can be logged by the activity that is performed, by the percentage of work that is completed, or by the completion of milestone activities that are defined for the project.
project transaction	In PeopleSoft Project Costing, an individual transaction line that represents a cost, time, budget, or other transaction row.
promotion	In PeopleSoft Promotions Management, a trade promotion, which is typically funded from trade dollars and used by consumer products manufacturers to increase sales volume.

prospects	<p>In PeopleSoft Enterprise Campus Solutions, students who are interested in applying to the institution.</p> <p>In PeopleSoft Enterprise Contributor Relations, individuals and organizations that are most likely to make substantial financial commitments or other types of commitments to the institution.</p>
publishing	In PeopleSoft Enterprise Incentive Management, a stage in processing that makes incentive-related results available to participants.
rating components	In PeopleSoft Enterprise Campus Solutions, variables used with the Equation Editor to retrieve specified populations.
record group	A set of logically and functionally related control tables and views. Record groups help enable TableSet sharing, which eliminates redundant data entry. Record groups ensure that TableSet sharing is applied consistently across all related tables and views.
record input VAT flag	Abbreviation for <i>record input value-added tax flag</i> . Within PeopleSoft Purchasing, Payables, and General Ledger, this flag indicates that you are recording input VAT on the transaction. This flag, in conjunction with the record output VAT flag, is used to determine the accounting entries created for a transaction and to determine how a transaction is reported on the VAT return. For all cases within Purchasing and Payables where VAT information is tracked on a transaction, this flag is set to Yes. This flag is not used in PeopleSoft Order Management, Billing, or Receivables, where it is assumed that you are always recording only output VAT, or in PeopleSoft Expenses, where it is assumed that you are always recording only input VAT.
record output VAT flag	<p>Abbreviation for <i>record output value-added tax flag</i>.</p> <p>See <i>record input VAT flag</i>.</p>
recname	The name of a record that is used to determine the associated field to match a value or set of values.
recognition	In PeopleSoft Enterprise Campus Solutions, the recognition type indicates whether the PeopleSoft Enterprise Contributor Relations donor is the primary donor of a commitment or shares the credit for a donation. Primary donors receive hard credit that must total 100 percent. Donors that share the credit are given soft credit. Institutions can also define other share recognition-type values such as memo credit or vehicle credit.
reference data	In PeopleSoft Sales Incentive Management, system objects that represent the sales organization, such as territories, participants, products, customers, channels, and so on.
reference object	In PeopleSoft Enterprise Incentive Management, this dimension-type object further defines the business. Reference objects can have their own hierarchy (for example, product tree, customer tree, industry tree, and geography tree).
reference transaction	In commitment control, a reference transaction is a source transaction that is referenced by a higher-level (and usually later) source transaction, in order to automatically reverse all or part of the referenced transaction's budget-checked amount. This avoids duplicate postings during the sequential entry of the transaction at different commitment levels. For example, the amount of an encumbrance transaction (such as a purchase order) will, when checked and recorded against a budget, cause the system to concurrently reference and relieve all or part of the amount of a corresponding pre-encumbrance transaction, such as a purchase requisition.
regional sourcing	In PeopleSoft Purchasing, provides the infrastructure to maintain, display, and select an appropriate vendor and vendor pricing structure that is based on a regional sourcing model where the multiple ship to locations are grouped. Sourcing may occur at a level higher than the ship to location.

