



PeopleTools 8.12 Globalization
PeopleBook

SKU MTGBr8SP1B 1200

PeopleBooks Contributors: Teams from PeopleSoft Product Documentation and Development.

Copyright © 2001 by PeopleSoft, Inc. All rights reserved.

Printed in the United States of America.

All material contained in this documentation is proprietary and confidential to PeopleSoft, Inc. and is protected by copyright laws. 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, Inc.

This documentation is subject to change without notice, and PeopleSoft, Inc. 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, Inc. in writing.

The copyrighted software that accompanies this documentation 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 documentation, including the disclosure thereof.

PeopleSoft, the PeopleSoft logo, PeopleTools, PS/nVision, PeopleCode, PeopleBooks, Vantive, and Vantive Enterprise are registered trademarks, and *PeopleTalk* and "People power the internet." are trademarks of PeopleSoft, Inc. All other company and product names may be trademarks of their respective owners.

Contents

About This PeopleBook

Before You Begin	xiv
Related Documentation	xiv
Documentation on the Internet.....	xiv
Documentation on CD-ROM	xv
Hardcopy Documentation	xv
Typographical Conventions and Visual Cues.....	xv
Comments and Suggestions.....	xvii

Chapter 1

Overview of PeopleSoft Globalization

Globalization Features	1-1
International Language Architecture.....	1-1
International Preference Settings	1-1
Related Language Tables	1-1
Global Configuration Considerations	1-2
Translation Features	1-2
Translating Application Data	1-3
Translating PeopleTools Objects	1-3
Translating Windows Resources.....	1-4
Translating Terms	1-4
International Formatting.....	1-4
Understanding Time Zones	1-4
Controlling Currency Display	1-5
Working With Character Sets	1-5
Global Reporting	1-6
Planning a Globalization Strategy	1-6
Choosing a Base Language	1-7
Selective Translation.....	1-7
Planning for Globalization Dependencies.....	1-7
Other International Considerations.....	1-8
Language Considerations for Upgrade.....	1-8
Global Documentation	1-8

Chapter 2

Controlling International Preferences

Understanding International Preferences.....	2-1
Understanding Language Codes.....	2-1
PeopleSoft Internet Architecture Language Preferences.....	2-2
Windows Client Language Preferences	2-2
PeopleSoft Internet Architecture Language Preferences	2-3
Controlling the Signon Language	2-3
Choosing a Signon Language.....	2-3
Changing the Signon Language While Signed On.....	2-5
Changing the Signon Language Programmatically.....	2-6
Determining the Signon Language When the Signon Screen Is Bypassed.....	2-6
Understanding the Browser Language Preference	2-7
Setting Browser Language Preferences.....	2-7
Applying Browser Preferences to Determine the Default Signon Language.....	2-7
Applying Browser Language Preferences to Determine Formatting Rules	2-8
Controlling the User Profile Language Preference	2-10
Windows Client Language Preferences.....	2-12
Controlling the Configuration Manager Language Setting.....	2-12
Controlling Windows Regional Settings.....	2-13

Chapter 3

Global Configuration Considerations

Swapping the Base Language	3-1
Before You Swap the Base Language.....	3-1
Running the Auditing Report	3-1
Checking Space and Resources.....	3-3
Swapping the Base Language	3-3
Creating New Languages.....	3-4
Determining PeopleSoft and ISO Standard Codes for Your Language	3-5
Determining the Appropriate Non-Unicode Character Set	3-6
Adding the New Language to the Translate Table (XLATTABLE).....	3-7
Adding the New Language to the PSLANGUAGES Table.....	3-8
Modifying the Configuration Manager Windows Resources.....	3-9
Modifying the PeopleSoft Internet Architecture Signon Page.....	3-10
Adding the New Language to the pstools.properties File	3-11
Loading Translations	3-12
Adding Translations to an Existing Database	3-13
Upgrading Your Customizations into a Multilingual Database	3-15

Chapter 4

Understanding Related Language Tables

Related Language Table Structure.....	4-2
How Related Language Tables Store Translations.....	4-4
How Related Language Tables Are Used.....	4-4
Creating Related Language Tables.....	4-6
Creating Related Language Views	4-8
One Base Table, One Related Language Table.....	4-8
Two Base Tables, One Related Language Table	4-9
Two Base Tables, Two Related Language Tables	4-11
The Translate Table (XLATTABLE)	4-14

Chapter 5

Working With Language-Sensitive Application Data

Translating Data by Changing the Signon Language	5-1
How the System Maintains Related Language Tables and Base Tables	5-3
Editing Data in Multiple Languages Using MLS	5-4

Chapter 6

Translating With Design Tools

Choosing a Translation Tool	6-1
Overview of System Object Translation.....	6-2
Translating Object Descriptions and Labels Using PeopleSoft Application Designer....	6-2
Overview of Description and Label Translation	6-3
Translating Field Descriptions in PeopleSoft Application Designer	6-5
Translating Record Descriptions in PeopleSoft Application Designer	6-5
Translating Menu Navigation in Application Designer	6-6
Translating Folder Tab Labels	6-9
Translating Icon Labels on Workflow Maps.....	6-10
Translating PeopleSoft Process Scheduler Definitions	6-12
Translating Tree Descriptions and Tree Node Labels.....	6-13
Translating Non-Text Objects	6-15
Translating Images	6-15
Translating HTML Objects	6-16
Translating Translate Table Values	6-16
Globalizing Pages	6-18
Designing Global-Ready, Base Language Pages	6-19
Creating a Non-Base Language Page	6-20

Chapter 7

Using the Translate Utilities

Using the Translation Workbench	7-1
Opening the Translation Grid.....	7-2
Translation Grid Display Options	7-4
Understanding the Information Bar	7-4
Expanding and Collapsing Nodes	7-5
Choosing Which Rows to View	7-5
Resizing Columns and Rows	7-6
Working in the Translation Grid	7-6
Entering Translations	7-6
Navigating the Translation Grid.....	7-7
Searching and Replacing.....	7-8
Using Application Designer Functionality.....	7-8
External Integration.....	7-9
Using the Translate Pages.....	7-9
Checking for Override Text in Pages	7-21

Chapter 8

Translating Windows Resources

Before You Begin.....	8-2
Required Files	8-2
Required Tools	8-2
Understanding Alternate Language DLLs	8-2
Locating Resource Directories.....	8-3
Translating Resource Files	8-4
Using MAKEALTL.BAT	8-5
Using MAKEUNIX.BAT.....	8-6

Chapter 9

Modifying Terminology

Terminology Management Overview	9-1
Searching.....	9-1
Replacing.....	9-2
Undoing.....	9-2
Understanding Search Statuses	9-3
Granting Search Privileges	9-3
Searching	9-5
Defining Search Criteria.....	9-5
Cloning Existing Search Criteria.....	9-10

Sample Search Criteria.....	9-12
Searching Base Language Text.....	9-12
Searching for Base Language Text With Existing Translations	9-13
Searching Non-Base Language Text.....	9-13
Replacing Non-Base Language Text.....	9-14
Searching Base Language and Non-Base Language Text	9-15
Running the Search Process	9-15
Viewing and Replacing Search Results	9-18
Replacing Non-Message Text	9-18
Replacing Message Text	9-22
Running the Replace Process	9-23
Viewing and Undoing Replace Results	9-25
Undoing Data Replacements	9-25
Undoing Text Replacements	9-27
Running the Undo Process	9-28
Reviewing Search Information	9-29
Reviewing Search Criteria	9-29
Viewing Results for Search Only Searches.....	9-31
Viewing Search Results for All Searches	9-31
Understanding Searchable Objects	9-32
Viewing Searchable Objects	9-33
Modifying Searchable Objects	9-34
Upgrade Considerations.....	9-35

Chapter 10

Understanding Time Zones

Types of Time Zones	10-1
Base Time Zone	10-1
User Time Zone.....	10-2
Specified Time Zone	10-2
Setting the Base Time Zone.....	10-2
Controlling Time Zone Display.....	10-3
Setting Time Zone Options in Record Field Properties	10-4
Setting Time Zone Options in Page Field Properties	10-6
PeopleCode Functions.....	10-7
Maintaining Time Zones.....	10-7
Viewing Time Zone Definitions	10-8
Viewing Daylight Savings Start and End Dates.....	10-10

Chapter 11

Controlling Currency Display Format

Maintaining Currency-Specific Settings.....	11-1
Setting Up a Currency Amount Field	11-2
Setting Currency Field Display Properties	11-4
Using System-Wide Multi-Currency Settings	11-5
Resizing Currency Fields Using the International Field Size Report	11-7

Chapter 12

Market Rates and Currency Conversion

Market Rates Functional Requirements.....	12-1
Market Rates	12-1
Quotation Method	12-2
Avoiding Calculated Reciprocal Rates	12-2
Quotation Units	12-2
Triangulation	12-3
Tolerance Checking.....	12-3
Understanding the PeopleSoft Approach to Market Rates and Currency Conversion ..	12-3
RATE_MULT, RATE_DIV, and the Visual Rate	12-4
Triangulated Rates	12-5
Market Rate Indexes	12-6
Market Rate Data	12-7
Interim Exchange Rate View	12-8
Market Rate Default View	12-9
Market Rate Type.....	12-9
Interim View of Market Rate Types	12-9
Currency Quotation Method.....	12-10
Market Rate Definition.....	12-10
Market Rate Definition Default View	12-11
Common Pages for Market Rate Setup and Maintenance	12-11
Currency Code Page.....	12-11
Market Rate Index Page	12-13
Market Rates Page.....	12-14
Rate Definition Page	12-16
Rate Detail Page	12-16
Currency Quotation Method Page.....	12-19
Automatic Reciprocation of Quote Methods	12-21
Market Rate Definition Page.....	12-22
Rate Type Page	12-24
PeopleSoft Common Objects for Currency Conversion.....	12-25

Exchange Rate Work Record	12-25
Rate Detail Secondary Page	12-25
PeopleCode Built-in Functions	12-26
ConvertCurrency	12-26
RoundCurrency	12-27
Common PeopleCode Library Functions	12-27
GetQuoteMethod	12-28
GetMaxVariance	12-29
DetermineVisualRate	12-29
SetVisualRate	12-30
GetExchangeRate	12-30
RetrieveExchangeRate	12-31
CheckRateVariance	12-32
ProcessVisualRateChange	12-34
InitializeDetailPanel	12-34
UpdateRates	12-36
Application-Specific Requirements for Currency Conversion	12-36
Sample Code	12-37
RowInit on the Visual Rate	12-37
FieldEdit on the Visual Rate	12-38
FieldChange on the Visual Rate	12-39
Function to Process the Detail Page	12-39
RowInit on the Foreign Currency Rate	12-40
FieldChange on the Foreign Currency	12-41
FieldChange on the Rate Type or Effective Date	12-42
Formulas	12-43
Formula 1: Determining Primary Visual Rate	12-43
Formula 2: Tolerance Check on Visual Rate Override	12-43
Formula 3: Determining RATE_MULT and RATE_DIV After Visual Rate Entry or Override	12-44
Formula 4: Determining Quote Method Description for Pop-up Page	12-45
Formula 5: Determining Historic Quote Method Description for Pop-up Page....	12-45
Formula 6: Determining Visual Rates for Pop-up Page	12-46
Formula 7: Calculation of Triangulation Rates	12-46
Formula 8: Calculation of Cross Rates	12-47
Formula 9: Calculation of Reciprocal Rates	12-49

Chapter 13

Character Sets and Language Input/Output

Understanding Character Sets	13-1
------------------------------------	------

The Unicode Standard.....	13-3
Unicode Encodings	13-3
Non-Unicode Character Sets.....	13-6
Single-Byte Character Sets	13-6
Double-Byte Character Sets.....	13-7
Character Sets in the PeopleSoft Architecture	13-8
Database Character Sets.....	13-12
Application Server Character Sets	13-13
Client Workstation Character Sets and Language Management.....	13-13
Character Sets and Fonts in PeopleSoft Internet Architecture	13-14
Fonts and the Windows Client	13-15
Input Methods	13-17
Data Sorting and Collation	13-18
Field Length Validation.....	13-18
Application Designer Field Length Semantics.....	13-18
Field Length Checking for Non-Unicode Databases	13-19
International Character Handling in PeopleTools.....	13-22
Standard Name Formats for Chinese, Japanese, and Korean Characters.....	13-22
Detecting and Converting Characters	13-22
Common Char_Code Values.....	13-23
CharType.....	13-23
ContainsCharType.....	13-24
ContainsOnlyCharType.....	13-25
ConvertChar	13-26
Returns	13-27
Parameters	13-27
Supported Conversions	13-28

Chapter 14

COBOL Globalization

PeopleSoft and Unicode	14-1
COBOL and Unicode	14-2
Expanded Storage Space Requirements	14-2
Special Logic for Parsing Unicode Strings	14-3
Unicode-Specific Error Messages	14-3
The COBOL Conversion Utility.....	14-3
Running the Conversion Utility	14-4
Identifying Converted COBOL Programs.....	14-5
Understanding What Is Expanded.....	14-5
Exception 1: Certain SQL Buffer Setup Data	14-6

Exception 2: Certain Redefined Character Fields	14-9
Exception 3: Certain Fields That Appear to Be Dates	14-10
Exception 4: Arrays Composed of a Single Character	14-12
Using Utility Directives	14-13
Sample Uses for Utility Directives	14-13
Understanding Error Logs	14-15
Manual Fine-Tuning of COBOL Programs	14-17
Identifying Unicode and ANSI Data	14-17
Specifying Column Lengths in Dynamic SQL	14-18
Single Character Arrays	14-20
Requesting the Length of a Single Character	14-21
Requesting a Map of an Entire Character String	14-26

Chapter 15

Global Reporting and Analysis

Using the Strings Table	15-1
Evaluation of PS/nVision String Variables	15-2
Using the Strings Table	15-3
PS/nVision Layout Translation	15-7
Design-Time Features	15-8
Column Headings	15-8
Tree Names	15-8
String Variables	15-8
PS/nVision Runtime Globalization Features	15-10
Path Search	15-10
Request Language	15-10
Labels	15-12
PS/nVision Variables	15-12
International Versions of Microsoft Excel	15-12
SQR Globalization	15-12
SQR Printing for A4 Paper	15-12
Currency Precision	15-12
Date and Time Formatting	15-13
SQR Report Translation	15-14
Init_Report_Translation	15-14
Get_Field_Information	15-14
Append_Report_Translation	15-15
Add_Report_Translation	15-15
Sample Strings Table Enabled SQR	15-15
Required Settings for SQR and Unicode	15-17

UseUnicodeInternal.....	15-17
AutoDetectUnicodeFiles	15-17
Additional Options for SQR and Unicode	15-20
String Length.....	15-20
OPEN command.....	15-21
Language-Sensitive Queries	15-22
PeopleSoft Cube Manager	15-23
Cognos PowerPlay Local-Language Cubes	15-23
Hyperion Essbase Multiple-Language Cubes	15-25
Crystal Reports	15-27
Language Parameter for PSCRRUN.EXE	15-28

Index

ABOUT THIS PEOPLEBOOK

This book describes the features of PeopleTools related to the development and implementation of global applications. These features include support for multiple languages, regional formatting of numbers, dates, and times, as well as the use of international currencies and market rates.

This book provides information on the role of PeopleTools in the globalization of PeopleSoft applications. For information specific to your application, please refer to your PeopleSoft application documentation.

This book is written for technical users, project leaders, and programmers who develop, customize, implement, or support global PeopleSoft applications. To take full advantage of the information in this book, we recommend that you first have an understanding of the component objects that comprise PeopleSoft applications, as well as the tools related to your globalization task. For example, to translate an application's panels and menus, you should understand how to create panels and menus using Application Designer.

Overview of PeopleSoft Globalization presents a bird's-eye view of PeopleTools globalization features related to globalization, and provides cross references to more detailed information.

Controlling International Preferences describes the settings used to control a user's international language and formatting preferences.

Global Configuration Considerations explains how to set your database's base language, load translated data into the database, and create new languages if you plan to translate into a language not already defined by PeopleSoft.

Understanding Related Language Tables provides information about the special tables used to store language-sensitive object descriptions and data in PeopleSoft application databases; and how the system accesses information from these tables based on the user's preferred language.

Working With Language-Sensitive Application Data describes how to access and edit foreign-language application data stored in related language tables, and how the system maintains database tables when users make changes to this data.

Translating With Design Tools covers the translation of objects defined in PeopleTools into foreign languages using design tools, including Application Designer, Tree Manager, and Process Scheduler.

Using the Translate Utilities describes how to use PeopleTools utilities to translate system object descriptions, translate value descriptions, and system messages.

Translating Windows Resources covers the translation of the Windows resources used by PeopleTools at runtime, which include strings, dialogs, icons, cursors, and bitmaps.

Modifying Terminology documents how to use the terminology search and replace tool to update terms across your entire database.

Understanding Time Zones explains how PeopleSoft stores and displays times across different time zones.

Controlling Currency Display Format covers definitional settings that control the formatting and display of foreign currencies.

Market Rates and Currency Conversion explains the PeopleSoft features that support currency conversions.

Character Sets and Language Input/Output describes the function of character sets on PeopleSoft client workstations, application servers, and database servers. This chapter also discusses PeopleSoft support for Unicode.

COBOL Globalization discusses what it means to run COBOL in a Unicode environment.

Global Reporting and Analysis describes the globalization features of PeopleSoft reporting and analysis tools, as well as third-party reporting and analysis tools that are used in conjunction with PeopleTools.

Before You Begin

To benefit fully from the information covered in this book, you need to have a basic understanding of how to use PeopleSoft applications. We recommend that you complete at least one PeopleSoft introductory training course.

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

Related Documentation

To add to your knowledge of PeopleSoft applications and tools, you may want to refer to the documentation of the specific PeopleSoft applications your company uses. You can access additional documentation for this release from PeopleSoft Customer Connection (www.peoplesoft.com). We post updates and other items on Customer Connection, as well. In addition, documentation for this release is available on CD-ROM and in hard copy.



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

Documentation on the Internet

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

You'll also find updates to the documentation for this and previous releases on Customer Connection. Through the Documentation section of Customer Connection, you can download files to add to your PeopleBook library. You'll find a variety of useful and timely materials, including updates to the full PeopleSoft documentation delivered on your PeopleBooks CD.