relationship object	In PeopleSoft Enterprise Incentive Management, these objects further define a compensation structure to resolve transactions by establishing associations between compensation objects and business objects.
remote data source data	Data that is extracted from a separate database and migrated into the local database.
REN server	Abbreviation for <i>real-time event notification server</i> in PeopleSoft MultiChannel Framework.
requester	In PeopleSoft eSettlements, an individual who requests goods or services and whose ID appears on the various procurement pages that reference purchase orders.
reversal indicator	In PeopleSoft Enterprise Campus Solutions, an indicator that denotes when a particular payment has been reversed, usually because of insufficient funds.
role	Describes how people fit into PeopleSoft Workflow. A role is a class of users who perform the same type of work, such as clerks or managers. Your business rules typically specify what user role needs to do an activity.
role user	A PeopleSoft Workflow user. A person's role user ID serves much the same purpose as a user ID does in other parts of the system. PeopleSoft Workflow uses role user IDs to determine how to route worklist items to users (through an email address, for example) and to track the roles that users play in the workflow. Role users do not need PeopleSoft user IDs.
roll up	In a tree, to roll up is to total sums based on the information hierarchy.
run control	A run control is a type of online page that is used to begin a process, such as the batch processing of a payroll run. Run control pages generally start a program that manipulates data.
run control ID	A unique ID to associate each user with his or her own run control table entries.
run-level context	In PeopleSoft Enterprise Incentive Management, associates a particular run (and batch ID) with a period context and plan context. Every plan context that participates in a run has a separate run-level context. Because a run cannot span periods, only one run-level context is associated with each plan context.
SCP SCBM XML message	Abbreviation for <i>Supply Chain Planning Supply Chain Business Modeler Extensible Markup Language message</i> . PeopleSoft EnterpriseOne Supply Chain Business Modeler uses XML as the format for all data that it imports and exports.
search query	You use this set of objects to pass a query string and operators to the search engine. The search index returns a set of matching results with keys to the source documents.
search/match	In PeopleSoft Enterprise Campus Solutions and PeopleSoft Enterprise Human Resources Management Solutions, a feature that enables you to search for and identify duplicate records in the database.
seasonal address	In PeopleSoft Enterprise Campus Solutions, an address that recurs for the same length of time at the same time of year each year until adjusted or deleted.
section	In PeopleSoft Enterprise Incentive Management, a collection of incentive rules that operate on transactions of a specific type. Sections enable plans to be segmented to process logical events in different sections.
security event	In commitment control, security events trigger security authorization checking, such as budget entries, transfers, and adjustments; exception overrides and notifications; and inquiries.
serial genealogy	In PeopleSoft Manufacturing, the ability to track the composition of a specific, serial-controlled item.

serial in production	In PeopleSoft Manufacturing, enables the tracing of serial information for manufactured items. This is maintained in the Item Master record.
service impact	In PeopleSoft Enterprise Campus Solutions, the resulting action triggered by a service indicator. For example, a service indicator that reflects nonpayment of account balances by a student might result in a service impact that prohibits registration for classes.
service indicator	In PeopleSoft Enterprise Campus Solutions, indicates services that may be either withheld or provided to an individual. Negative service indicators indicate holds that prevent the individual from receiving specified services, such as check-cashing privileges or registration for classes. Positive service indicators designate special services that are provided to the individual, such as front-of-line service or special services for disabled students.
session	<p>In PeopleSoft Enterprise Campus Solutions, time elements that subdivide a term into multiple time periods during which classes are offered. In PeopleSoft Contributor Relations, a session is the means of validating gift, pledge, membership, or adjustment data entry . It controls access to the data entered by a specific user ID. Sessions are balanced, queued, and then posted to the institution's financial system. Sessions must be posted to enter a matching gift or pledge payment, to make an adjustment, or to process giving clubs or acknowledgements.</p> <p>In PeopleSoft Enterprise Learning Management, a single meeting day of an activity (that is, the period of time between start and finish times within a day). The session stores the specific date, location, meeting time, and instructor. Sessions are used for scheduled training.</p>
session template	In PeopleSoft Enterprise Learning Management, enables you to set up common activity characteristics that may be reused while scheduling a PeopleSoft Enterprise Learning Management activity—characteristics such as days of the week, start and end times, facility and room assignments, instructors, and equipment. A session pattern template can be attached to an activity that is being scheduled. Attaching a template to an activity causes all of the default template information to populate the activity session pattern.
setup relationship	In PeopleSoft Enterprise Incentive Management, a relationship object type that associates a configuration plan with any structure node.
share driver expression	In PeopleSoft Business Planning, a named planning method similar to a driver expression, but which you can set up globally for shared use within a single planning application or to be shared between multiple planning applications through PeopleSoft Enterprise Warehouse.
single signon	With single signon, users can, after being authenticated by a PeopleSoft application server, access a second PeopleSoft application server without entering a user ID or password.
source key process	In PeopleSoft Enterprise Campus Solutions, a process that relates a particular transaction to the source of the charge or financial aid. On selected pages, you can drill down into particular charges.
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.
speed key	See <i>communication key</i> .
SpeedChart	A user-defined shorthand key that designates several ChartKeys to be used for voucher entry. Percentages can optionally be related to each ChartKey in a SpeedChart definition.