Documentation on CD-ROM

Complete documentation for this PeopleTools release is provided in HTML format on the PeopleTools PeopleBooks CD-ROM. The documentation for the PeopleSoft applications you have purchased appears on a separate PeopleBooks CD for the product line.

Hardcopy Documentation

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

We make printed documentation for each major release available shortly after the software is first shipped. Customers and partners can order printed PeopleSoft documentation using any of the following methods:

Internet

From the main PeopleSoft Internet site, go to the Documentation section of Customer Connection. You can find order information under the Ordering PeopleBooks topic. Use a Customer Connection ID, credit card, or purchase order to place your order.

PeopleSoft Internet site: <http://www.peoplesoft.com/>.

Telephone

Contact Consolidated Publishing Incorporated (CPI) at 800 888 3559.

Email

Email CPI at callcenter@conpub.com.

Typographical Conventions and Visual Cues

To help you locate and interpret information, we use a number of standard conventions in our online documentation.

Please take a moment to review the following typographical cues:

`monospace font`

Indicates PeopleCode.

Bold

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

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

Italics

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

We also use italics when we refer to words as words or letters as letters, as in the following: Enter the number *0*, not the letter *O*.

KEY+KEY

Indicates a key combination action. For example, a plus sign (+) between keys means that you must hold down the first key while you press the second key. For ALT+W, hold down the ALT key while you press W.

Jump Links

Indicates a jump (also called a link, hyperlink, or hypertext link). Click a jump to move to the jump destination or referenced section.

Cross-references

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

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

- Topic list

Contains jump links to all the topics in the section. Note that these correspond to the heading levels you'll find in the Contents window.



Name of Page or
Dialog Box

Opens a pop-up window that contains the named page or dialog box. Click the icon to display the image. Some screen shots may also appear inline (directly in the text).



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



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



Text within this bar outlined in red indicates a crucial configuration consideration. Pay very close attention to these warning messages.

Comments and Suggestions

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

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

Or send comments by email to the authors of the PeopleSoft documentation at:

DOC@PEOPLESOFT.COM

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

CHAPTER 1

Overview of PeopleSoft Globalization

PeopleSoft applications are fully equipped for implementation in global enterprises. The applications are designed so that a single implementation can serve users in different regions: users who share the same data while operating in different languages and applying different date/time and numeric formatting conventions.

Globalization Features

This topic provides an overview of the PeopleSoft globalization system. The topics covered are summaries of material covered in greater detail elsewhere in this document.

International Language Architecture

PeopleTools is designed to incorporate multiple translations of virtually all elements displayed to the user. The following topics summarize the PeopleTools international language architecture.

International Preference Settings

Users can control the language in which translated elements are displayed, as well as regional date/time and number formatting.

The international preference settings control the language in which users view and use the PeopleSoft system, as well region-specific formatting conventions for dates, times, and numbers.

International preferences are really a combination of settings. How you define your preferences depends on whether you access PeopleSoft via the PeopleSoft Internet Architecture or through the Windows client.



For more information about international preferences, see Controlling International Preferences.

Related Language Tables

The *base language* of an application is the application's primary language—normally the language that is used most commonly throughout the enterprise. A database can have only one

base language. All other languages translations stored in the database are referred to as *non-base languages*.

Base-language data is stored on an application's common, or base, database tables. Non-base language data is stored in special tables called *related language tables* or, sometimes, in the Translate table (XLATTABLE). In addition to the language-sensitive data fields, related-languages tables share all key fields with the base table, plus an additional language code field that designates the language into which description fields are translated.

At runtime the system uses the language preference settings to determine how to display language-sensitive elements. If the language setting is a non-base language, the system looks for a translation first, before displaying language-sensitive elements. If no translation is found, the system displays the base language.



For more information about related-language tables, see Understanding Related Language Tables.

Global Configuration Considerations

To properly manage a database with multiple languages, you need to consider certain configuration issues such as setting the base language, loading translated data into your system, and, if you plan to translate into a language that is not already defined by PeopleSoft, enabling new languages.



For more information about global configuration issues, see Global Configuration Considerations.

Translation Features

Specific application data tables and virtually all PeopleTools objects have description, or label, fields that can be translated into non-base languages: these fields are called *language-sensitive fields*.

In almost all cases, the system does not store a redundant definition for each translation. The only parts of the definition that are usually translated are the language-sensitive fields themselves. The translations of these fields are stored in related-language tables in rows that are keyed to a parent row in the base table.

Pages are the exception to this rule. Because the lengths of strings vary from language to language, you generally need to rearrange page controls to accommodate translated text. Pages may also contain language-specific or market-specific elements. Therefore, pages do not have related language tables; instead, each page is stored separately in each language and has a language code as part of its definition.

The following topics summarize the translation tools that you can use to translate the PeopleSoft system.

Translating Application Data

To access language-sensitive data, you can simply log on to the database, using the language in which you want to work. Remember, if a translation does not yet exist, you will see the base language data, even if you logged on using a non-base language. But when you enter data in this mode, the data is saved directly to the related language table.

This situation makes it easy to translate application data. Simply sign on using the target language. Go to the page with the data, where you will see the base language data. Overwrite the base language data with the translated data, and when you save the page, the translation is saved in the related language table.

The Multilingual Support (MLS) feature enables multilingual users (such as French Canadian users) to maintain language-sensitive data by switching from one language to another dynamically, while working on a page. This means that they can conveniently maintain related language tables without closing the page and changing the signon language setting.



For more information about translating application data, see [Working With Language-Sensitive Application Data](#).

Translating PeopleTools Objects

You can translate PeopleTools objects with the same tools you use to build them: PeopleSoft Application Designer, PeopleSoft Tree Manager, and PeopleSoft Process Scheduler. Some objects, such as pages, images, HTML areas, and icon labels in workflow maps, can be translated only with the design tools.

For many objects, however, a second option exists: translate utilities. The translate utilities are a particularly powerful option for translating fields, xlat values, and hard-coded text on pages (that is, text that is not derived from the field description).

The translate utilities can also be useful for translating navigational objects such as menus, menu items, and folder tabs. For these objects, the translate utilities are generally the more convenient alternative if you are translating the entire system at once, or if the translator is a linguist with limited experience using PeopleTools. System messages are typically translated only with the translate utilities. For a quick translation, or for a more technical translator, it's more a matter of personal preference. A translator who has good technical skills and who is familiar with PeopleTools may prefer to translate using the design tools.



For more information about translating PeopleTools objects, see [Translating With Design Tools and Using the Translate Utilities](#).

Translating Windows Resources

System resources consist of dialog boxes, messages, and strings, as well as graphic elements that are contained in the compiled PeopleTools executables. PeopleSoft provides *alternate language DLLs* that contain translations of these strings into several languages. Typically, you do not need to modify these resources, but PeopleSoft provides utilities that allow you to modify these translations or perform your own translations into additional languages.



For more information about translating resources see Translating Windows Resources.

Translating Terms

A special terminology management tool enables you to streamline certain language changes through the efficient use of text search and replace processing. This is particularly relevant to the task of adapting a language that is based on another language. For example, you can implement Mexican Spanish by adapting delivered Spanish translations, using Search and Replace to substitute Mexican Spanish terms where appropriate.

Even if you're not creating an entirely new language translation, you can use this functionality to adapt terminology according to your organizational standards. Does your organization use the words *Cost Center* instead of *Department*? You now have the ability to efficiently make this substitution throughout the database.



For more information about translating resources see Modifying Terminology.

International Formatting

International date, time, and number formats are controlled by a user's language preferences: in the PeopleSoft Internet Architecture, browser settings determine the preferred format; in the Windows client, the Windows Regional Settings determine the preferred format.

In addition to these system-wide formatting preferences, PeopleSoft provides other features to ensure the proper display of times, currencies, and characters.

Understanding Time Zones

PeopleSoft databases always store times based on a system-wide base time zone. That is, if the base time zone is Universal Coordinated Time (also known as Greenwich Mean Time), a time entered as 10 a.m. U.S. Eastern standard time is stored as 6 p.m. because of the time difference between the two locations.

Although the time is stored as 6 a.m., it can still be displayed as 10 a.m. Eastern standard time. In fact, it could be displayed in any time zone you choose. You can even set up the system to enable users to choose the time zone for specific time or datetime displays.

To support display in alternate time zones, PeopleSoft maintains a list of time zones across the globe. This list includes information about daylight savings time observances.

This approach ensures chronological consistency throughout an organization's geographically dispersed locations, while allowing for flexibility in how users see times.



For more information about time zones, see [Understanding Time Zones](#).

Controlling Currency Display

PeopleSoft lets you control currency display formats, including currency symbols, decimal precision, and thousands separators. When user's enter currency data, you can validate the amount against the currency's defined decimal precision.

If you process currencies like the Italian lira, which require fields that are longer than those provided in your standard application, you can use the Set International Field Sizes (TLSINST1) SQR report to increase the length and number of decimal positions of the fields throughout your database.



For more information about currency display, see [Controlling Currency Display Format](#). For more information about other currency features, see [Market Rates and Currency Conversion](#).

Working With Character Sets

Three types of character sets are supported by PeopleSoft:

- Unicode supports all characters from all major languages.
- ANSI Single Byte character sets support a limited number of languages such as those found in the Latin alphabet.
- ANSI Double Byte character sets support complex written languages such as Japanese, Chinese, and Korean.

ANSI databases can support single languages or specific combinations of languages that use the same characters (such as English and German). Unicode provides the ability to mix dissimilar languages in the same database.

Using Unicode, you can maintain a single database with data in any combination of languages you choose. A single PeopleSoft application server can serve multiple users that connect to the mixed-language database, regardless of the language or character set of those users' client machines. The only restriction on a user's ability to access mixed-language data is the capability of the user's client workstation to interpret, display, and accept keyboard entry of the characters from the various languages.

Some character sets require that the client workstation be installed with appropriate fonts or input methods in order to view and enter characters in that language.



For more information about character sets, see *Character Sets and Language Input/Output*. For more information about using COBOL in a Unicode environment, see *COBOL Globalization*.

Global Reporting

The PeopleSoft reporting tools are language-sensitive, enabling you to produce reports in the languages that your users prefer.

PS/nVision and SQR report layouts can both retrieve information from a strings table where you maintain multiple translations of the text strings that appear on your report. This functionality helps you avoid hard-coding labels in the report file itself, enabling translators to translate the report layout in the database without having to edit the report code. Additionally, PS/nVision and SQR allow each page of a report to be printed in a different language, facilitating distribution of multiple-recipient reports.

SQR Unicode support further enables you to produce reports using any characters that are supported by your printer.

PeopleSoft Query supports language-sensitive query output. If you create a query on a table that has a related language record, or if your query includes such a table, PeopleTools performs the language lookup on the related language record. This means that the output of the query will appear in the preferred language as long as translations exist in the related language tables.

In PeopleSoft Cube Manager, you can build cubes that are displayed in local languages for both Cognos PowerPlay and Hyperion Essbase. The local language cubes do not have to be translated; they work by accessing translations, stored in related language tables, of language-sensitive query output and tree nodes that correspond to the cube's members.

PeopleTools also supports multiple language editions of the Crystal Reports Designer program and produces translated output from Crystal.



For more information about reports, see *Global Reporting and Analysis*.

Planning a Globalization Strategy

When you implement a global PeopleSoft application, there are a number of factors related to globalization that can affect system performance and the efficiency of the implementation effort.

Choosing a Base Language

Each PeopleSoft database can have only one base language. Typically, the base language of your database should match the language that is most commonly used by your organization. This is important because it affects the performance of PeopleSoft applications.

When PeopleTools attempts to open language-sensitive objects (such as pages, record descriptions, and menus), it first compares the operator's preferred language to the base language of the database. If the preferred language matches the base language, PeopleTools immediately loads the required definition from the base language PeopleTools tables. However, if the user's preferred language differs from the database's base language, PeopleTools must first query the related language tables for the object. If a translation of the object cannot be found in the operator's preferred language, then a query is performed on the base language tables.

While these queries typically occur very quickly, they still take up valuable processing time. To optimize performance, you should set the base language of your database to the language that is used most often by your users.



For more information about changing the base language, see [Swapping the Base Language](#).

Selective Translation

If you add custom objects or data tables to a PeopleSoft application, you may want to translate the objects and data fields for display in multiple languages. PeopleTools is designed to permit selective translation of the elements in an application database, depending on an organization's requirements and the time and resources it devotes to a translation project.

Because the system displays base-language versions of any language-sensitive elements for which no translation is provided, you can choose to translate only the pages, menus, messages, and so forth, that are required by the operators who are working in non-base languages.

Just because an object is not translated into a user's preferred language doesn't mean that the user can't access that object—the user simply sees the object in the system's base language.

Planning for Globalization Dependencies

You can make your globalization effort more efficient by being aware of certain dependencies while you are designing application objects (especially pages) and during the translation effort. The following suggestions may help:

- Avoid the use of text in page control labels and other labels that can be derived from field names (such as Query heading names and string definitions in the Strings table).
- Design pages so that they have enough room to accommodate object labels that become longer when translated.

- Try to complete the layout of your pages early in the development cycle. Because the entire page layout is stored individually for each language, any changes you make to the page layout, such as adding or deleting fields, have to be reapplied to each language version of the page.
- Choose the appropriate translation method. You can translate PeopleTools objects using a combination of design tools and translation utilities.
- Translate fields first. Translated field names appear throughout the system in page control labels, search dialog field labels, record field names, strings table definitions, query heading labels, and so forth.

Other International Considerations

This topic directs you to other sources of information related to PeopleSoft globalization.

Language Considerations for Upgrade

When using the upgrade functionality in PeopleSoft Application Designer, you have options to help you compare objects across databases. These comparisons are language-sensitive.



For more information about using language settings when you are using PeopleSoft Application Designer upgrade tools, see *Upgrading with Application Designer*.

Global Documentation

PeopleSoft delivers translated versions of its PeopleBooks for PeopleSoft applications. If an international version of PeopleBooks is installed, the user can access the translated documentation.

If a user accesses a document from a PeopleSoft application by clicking Help in a PeopleSoft page or by pressing F1 from the Windows client, the system opens the version of the document that corresponds to the database's current language preference. If the system doesn't find a version of the documentation in the appropriate language, it brings up the English documentation.

PeopleSoft provides a Documentation Customization Kit with information on customizing the documentation. This kit includes all information necessary for setting up F1 help and for customizing the documentation in any language. You can download the kit from Customer Connection.



For more information about translating documentation, download the Documentation Customization Kit from Customer Connection.

CHAPTER 2

Controlling International Preferences

The international preference settings control the language in which users view and use the PeopleSoft system. These settings also control region-specific formatting conventions for dates, times, and numbers.

This section describes how to set and change international preferences.

Understanding International Preferences

International preferences are really a combination of settings. How you define your preferences depends on whether you access PeopleSoft using the PeopleSoft Internet Architecture or using the Windows client.

Understanding Language Codes

Every PeopleSoft session is associated with a particular language code that is established when you sign on to the system. PeopleSoft language codes contain three characters.

PeopleSoft language codes are mapped to corresponding ISO locale codes in the pstools.properties file on the Web server. This mapping enables the PeopleSoft Internet Architecture to derive certain defaults from the ISO locales that are stored in a user's browser settings.



For more information, see Understanding the Browser Language Preference.

As delivered, PeopleSoft includes the following language codes and ISO locale mappings.

<i>PeopleSoft Language Code</i>	<i>Description</i>	<i>Corresponding ISO Locales</i>
CFR	Canadian French	fr_CA
DUT	Dutch	nl_BE
ENG	US English	en_US,en_GB
ESP	Spanish	es_ES,es_AR,es_CO,es_MX
FRA	French	fr_FR,fr_BE,fr_CH

PeopleSoft Language Code	Description	Corresponding ISO Locales
GER	German	de_DE,de_AT,de_AU,de_CH
GRK	Greek	el_EL
INE	International English	en_GB
ITA	Italian	it_IT,it_CH
JPN	Japanese	ja_JP
KOR	Korean	kr_KR
POR	Portuguese	pt_BR

Note that ISO locale codes have two parts: a language code and a country code. This provides flexibility when languages vary from country to country. For example, the ISO code for French is *fr*. French is spoken in several countries, each with its own two-character country code: France (FR), Belgium (BE), Canada (CA), and Switzerland (CH). The full locale code consists of a lowercase language code, a hyphen, and an uppercase country code. So the code for Canadian French is *fr-CA*.

PeopleSoft Internet Architecture Language Preferences

In the PeopleSoft Internet Architecture, language-related settings come from three places:

Setting	Objects Controlled
Signon page language	Most language-sensitive objects, including PeopleTools objects, messages, and language-sensitive data.
Browser language preference	Date, time, and number formats as well as the default language for the PeopleSoft Internet Architecture signon page.
User Profile language	The language used in SQR reports, COBOL, and workflow email.



For more information, see PeopleSoft Internet Architecture Language Preferences.

Windows Client Language Preferences

In the Windows client, language-related settings come from two places:

Setting	Objects Controlled
Configuration Manager settings	Most language-sensitive objects, including PeopleTools objects, messages, and language-sensitive data.
Windows regional settings	Date, time, and number formats.



For more information, see Windows Client Language Preferences.

PeopleSoft Internet Architecture Language Preferences

In the PeopleSoft Internet Architecture, the *signon language* is the language selected on the signon page. This is the master switch that controls the language in which the user views language-sensitive data and objects in the application.

If the signon language is the base language, then all language-sensitive data, messages, and objects appear in the base language. If the signon language is a non-base language, then any language-sensitive data, messages, and objects *that have been translated* appear to the user in the preferred language; those that have not been translated appear in the base language.



For more information about base and non-base languages, see Related Language Tables.

There are two other settings that control PeopleSoft Internet Architecture settings. Each Web browser has its own language preference list that controls date, time, and number formatting, as well as the default signon language. User Profile settings, defined using the security tools, control the language that is used for SQR reports and workflow email sent to the user by the system.

Controlling the Signon Language

As you might guess from the name, users choose a signon language at the time they sign on to the system. Users can also change their signon language during the PeopleSoft session; however this is typically done only for development or testing purposes. Applications can also change the signon language programmatically, to allow applications to provide their own language selection options.

Choosing a Signon Language

When the signon page initially appears, the language used to display the labels on the page indicates the language that will be used for the session. The default language of the signon page is based on the user's browser language settings.