SpeedType	A code representing a combination of ChartField values. SpeedTypes simplify the entry of ChartFields commonly used together.
staging	A method of consolidating selected partner offerings with the offerings from the enterprise's other partners.
standard letter code	In PeopleSoft Enterprise Campus Solutions, a standard letter code used to identify each letter template available for use in mail merge functions. Every letter generated in the system must have a standard letter code identification.
statutory account	Account required by a regulatory authority for recording and reporting financial results. In PeopleSoft, this is equivalent to the Alternate Account (ALTACCT) ChartField.
step	In PeopleSoft Sales Incentive Management, a collection of sections in a plan. Each step corresponds to a step in the job run.
storage level	In PeopleSoft Inventory, identifies the level of a material storage location. Material storage locations are made up of a business unit, a storage area, and a storage level. You can set up to four storage levels.
subcustomer qualifier	A value that groups customers into a division for which you can generate detailed history, aging, events, and profiles.
Summary ChartField	You use summary ChartFields to create summary ledgers that roll up detail amounts based on specific detail values or on selected tree nodes. When detail values are summarized using tree nodes, summary ChartFields must be used in the summary ledger data record to accommodate the maximum length of a node name (20 characters).
summary ledger	An accounting feature used primarily in allocations, inquiries, and PS/nVision reporting to store combined account balances from detail ledgers. Summary ledgers increase speed and efficiency of reporting by eliminating the need to summarize detail ledger balances each time a report is requested. Instead, detail balances are summarized in a background process according to user-specified criteria and stored on summary ledgers. The summary ledgers are then accessed directly for reporting.
summary time period	In PeopleSoft Business Planning, any time period (other than a base time period) that is an aggregate of other time periods, including other summary time periods and base time periods, such as quarter and year total.
summary tree	A tree used to roll up accounts for each type of report in summary ledgers. Summary trees enable you to define trees on trees. In a summary tree, the detail values are really nodes on a detail tree or another summary tree (known as the <i>basis</i> tree). A summary tree structure specifies the details on which the summary trees are to be built.
syndicate	To distribute a production version of the enterprise catalog to partners.
system function	In PeopleSoft Receivables, an activity that defines how the system generates accounting entries for the general ledger.
TableSet	A means of sharing similar sets of values in control tables, where the actual data values are different but the structure of the tables is the same.
TableSet sharing	Shared data that is stored in many tables that are based on the same TableSets. Tables that use TableSet sharing contain the SETID field as an additional key or unique identifier.
target currency	The value of the entry currency or currencies converted to a single currency for budget viewing and inquiry purposes.

tax authority	In PeopleSoft Enterprise Campus Solutions, a user-defined element that combines a description and percentage of a tax with an account type, an item type, and a service impact.
template	A template is HTML code associated with a web page. It defines the layout of the page and also where to get HTML for each part of the page. In PeopleSoft, you use templates to build a page by combining HTML from a number of sources. For a PeopleSoft portal, all templates must be registered in the portal registry, and each content reference must be assigned a template.
territory	In PeopleSoft Sales Incentive Management, hierarchical relationships of business objects, including regions, products, customers, industries, and participants.
third party	A company or vendor that has extensive PeopleSoft product knowledge and whose products and integrations have been certified and are compatible with PeopleSoft applications.
3C engine	Abbreviation for <i>Communications, Checklists, and Comments engine</i> . In PeopleSoft Enterprise Campus Solutions, the 3C engine enables you to automate business processes that involve additions, deletions, and updates to communications, checklists, and comments. You define events and triggers to engage the engine, which runs the mass change and processes the 3C records (for individuals or organizations) immediately and automatically from within business processes.
3C group	Abbreviation for <i>Communications, Checklists, and Comments group</i> . In PeopleSoft Enterprise Campus Solutions, a method of assigning or restricting access privileges. A 3C group enables you to group specific communication categories, checklist codes, and comment categories. You can then assign the group inquiry-only access or update access, as appropriate.
TimeSpan	A relative period, such as year-to-date or current period, that can be used in various PeopleSoft General Ledger functions and reports when a rolling time frame, rather than a specific date, is required. TimeSpans can also be used with flexible formulas in PeopleSoft Projects.
trace usage	In PeopleSoft Manufacturing, enables the control of which components will be traced during the manufacturing process. Serial- and lot-controlled components can be traced. This is maintained in the Item Master record.
transaction allocation	In PeopleSoft Enterprise Incentive Management, the process of identifying the owner of a transaction. When a raw transaction from a batch is allocated to a plan context, the transaction is duplicated in the PeopleSoft Enterprise Incentive Management transaction tables.
transaction state	In PeopleSoft Enterprise Incentive Management, a value assigned by an incentive rule to a transaction. Transaction states enable sections to process only transactions that are at a specific stage in system processing. After being successfully processed, transactions may be promoted to the next transaction state and “picked up” by a different section for further processing.
Translate table	A system edit table that stores codes and translate values for the miscellaneous fields in the database that do not warrant individual edit tables of their own.
tree	The graphical hierarchy in PeopleSoft systems that displays the relationship between all accounting units (for example, corporate divisions, projects, reporting groups, account numbers) and determines roll-up hierarchies.
tuition lock	In PeopleSoft Enterprise Campus Solutions, a feature in the Tuition Calculation process that enables you to specify a point in a term after which students are charged a minimum (or <i>locked</i>) fee amount. Students are charged the locked fee amount even if they later drop classes and take less than the normal load level for that tuition charge.

unclaimed transaction	In PeopleSoft Enterprise Incentive Management, a transaction that is not claimed by a node or participant after the allocation process has completed, usually due to missing or incomplete data. Unclaimed transactions may be manually assigned to the appropriate node or participant by a compensation administrator.
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.
update access	In PeopleSoft Enterprise Campus Solutions, a type of security access that permits the user to edit and update data. See also <i>inquiry access</i> .
user interaction object	In PeopleSoft Sales Incentive Management, used to define the reporting components and reports that a participant can access in his or her context. All Sales Incentive Management user interface objects and reports are registered as user interaction objects. User interaction objects can be linked to a compensation structure node through a compensation relationship object (individually or as groups).
variable	In PeopleSoft Sales Incentive Management, the intermediate results of calculations. Variables hold the calculation results and are then inputs to other calculations. Variables can be plan variables that persist beyond the run of an engine or local variables that exist only during the processing of a section.
VAT exception	Abbreviation for <i>value-added tax exception</i> . A temporary or permanent exemption from paying VAT that is granted to an organization. This term refers to both VAT exoneration and VAT suspension.
VAT exempt	Abbreviation for <i>value-added tax exempt</i> . Describes goods and services that are not subject to VAT. Organizations that supply exempt goods or services are unable to recover the related input VAT. This is also referred to as exempt without recovery.
VAT exoneration	Abbreviation for <i>value-added tax exoneration</i> . An organization that has been granted a permanent exemption from paying VAT due to the nature of that organization.
VAT suspension	Abbreviation for <i>value-added tax suspension</i> . An organization that has been granted a temporary exemption from paying VAT.
warehouse	A PeopleSoft data warehouse that consists of predefined ETL maps, data warehouse tools, and DataMart definitions.
work order	In PeopleSoft Services Procurement, enables an enterprise to create resource-based and deliverable-based transactions that specify the basic terms and conditions for hiring a specific service provider. When a service provider is hired, the service provider logs time or progress against the work order.
worker	A person who is part of the workforce; an employee or a contingent worker.
workset	A group of people and organizations that are linked together as a set. You can use worksets to simultaneously retrieve the data for a group of people and organizations and work with the information on a single page.
worksheet	A way of presenting data through a PeopleSoft Business Analysis Modeler interface that enables users to do in-depth analysis using pivoting tables, charts, notes, and history information.
worklist	The automated to-do list that PeopleSoft Workflow creates. From the worklist, you can directly access the pages you need to perform the next action, and then return to the worklist for another item.