PeopleSoft signon screen - Spanish

You can accept the default language as the signon language just by signing on to the application..

Alternatively, to choose a different signon language, click the link for any other available **Language**. Clicking the link updates the signon screen, and signing on using the updated screen confirms the new signon language.



PeopleSoft signon screen - English



For more information about the languages that are available on the signon screen, see [Creating New Languages](#).

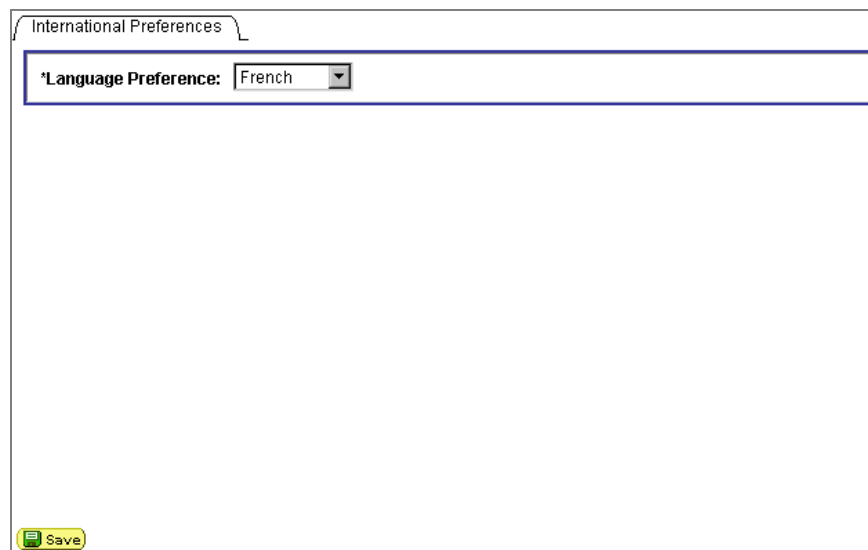
Changing the Signon Language While Signed On

The International Preferences page enables users to change their own signon language mid-session. This is typically done only by developers or system administrators for development, debugging or testing purposes, as not all language settings can be changed mid-session. For example, the language in which the menu navigation is presented can be changed only by signing off and signing back on to the PeopleSoft Internet Architecture.

To change international preferences:

1. Open the International Preferences page.

Choose **PeopleTools, Utilities, Use, International Preferences**. The International Preferences page appears.



International Preferences page

2. Set the Language Preference.

The **Language Preference** control prompts against the translate values for the language code (LANGUAGE_CD) field.

The language you select overrides the language that was used to sign on to the database. This change lasts until the user exits the PeopleSoft session or changes the language preference again.

3. Save the International Preferences page.

Once the International Preference page is saved, changes take effect immediately.

Changing the Signon Language Programmatically

PeopleCode language functions enable you to reset users' language preferences programmatically. The change lasts only for the duration of the PeopleSoft session.

PeopleCode includes the `SetLanguage` built-in function for temporarily overriding the user's default language preference and the `%Language` system variable, which evaluates to the language code for the current preferred language.

Calling the `SetLanguage` function is the same as changing the Language Preference setting on the International Preferences page. In fact, that's how the **Language Preference** control works—by calling the `SetLanguage` PeopleCode function.

The `%Language` system variable makes it simple for application developers to determine the current preferred language and build language-specific functionality in PeopleCode. The `%Language_Base` system variable can be used in PeopleCode to determine the base language of the current database.



For more information about using PeopleCode to set language preferences, see `SetLanguage`, `%Language`, and `%Language_Base`.

Determining the Signon Language When the Signon Screen Is Bypassed

In certain situations, users start a PeopleSoft session without ever seeing the signon screen.

- PeopleSoft offers a direct access option: you can configure the Web server to enable users to access PeopleSoft without having to sign on. Users access the system using a predetermined User ID. The language preference for users who access the system this way is determined by the User Profile language preference for that User ID.

To override the User Profile language preference, you can explicitly set the language code within a direct query string, using the parameter `&languageCd=<target language>`.



For more information about the language preference setting in a User Profile, see [Controlling the User Profile Language Preference](#).

- Users who take advantage of PeopleSoft single-signon capabilities need to select a language preference only once during the single signon. All PeopleSoft sessions that are controlled by the single signon use the same language preference.



For more information about single signon, see [Single Signon](#)

Understanding the Browser Language Preference

Each Web browser allows a user to specify a list of preferred languages for Web content. The preferred language is expressed as an ISO language-country code. For example, *fr* represents French (France) and *fr-CA* represents French (Canada).

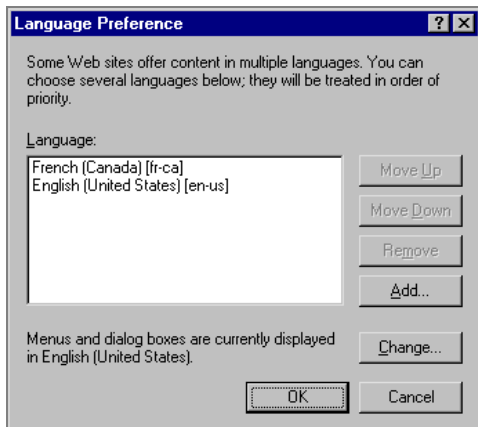
PeopleSoft reads the `pstools.properties` file located on the Web server to derive two pieces of information:

- PeopleSoft maps the ISO local to one of PeopleSoft's three-letter language codes. This code determines the default signon language.
- PeopleSoft maps the ISO local to certain formatting rules. These rules control the display of dates, times, and numbers in PeopleSoft.

Setting Browser Language Preferences

A Web browser's language preferences are maintained by the browser itself. The preferences consist of an ordered list of preferred languages, starting with the most preferred language and descending to the least preferred language. How you maintain this language preference list depends on your Web browser.

For example, in Microsoft Internet Explorer 5, you select **Tools, Internet Options** and then click the Languages button to access the **Language Preference** dialog box. Use this dialog box to list (in order of preference) the languages in which you would like the PeopleSoft Internet Architecture and other Internet sites to appear.



Microsoft Internet Explorer's Language Preference dialog box

In this illustration, French (Canada) is the user's first choice, and English (United States) is the user's second choice.

Applying Browser Preferences to Determine the Default Signon Language

The PeopleSoft Internet Architecture uses the browser's language preference to determine a user's default signon language.

When loading the signon page, the PeopleSoft Internet Architecture checks the language preference of the browser and attempts to find this language in the `pstools.properties` file located on the Web server. The `pstools.properties` file maps PeopleSoft language codes to the ISO locale codes that are used by the browser. As delivered, the mapping is as follows:

```
#psft language code and iso language and country code cross reference

ENG=en_US,en_GB

ESP=es_ES,es_AR,es_CO,es_MX

FRA=fr_FR,fr_BE,fr_CH

ITA=it_IT,it_CH

JPN=ja_JP

DUT=nl_BE

GER=de_DE,de_AT,de_AU,de_CH

POR=pt_BR

INE=en_GB

CFR=fr_CA

GRK=el_EL

KOR=kr_KR
```



If you add languages to your database, be sure to modify the `pstools.properties` file to add a mapping for that language.

If the first-choice language is found, the signon page is displayed in this language. If this language is not found, the PeopleSoft Internet Architecture looks for a translation that corresponds to the language portion of the code.

For example, if a user's first choice is Canadian French (`fr-ca`) and the second choice is U.S. English (`en-US`), PeopleSoft first looks for a Canadian French translation. If that translation does not exist, PeopleSoft checks for a French (`fr`) translation. PeopleSoft checks the next language (`en-US`) only after failing to find languages for both the fully specified locale and the language-only portion of the first choice.

If none of the user's preferred languages are available (as specified in `pstools.properties`), the default signon screen is displayed in U.S. English (`en-US`).

Applying Browser Language Preferences to Determine Formatting Rules

The PeopleSoft Internet Architecture uses the browser's language preference to determine the proper display format for dates, times and numbers from the signon language.

The `pstools.properties` file, located on the Web server, maps the language codes to date, time, and number formatting rules. You can modify the display options if you like. Here are some excerpts from the delivered `pstools.properties` file:

```
locale_en_US=mdy/.,n0
locale_en=mdy/.,n0
locale_en_GB=dmy/.,y0
locale_es=dmy/.,y3
locale_es_AR=dmy/.,y0
locale_es_CO=dmy/.,y3
locale_es_ES=dmy/.,y3
locale_pt=dmy-,.,y0
locale_pt_BR=dmy/.,n0
```

The formatting rules are documented in the file itself. Each option consists of seven characters, representing the following choices:

```
# The value must be 7 characters long and composed as follows:

#   The first 3 are the order of the date.  the only valid values are "mdy",
#   "dmy", and "ymd"

#   The fourth is the date separator character

#   The fifth is the decimal character

#   The sixth is the thousand separator character

#   The seventh is 24 hr time formatting.  Value must be either "y" (for 24-hour
#   clock) or "n" (for 12-hour clock and AM/PM display)

#   The eighth is to define currency formatting

#                               0 = prefix, no separation from number
#                               1  suffix, no separation
#                               2  prefix, 1 char sep
#                               3  suffix, 1 char sep
```

For example, as delivered, the `pt_BR` (Brazilian Portuguese) language code uses the code `dmy/.,n0` to describe its date and time formatting. Parsing this code provides the formatting details:

Dmy Dates are displayed as date, then month, then year.

/	Dates are displayed using a slash to separate the date, month, and year. Dates appear in the format 31/02/2000.
,	Numbers use a comma as the decimal character: the decimal representation of six and one-half is 6,5.
.	Numbers use a period as the thousands separator: the number one thousand appears as 1.000.
N	Time is expressed using a 12-hour AM/PM clock, not a 24-hour clock.
0	The currency symbol is prefixed to monetary amounts; there is no separation between the symbol and the amount.



PeopleTools also includes other currency display controls. For more information about currency formatting, see Controlling Currency Display Format. For more information about pstools.properties, see PeopleSoft Configuration Files.

Controlling the User Profile Language Preference

In the PeopleSoft Internet Architecture, the User Profile language preference determines the language used for SQR report outputs and workflow email sent to the user by PeopleSoft.

To set the User Profile language preference for a user:

1. Access the user's security profile.

Select **PeopleTools, Maintain Security, Use, User Profiles**. Use the standard search method to choose the desired User ID.

The General page appears.

General	ID	Roles	Workflow	Audit	Administrator	Links
User ID: PTDMO Description: Unger,Annette <input type="checkbox"/> Account Locked Out?						
Logon Information						
Symbolic Id:		sa1				
*Password:		*****				
*Confirm Password:		*****				
General Attributes						
Email ID:						
Language Code:		English <input type="checkbox"/> Multi Language Enabled?				
Currency Code:						
<input type="checkbox"/> Enable Expert Entry						
Permission Lists						
Navigator Homepage:		ALLPANLS		Primary:		ALLPANLS
Process Profile:		ALLPANLS		Row Security:		
		Explain				Explain
		Explain				Explain
Save		Return to Search		Add Update/Display		

User Profiles - General page

If you are changing your own language preference, you can, alternatively, use the simplified My Profile page. Select **PeopleTools, Maintain Security, Use, My Profile**. You do not have the opportunity to specify a User ID; you always see your own profile.

General Profile Information	
Unger,Annette	
Password	
Change password Change or set up forgotten password help	
Personalization	
My preferred language for reports and email is: English	
Currency Code:	
Set Personalizations	
Email	
E-mail Address:	
Alternate User	
If you will be temporarily unavailable, you can select an alternate user to receive your routings.	
Alternate User ID:	
From Date:	(example:12/31/2000)
To Date:	(example:12/31/2000)
Miscellaneous User Links	
Save	

My Profile page

2. Choose a default Language Preference.

If you're using the User Profile - General page, choose a language in the **Language Code** field.

If you're using the My Profile page, choose a language in the **My preferred language for reports and email is**.

In either case, the list box displays all the languages that are available in the system so that you can choose an appropriate language.



The **Enable multilingual support** option on the User Profile - General page determines whether the user can edit language-sensitive data in multiple languages without overriding the user language preference. For more information, see *Editing Data in Multiple Languages Using MLS*.

3. Save the page.

Windows Client Language Preferences

In the PeopleSoft Windows client, the Configuration Manager's language setting is the master switch that controls the language in which the user views almost all language-sensitive data and objects in the application.

If the Configuration Manager language is the base language, then all language-sensitive data, messages, and objects appear in the base language. If the signon language is a non-base language, then any language-sensitive data, messages, and objects *that have been translated* appear to the user in the preferred language; those that have not been translated appear in the base language.



This is different from earlier releases, which based most language preferences on the User Profile.



For more information about base and non-base languages, see *Related Language Tables*. For more information about Configuration Manager, see *Configuration Manager*.

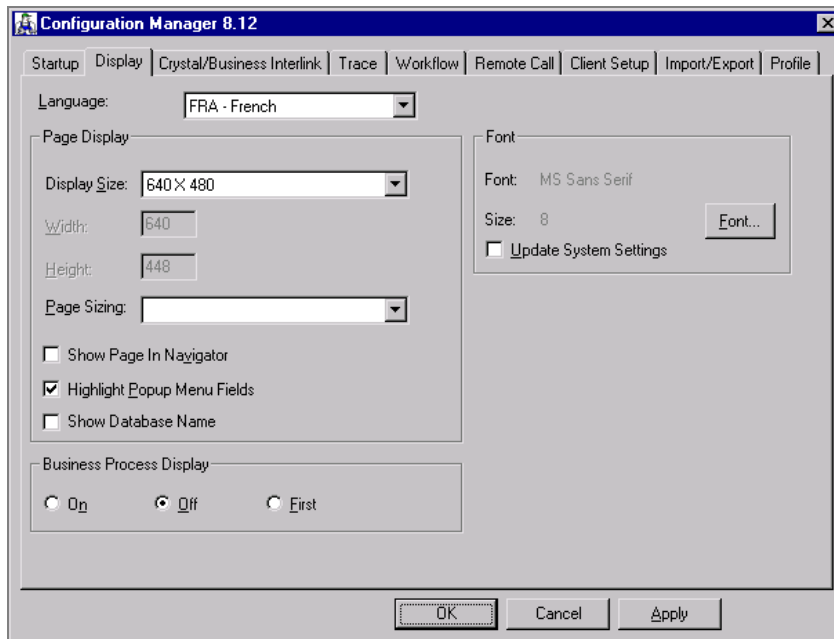
There is one other setting that controls PeopleSoft Windows client language settings: The Windows regional settings control the display of dates, time, and numbers.

Controlling the Configuration Manager Language Setting

The Configuration Manager language setting is set for each client workstation and determines the language in which the user will view and maintain almost all language-sensitive data and objects in the application. The following procedure describes how to change this setting.

To set the Configuration Manager language:

1. Start Configuration Manager.
2. Select the **Display** tab.



Configuration Manager - Display tab

3. Select a Language in the **Language** field.
4. Click **OK**.

Controlling Windows Regional Settings

The Windows Regional Settings control panel enables the user to set the display of numbers, dates, times, and currencies to comply with usage in a specific locale. Bear in mind that these are Windows settings, not PeopleTools settings. The Windows settings affect the entire Windows system and remain in effect until you change them in the Windows control panel.

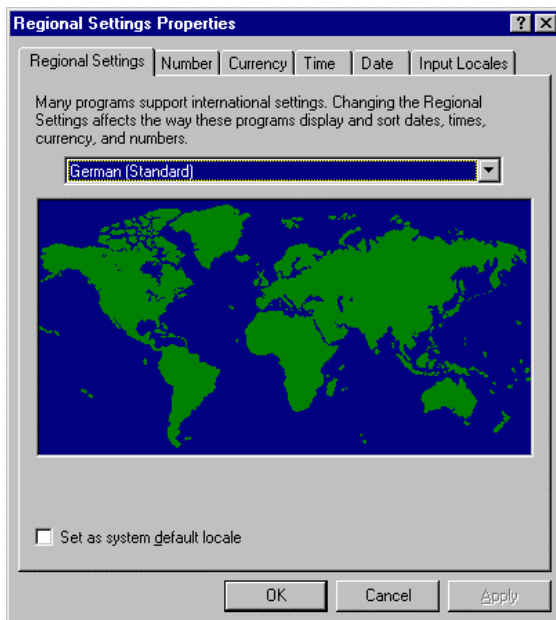


Currency formatting is controlled by PeopleTools and is independent of Windows Regional Settings. For more information about currency formatting, see Controlling Currency Display Format.

You can access Windows Regional Settings through your Windows control panel.

The first tab in the Windows Regional Settings Properties control panel contains a master setting, where you choose a region. Generally this is the only setting that you need to change, because when you choose a new region, the settings on the Number, Currency, Date, and Time tabs

automatically change to comply with normal usage in the region. However, if you want to override any of the settings on those subordinate tabs, you can do so.



Windows Regional Settings Control Panel in Windows NT

The following table describes how some of these settings affect the display of data in PeopleSoft panels.

Tab	Description
Regional Settings	This is the master regional settings control. Choosing a regional setting automatically changes the settings on the Number, Currency, Time, and Date tabs.
Number	<p>Specifies number formatting, including:</p> <p>The Decimal symbol character is used to separate the fractional portion of the number from the whole. Note that if you change this to a comma (,), you should adjust the List separator and Digit Grouping Symbol to a different character, or some applications may not operate correctly.</p> <p>No. of digits after decimal adjusts the number of characters to the right of the decimal point to be displayed. In the record definition, you define the number of integer and decimal positions for number field length. You further define how decimal places will be displayed online in the panel definition.</p> <p>Digit Grouping Symbol specifies the character used as a thousand or other grouping separator.</p> <p>Leading Zero specifies whether numbers less than one contain a leading zero. In the panel definition, use the Auto Fill option to turn on leading zeros for number fields.</p>

Tab	Description
Currency	<p>Most of the settings in the Currency tab are not applied to your PeopleSoft Application.</p> <p>The one setting that is applied by PeopleTools is the placement of the currency symbol. Using the Positive Currency Format option, you can determine if the currency symbol in your Currency Controlled fields is displayed before or after the amount. Note that PeopleTools uses the Positive Currency Format setting for both positive and negative currency values.</p>
Time	<p>Specifies time format defaults. PeopleTools supports the 12/24-hour clock, the separator character, and the AM/PM reference. When you add Time or DateTime fields in a record definition, you indicate the type of formatting that you prefer for the field: HH:MI, HH:MI:SS, or HH:MI:SS.999999.</p>
Date	<p>Specifies date format defaults. PeopleTools supports the Short Date Format specification (MDY, DMY, or YMD), and the Date separator setting. When you add a Date field in a record definition, you indicate whether you want to display the century. PeopleTools Date and DateTime fields always show leading zeros for month, day, and year.</p>

CHAPTER 3

Global Configuration Considerations

To properly manage a database with multiple languages, you need to consider certain configuration issues such as setting the base language, loading translated data into your system, and enabling new languages if you plan to translate into a language that is not already defined by PeopleSoft.