XML link	The XML Linking language enables you to insert elements into XML documents to create a links between resources.
XML schema	An XML definition that standardizes the representation of application messages, component interfaces, or business interlinks.
XPI	Abbreviation for <i>eXtended Process Integrator</i> . PeopleSoft XPI is the integration infrastructure that enables both real-time and batch communication with EnterpriseOne applications.
yield by operation	In PeopleSoft Manufacturing, the ability to plan the loss of a manufactured item on an operation-by-operation basis.
zero-rated VAT	Abbreviation for <i>zero-rated value-added tax</i> . 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. Organizations that supply zero-rated goods and services can still recover the related input VAT. This is also referred to as exempt with recovery.

Index

Numerics/Symbols

- 998 record format 25
- 999 record format 25

A

- action codes
 - defining 6
 - understanding 5
- additional documentation xii
- API calls (ODBC driver) 71
- Application Designer
 - creating staging tables and work records 27
 - extracting data 43
 - implementing routings 57
- application fundamentals xi
- audit inbound map 55

B

- business entities
 - defining 19
 - specifying types 31
- Business Entity Definition page 19

C

- calculation options
 - defining outbound PeopleSoft Business Document format 41
 - specifying staging tables 33
- catalog procedures
 - SQLProcedureColumns 83
 - SQLProcedures 82
- codes
 - defining action codes 6
 - defining event codes 6
 - defining PeopleSoft entity codes 16
 - identifying primary and secondary event codes 31
 - selecting entity codes 31
 - translating transaction codes 6
 - understanding action and event codes 5
 - understanding EDI and PeopleSoft codes 5
- comments, submitting xvi
- common elements xvii

- Component Interfaces 57
- connection functions support 76
- connection procedures, PeopleTools
 - SQLConnect 76
 - SQLDriverConnect 77
 - using 75
- contact information xvi
- control fields, EDI 43
- Conversion Data Profile page 8
- conversion data profiles, creating 8
- conversion IDs, assigning 7
- Conversion Type Definition page 7
- cross-references xv
- Customer Connection website xii
- customers, setting up trading partners 18

D

- data conversion
 - assigning external trading partner IDs 18
 - specifying values 6
- Data Definition Language (DDL) 85
- Data Manipulation Language (DML) 85
- data source name (DSN) 75
- DDL, *See* Data Definition Language (DDL)
- deleting EDI Manager objects 20
- DML, *See* Data Manipulation Language (DDL)
- documentation
 - printed xii
 - related xii
 - updates xii
- DSN, *See* data source name (DSN)

E

- Ec Delete page 20
- EC Map Profile Def page 42
- EDI 3
 - See Also* EDI agents; EDI Manager
 - converting codes 5
 - entering control field values 43
 - monitoring processes 47
 - understanding 3
- EDI agents
 - scheduling inbound 49

- scheduling outbound 51
 - understanding 47
 - EDI codes 5
 - EDI Manager
 - about implementing 1
 - converting codes 5
 - creating map profiles 42
 - defining action codes 6
 - defining EDI transactions 10
 - defining event codes 6
 - defining inbound maps 26
 - defining outbound maps 35
 - deleting objects 20
 - extracting data 43
 - loading non-X.1/EDIFACT EDI transactions 3
 - overview 1
 - preparing outbound maps 47
 - reviewing errors, packages and transaction groups 54
 - scheduling inbound EDI agents 49
 - scheduling outbound EDI agents 51
 - setting up trading partners 15
 - specifying data conversion values 6
 - using 3
 - EDIFACT
 - EDI transactions 3
 - format 4
 - electronic commerce maps
 - loading/extracting applications 26
 - understanding 23
 - Electronic Data Interchange, *See* EDI
 - Entity Code Definition page 16
 - entity codes
 - defining 16
 - selecting 31
 - error retrieval procedures 98
 - errors
 - PsfGetLastError 64
 - retrieval procedures 98
 - reviewing and correcting 54
 - SQLError 98
 - event codes
 - converting to action codes 14
 - defining 6
 - identifying primary and secondary 31
 - understanding 5
 - execution models
 - ExecDirect 95
 - prepare/execute 95
 - stored procedure 96
 - External Partner Definition page 18
 - external reporting tools 71
 - external trading partners, assigning IDs 18
 - See Also* trading partners
- F**
- fields
 - calculation option values 33, 41
 - counting form fields 60
 - EDI control 43
 - for records 24
 - listing form fields 61
 - required for staging tables 27
 - file formats
 - converting 4
 - PeopleSoft business documents 24
 - file layouts 3
 - forms routings, defining 57
- G**
- getting started 1
 - glossary 105
- I**
- implementation overview
 - EDI Manager 1
 - ODBC API 1
 - ODBC Driver 1
 - Open Query 1
 - Outgoing Forms API 1
 - inbound EDI agents
 - scheduling 49
 - understanding 47
 - inbound maps
 - audit 55
 - creating 28
 - defining 26
 - understanding 23
 - Inbound Maps — Business Document Layout page 29
 - Inbound Maps — Description page 28
 - Inbound Maps — Target Records page 31
 - information procedures
 - SQLFunctions 80
 - SQLGetInfo 79
 - SQLGetTypeInfo 81
 - using 79

- initialization procedures
 - SQLAllocConnect 74
 - SQLAllocEnv 74
 - SQLFreeEnv 75
- Internal Partner Definition page 17
- internal partners, creating IDs 17
 - See Also* trading partners

M

- MAPI 57
- maps
 - audit inbound maps 55
 - creating inbound 28
 - creating outbound 36
 - creating profiles 42
 - defining inbound 26
 - defining outbound 35
 - preparing outbound 47
 - understanding 23
- metadata (catalog procedures) 82
- Microsoft ODBC SDK reference
 - manual 79
- MMA Partners xii
- MS Windows registry, adding entries 57
- MVS, running EDI agent on a server 50

N

- notes xv

O

- ODBC API
 - about implementing 1
 - compliance 101
 - executing queries (example) 103
 - Internet access 71
 - overview 1
 - understanding 67, 70
- ODBC driver
 - executing queries (example) 103
 - external reporting tools 71
 - mapping to RDM data types 102
 - Microsoft ODBC SDK reference
 - manual 79
 - retrieving data 96
 - retrieving errors 98
 - supported API calls 71
 - supported execution models 95
 - supporting connection functions 76
 - supporting keywords 77

- supporting languages 85
- supporting query and column
 - listings 82
- supporting SQL 86
- understanding 67

ODBC Driver

- about implementing 1
- overview 1

open query

- architecture 68
- components 69

Open Query 67

- See Also* ODBC API; ODBC driver
 - overview 1

operations

- query 60
- send 65
- session-level 58

Outbound EDI Agent Preparer

- creating Outbound EDI Agent SQC 42
- processing outbound EDI
 - transactions 48

outbound EDI agents

- scheduling 51
- understanding 47

outbound maps

- creating 36
- defining 35
- preparing 47

running Outbound EDI Agent

- Preparer 42, 48
- understanding 23

Outbound Maps — Description page 36**Outbound Maps — Target Business Doc Layout page 38****Outgoing Forms API**

- about implementing 1
- overview 1

P**packages, reviewing 54****partner profiles**

- assigning to trading partners 18
- defining 13
- setting defaults 14

PeopleBooks

- ordering xii

PeopleCode, typographical

- conventions xiv

PeopleSoft Application Designer, *See*
 Application Designer
 PeopleSoft application fundamentals xi
 PeopleSoft business documents
 converting to 4, 8
 defining inbound format 29
 defining outbound format 38
 understanding 24
 using EDI agents 47
 PeopleSoft codes 5
 PeopleSoft Component Interfaces, *See*
 Component Interfaces
 PeopleSoft EDI Manager, *See* EDI Manager
 PeopleSoft Open Query, *See* Open Query
 PeopleTools
 connection procedures 75
 initialization procedures 74
 prerequisites xi
 primary event codes 5
 printed documentation xii
 procedures
 catalog 82
 connection 75
 information 79
 retrieval 96
 SQL execution 85
 status and error retrieval 98
 transaction and connection
 termination 99
 Profile Defaults page 14
 Profile Definition page 13
 PsfCloseSession 58
 PsfGetAPIInfo 58
 PsfGetFieldCount 60
 PsfGetFieldList 61
 PsfGetFormCount 62
 PsfGetFormList 63
 PsfGetLastError 64
 PsfOpenSession 59
 PSFORMS.DLL 57
 PsfSendForm 65
 purpose codes, *See* primary event codes