Swapping the Base Language

As delivered, the PeopleSoft system has English data in its base language tables and non-English data in the related language tables. If you want to choose a language other than English as your base language, you need to move all English data to the related language tables (with an appropriate language code), and you need to move all of the data in your new base language to the base tables. This process is called *swapping the base language* of your database. If you later decide to change base languages again, you'll need to go through the same process.

PeopleSoft provides a Data Mover command to automate this process:

```
SWAP_BASE_LANGUAGE <target language>;
```

To swap the base language of your database, you simply need to identify the language to swap in as the new base language; Data Mover finds all the affected tables and loads the appropriate language data into them.

This process handles all related language tables in the system: PeopleTools tables and application tables. Changes are committed after each table is swapped; a log file records the process of the swap and can help you troubleshoot any errors during the process.

Before You Swap the Base Language

Before swapping the base language of your database, you need to have a clean audit report audit report (to ensure that there are no data integrity problems) and you need to have sufficient space and resources to run the swapping process.

Running the Auditing Report

Before swapping the base language of your database, you must ensure that there are no data integrity problems that will cause errors during the swap process. The SQR process SWPAUDIT.SQR performs this validation.

Run SWPAUDIT.SQR using the procedures for your environment as documented in chapter 5, Creating a Database, of the *PeopleTools Installation Guide* for your platform. This SQR does not have a run control, and therefore cannot be run with PeopleSoft Process Scheduler.

SWPAUDIT.SQR may report one or more of the following errors. Refer to the instructions below to correct these errors before continuing with your swap process, and rerun the audit until it reports no errors.



Errors reported during the audit process indicate that the swap base language process will fail unless the errors are corrected.

(SWAP-1) Related Language Records That Are Not Valid Records

The records listed are defined as related language records for one or more base records, but they do not exist in your database.

To fix this problem, do one of the following:

- Create the related language record, as appropriate, based on the keys and translatable fields of the base record.
- Open the base language record in PeopleSoft Application Designer, and remove its association with the related language record in the Object Properties dialog box.



For more information about related language records, see Understanding Related Language Tables.

(SWAP-2) The field LANGUAGE_CD Is Not a Key in the Following Related Language Record(s)

A record is defined as a related language record, but it does not have LANGUAGE_CD as a component of its key. LANGUAGE_CD must be a key field on every related language record.

To fix this problem, make LANGUAGE_CD into a key field on the related language record.

(SWAP-3) The Following Related Language View(s) Have the Wrong Key Structure Defined

The keys of a view used as a related language record by one or more records do not conform to the requirements for PeopleSoft related language records. The related language record must share all of the same keys as its base language record, plus an additional key, LANGUAGE_CD.

To fix this problem, correct the key structure to ensure that it conforms to this requirement and then re-create the view in the database.

(SWAP-4) The Following Related Language Table(s) Have The Wrong Key Structure Defined

The keys of a table used as a related language record by one or more records do not conform to the requirements for PeopleSoft related language records. The related language record must share all of the same keys as its base language record, plus an additional key, LANGUAGE_CD.

To fix this problem, correct the key structure to ensure that it conforms to this requirement, and then alter the table in the database to match the new key structure.

Checking Space and Resources

Before running the SWAP_BASE_LANGUAGE command, be sure that your database has sufficient space and resources to perform the swap process. Although the swap base language process commits after each table is successfully swapped, you need sufficient log space (or rollback space, in the case of Oracle databases) to hold the contents of the largest table to be swapped.

On a newly installed PeopleSoft database, the PSPNLFIELD table is typically the largest table to be swapped, and this can be used as a benchmark for sizing your log space. Plan on at least 50 to 75Mb log or rollback segment space before running the swap process.

Swapping the Base Language

Once you are ready to swap the base language, start Data Mover in non-bootstrap mode by logging in using a regular PeopleSoft User ID (not the access ID). Once Data Mover is started, type and run the SWAP_BASE_LANGUAGE command:

```
SWAP_BASE_LANGUAGE <target language>;
```

Data Mover provides feedback during the swap process, including the name of the record currently being swapped and the number of records that remain to be swapped.

Even when SWPAUDIT completes with no errors, the swap base language process may fail. Typically, any errors during this stage of the process are caused by environmental issues such as lack of database space, log space, or rollback segment space. If a failure occurs, be sure to note the database-specific error message issued, and take the appropriate action, according to your database platform documentation.

Data Mover does not stop when errors are encountered; it continues the swap process. If you want to abort the swap process to fix database-level errors, you need to stop the Data Mover process.

Once you have corrected the problems that caused the failure in the swap base language process, you can restart the process without having to restore your database or remember where the first error occurred. To restart the swap process, simply rerun the SWAP_BASE_LANGUAGE command. Data Mover recognizes the tables whose data has already been swapped and does not attempt to swap the data in those tables again; it will report that zero rows were swapped for those tables.

It is safe to rerun this command as many times as needed, correcting errors between runs, until the log file reports no errors.

In some situations, you may want to swap a specific record. Typically, you would swap a single record only if errors occurred during the swap base language process and you want to verify that the swap will succeed without having to re-swap all the records in the database or trace or troubleshoot the swap process.

To swap a specific table, use the following Data Mover commands:

```
SET BASE_LANGUAGE <target language>;
```

```
SWAP_BASE_LANGUAGE <rename>;
```



You should swap individual tables only when there has been an error with system-wide swapping.



For more information about selecting a base language, see your *PeopleSoft Installation and Administration Guide*. For more information about Data Mover, see Data Mover.

Creating New Languages

PeopleSoft provides translations in nine languages for all end-user objects. However, you can maintain data in your PeopleSoft database in as many languages as required, as long as the characters needed to represent these languages exist in the character set used to create your database. Of course, if you are running a Unicode database, all languages supported by The Unicode Standard can be maintained in a single PeopleSoft database.

As shipped, the PeopleTools package includes definitions for only the nine languages shipped by PeopleSoft. If you plan on using additional languages in your system, you first need to define these languages in PeopleTools.

There are several places where new languages need to be defined before they can be used and recognized by PeopleTools. Work with your system administrator to ensure that you complete all the necessary steps (shown below) before using a new language in PeopleTools.



You do not need to perform these steps for any language where translations have been provided by PeopleSoft.



For more information about how to install PeopleSoft-provided translations, see Loading Translations and the *PeopleSoft Installation and Administration Guide*.

The following tasks must be performed to add a new language code to the system:

1. Determine the PeopleSoft and ISO standard codes for your language.
2. Determine the default non-Unicode character set for your language.
3. Add the new language to the Translate Table (XLATTABLE).
4. Add the new language to the PSLANGUAGES table.
5. Modify the Configuration Manager Windows resources to include the new language in the Configuration Manager's language selection drop-down list.
6. Modify the PeopleSoft Internet Architecture signon page to include a selection for the new language.
7. Add the new language to the PeopleSoft Internet Architecture *pstools.properties* file, providing a mapping between the browser's ISO standard language code and your PeopleSoft language code.

The following topics explain how you'll perform each task. The explanations all follow an example in which we add Swedish as a new language in the PeopleSoft system.

Determining PeopleSoft and ISO Standard Codes for Your Language

PeopleSoft does not currently use the ISO standard language and locale identifiers in PeopleTools. Instead, PeopleTools uses a proprietary three-letter language code that is determined and maintained by PeopleSoft. When adding a new language, you need to choose a three-letter language code to represent the name of your language.

Because this code is used to key several PeopleTools database objects and the PeopleTools cache file, be sure that the language code contains only uppercase U.S. ASCII letters—no numbers or accented characters.

The existing PeopleSoft language codes are as follows:

Code	Language
ENG	English
CFR	Canadian French
DUT	Dutch
ESP	Spanish
FRA	French
GER	German
JPN	Japanese
POR	Portuguese

In our example, we will choose SVE as our new Swedish language code (derived from Svensk, the Swedish name for the Swedish language).

Once you have chosen a three-letter code for your new language, you need to determine the ISO locale that corresponds to that language. ISO locales comprise two components, a language identifier and a territory identifier, separated by an underscore or a hyphen. Typically, the language component is in lowercase letters, and the territory identifier is in uppercase letters. For example, *en_US* is the ISO locale for U.S. English, and *de_CH* is the ISO locale for Swiss German.

The language component of the locale is the two-letter language code defined by the ISO 639 standard. The territory component of the locale is the two-letter country code defined by the ISO 3166 standard.

The territory portion of the ISO locale can be omitted if you want to indicate only a language and not a country-specific derivation of that language. In our example, we plan to add the Swedish language, so the appropriate ISO locale is *sv*. If we were adding a specific dialect of Swedish such as that spoken in Finland, we would use the a specific ISO locale such as *sv_FI*. But as long as we're adding the generic Swedish language, we can use *sv* as the complete ISO locale code.