Q

query operations
 PsfGetFieldCount 60
 PsfGetFieldList 61
 PsfGetFormCount 62
 PsfGetFormList 63
 PsfGetLastError 64

queue control numbers 31

R

records 24
 See Also work records
 999 and 998 formats 25
 PeopleSoft business documents 24
 selecting IDs 31
 specifying source 37
 specifying staging tables 31
 registry (MS Windows), adding entries 57
 related documentation xii
 reporting tools, external 71
 retrieval procedures
 SQLFetch 96
 SQLGetData 97
 using 96
 reviewing errors, packages and transaction
 groups 54
 routings, designing forms 57
 Run Control Parameters page 49
 Run Parameters page 47
 preparing outbound maps 47
 scheduling outbound EDI agents 51

S

scheduling
 inbound EDI agents 49
 outbound EDI agents 51
 secondary event codes 5
 send operations 65
 PsfSendForm 65
 session-level operations
 PsfCloseSession 58
 PsfGetAPIInfo 58
 PsfOpenSession 59
 setIDs 10
 source records, specifying 37
 SQL Access Group Call Level
 Interface 101
 SQL execution procedures
 SQLAllocStmt 86
 SQLBindCol 93
 SQLBindParameter 94
 SQLColAttributes 89
 SQLDescribeCol 90
 SQLDescribeParam 91
 SQLExecDirect 87
 SQLExecute 88

- SQLGetRowCount 91
- SQLNumParams 92
- SQLNumResultCols 92
- SQLPrepare 88
 - using 85
- SQLAllocConnect 74
- SQLAllocEnv 74
- SQLAllocStmt 86
- SQLBindCol 93
- SQLBindParameter 94
- SQLColAttributes 89
- SQLConnect 76
- SQLDescribeCol 90
- SQLDescribeParam 91
- SQLDisconnect 100
- SQLDriverConnect 77
- SQLError 98
- SQLExecDirect 87
- SQLExecute 88
- SQLFetch 96
- SQLFreeConnect 100
- SQLFreeEnv 75, 101
- SQLFunctions 80
- SQLGetData 97
- SQLGetInfo 79
- SQLGetRowCount 91
- SQLGetTypeInfo 81
- SQLNumParams 92
- SQLNumResultCols 92
- SQLPrepare 88
- SQLProcedureColumns 83
- SQLProcedures 82
- SQLTransact 99
- staging table fields
 - ECQUEUEINSTANCE 27
 - ECTRANSID 27
 - ECTRANSINOUTSW 27
- staging tables
 - creating 27
 - EDI architecture 4
 - specifying 31
- status retrieval procedures 98
- suggestions, submitting xvi
- Supported Integration Technologies
 - about implementing 1
 - getting started 1
 - other sources of information 2
 - overview of 1

T

- terms 105
- third-party applications
 - accessing PeopleSoft data 67
 - communicating with forms software 57
- trading partners
 - assigning conversion IDs 7
 - assigning external partner IDs 18
 - assigning map profiles 42
 - conversion profiles 6
 - creating internal partner IDs 17
 - defining business entities 19
 - setting up 15
 - understanding 15
- transaction and connection termination
 - procedures 99
 - SQLDisconnect 100
 - SQLFreeConnect 100
 - SQLFreeEnv 101
 - SQLTransact 99
- transaction codes, *See* secondary event codes
- Transaction Definition page 11
- transaction groups, reviewing 54
- transaction IDs 11
- transactions
 - adding 23
 - creating IDs 11
 - defining 10
 - mapping 23
- typographical conventions xiv

V

- Value-Added Network (VAN) 4
- VAN, *See* Value-Added Network (VAN)
- vendors, setting up trading partners 18
- VIM 57
- visual cues xv

W

- warnings xv
- work records 27
 - See Also* records
 - creating 27
 - selecting IDs 31
 - specifying definitions (inbound) 29
 - specifying definitions (outbound) 38

X

- X.1 EDI transactions 3
- X.12 format 4
- X/Open 101