In most situations, you can use only the language portion of the ISO locale. However, the two-component locale is important if you are planning to add derivations of existing languages such as Mexican Spanish (*es_MX*) or Australian English (*en_AU*). Those language derivations can be easily created from a related language using the PeopleTools Terminology Management tool.



For more information about the PeopleTools Terminology Management tool, see [Modifying Terminology](#).

Determining the Appropriate Non-Unicode Character Set

Although the majority of the PeopleSoft system runs in Unicode, some operations cannot use Unicode characters; for example, file system operations on Unix platforms and integration to some third-party products. For each language in your database, you must determine which non-Unicode or ANSI character set should be used when Unicode operations cannot be performed. The following table lists some of the non-Unicode character sets that are supported by PeopleTools. The complete list is in the PSCHARSETS table in your PeopleTools database. For each new language you are adding, select the appropriate non-Unicode character set from this table.

If you are not using a Unicode PeopleSoft database, the character set you select must be the character set you used to create your PeopleSoft database. For example, if you are using a OS/390 database, you need to specify the EBCDIC CCSID specified in your subsystem's ZPARMS.

Remember that this table is an excerpt from the complete list of character sets that are supported by PeopleTools. For the complete list, query the PSCHARSETS table in your SQL tool or in PeopleSoft Query.

Character Set	Description
ISO-2022-KR	ISO-2022-JP Korean
ISO_8859-1	ISO 8859-1 (Latin1)
ISO_8859-10	ISO 8859-10 (Latin6)
ISO_8859-11	ISO 8859-11 (Thai)
ISO_8859-14	ISO 8859-14 (Latin8)
ISO_8859-15	ISO 8859-15 (Latin9 / Latin0)
ISO_8859-2	ISO 8859-2 (Latin2)
ISO_8859-3	ISO 8859-3 (Latin3)
ISO_8859-4	ISO 8859-4 (Latin4)
ISO_8859-5	ISO 8859-5 (Cyrillic)
ISO_8859-6	ISO 8859-6 (Arabic)
ISO_8859-7	ISO 8859-7 (Greek)
ISO_8859-8	ISO 8859-8 (Hebrew)
ISO_8859-9	ISO 8859-9 (Latin5)
Shift_JIS	Shift-JIS (Japanese)

Adding the New Language to the Translate Table (XLATTABLE)

Once you have determined the PeopleSoft three-letter language code for your new language, you must add it to the Translate table so that it can be recognized by many PeopleTools utilities and PeopleSoft applications.

To add the language code to the Translate table, open the LANGUAGE_CD field in PeopleSoft Application Designer, and select File, Object Properties. Move to the Translate Values tab, and click the **Add** button.

Field Properties

General | International Format Settings | Translate Values

Field Name: LANGUAGE_CD

	Value	Active	Eff Dt	Long Name	Short Name
1	CFR	<input checked="" type="checkbox"/>	01/01/1900	Canadian French	Can French
2	DUT	<input checked="" type="checkbox"/>	01/01/1900	Dutch	Dutch
3	ENG	<input checked="" type="checkbox"/>	01/01/1900	English	English
4	ESP	<input checked="" type="checkbox"/>	01/01/1900	Spanish	Spanish
5	FRA	<input checked="" type="checkbox"/>	01/01/1900	French	French
6	GER	<input checked="" type="checkbox"/>	01/01/1900	German	German
7	INE	<input checked="" type="checkbox"/>	01/01/1900	International English	Intl Eng
8	ITA	<input checked="" type="checkbox"/>	01/01/1900	Italian	Italian
9	JPN	<input checked="" type="checkbox"/>	01/01/1900	Japanese	Japanese
10	POR	<input checked="" type="checkbox"/>	01/01/1900	Portuguese	Portuguese

Add Translate Table Value

Last Updated: 03/21/1998 3:00
 Date/Time: 03/21/1998 3:00
 By User: PPLSOFT

Field Value: SVE
 Effective Date: 01/01/1900 ☐ Inactive
 Long Name: Swedish
 Short Name: Swedish

Translate Values - adding a new language code

Add the new three-letter language code and a description of your new language. Use an effective date of 01/01/1900 to ensure that the language is always accessible to your applications. When adding the **Long Name** and **Short Name** for your new language code, enter the name of the language as it is referred to in the base language of your database. Once you have defined the translate value for your language code, you can translate it into each language.

In our example, the database's base language is English, so we will use *Swedish* as both the Long Name and Short Name of our new language. Once we have finished defining this new language, our Swedish translators may want to translate this new value into Swedish, as *Svensk*.

Adding the New Language to the PSLANGUAGES Table

The PSLANGUGES table is used by many PeopleTools utilities to determine which languages defined in the database are actually in use (rather than being defined but unused). This table also provides a mapping to a default non-Unicode character set that is used to represent that language when a Unicode representation is not possible (for example, during some file system operations).

You must use SQL to manually insert new languages into the PSLANGUGES table. There is no page that provides access to the data in this table, as it is an internal PeopleTools table.

Using Data Mover, log in to your database in non-bootstrap mode. Alternatively, you can use your preferred SQL tool to log in to the database.

Execute the following SQL statement, replacing *SVE* with your three-letter PeopleSoft language code, and replacing *ISO_8859-1* with the default character set that you selected in step 2. The numeral 1 indicates that the language is installed and enabled in your PeopleSoft database.

```
INSERT INTO PSLANGUAGES (LANGUAGE_CD, CHARSET, INSTALLED)
VALUES ('SVE', 'ISO_8859-1', 1)
```

Remember that values in this table are case sensitive. The language code you specify must match the language code that you defined in the Translate table, and the character set you specify must match the character set you selected from the PSCHARSETS table.

Modifying the Configuration Manager Windows Resources

Configuration Manager provides a list of languages that are available to users for their PeopleTools Windows client sessions. This list is embedded in the Windows Resource File for Configuration Manager itself.

To add your new language to the list of languages in Configuration Manager, follow the instructions in Translating Windows Resources to edit the Windows resource files for the PeopleTools module PSCFG.

PSCFG.RC stores the list of languages used by Configuration Manager's Display tab. These languages are stored as strings with the name *IDS_LANG x* , where x is a sequential number. You can edit PSCFG.RC using any text editor. To add your new language, find the next available *IDS_LANG* entry, change the text to match your new language, using the format *LanguageCode - Description*.

Once you have updated the next available *IDS_LANG* entry, update the entry *IDS_NUMLANGUAGES* to specify the new highest *IDS_LANG* entry you used. The example below shows the addition of Swedish as the twelfth language in PSCFG.RC. The boldface text shows the entries that have been changed:

IDS_NUMLANGUAGES	"12"
IDS_LANG1	"CFR - Canadian French"
IDS_LANG2	"DUT - Dutch"
IDS_LANG3	"ENG - US English"
IDS_LANG4	"ESP - Spanish"
IDS_LANG5	"FRA - French"
IDS_LANG6	"GER - German"
IDS_LANG7	"INE - International English"
IDS_LANG8	"JPN - Japanese"
IDS_LANG9	"POR - Portuguese"
IDS_LANG10	"THA - Thai"
IDS_LANG11	"ITA - Italian"
IDS_LANG12	"SVE - Swedish"

```
IDS_LANG13          " (Add new language type here, and update
                     IDS NUMLANGUAGES) "
```

Once you have modified the resource files to add your new language, use the MAKEALTL.BAT batch file to compile the alternate language DLL for Configuration Manager in each language where you added the new entry.



Typically, no alternate language DLLs are required for the English language, as English resources are included in each core PeopleTools DLL. However, in this case, you need to modify the English language resources for Configuration Manager in order to add your new language. This means that you need to compile an English alternate language DLL for Configuration Manager (PSCFGENG.DLL), just as you do for any other language. Use the MAKEALTL.BAT batch file to compile your English resources, specifying ENG as the language code argument.



For more information about MAKEALTL.BAT, see Translating Windows Resources.

Additionally, if you plan on translating your Windows Resources into your new language, you will want to create another directory structure under %PS_HOME%\SRC\RES on your file server to contain the new translations. Simply copy the existing %PS_HOME%\SRC\RES\ENG tree (or use another language, if you prefer not to use English as the source for your translations) to a new directory under %PS_HOME%\SRC\RES, matching the PeopleSoft three-letter code for your new language.

Modifying the PeopleSoft Internet Architecture Signon Page

The PeopleSoft Internet Architecture enables users to select their signon language by selecting the language from the signon page. This page is maintained on your Web server as a static HTML file and is not generated from a PeopleTools page definition. You need to edit this HTML file to add your new language code.

Work with your Web server administrator to edit the file `signin.html` on your PeopleTools Web server. The `signin.html` file contains an HTML table that lists each PeopleTools language. Add a new entry to this table to reference the name of your new language *in the language of the new language*. If the language does not use Latin characters, you may want to reference an image that contains the name of the language, to ensure that all users can correctly see the language name, even if their Web browser settings are not configured for that language.

In our example, we add the line shown in boldface to our `signin.html` file. Note the Japanese entry demonstrating how to reference an image that contains the language name for a non-Latin language:

| | |

[illegible]

Adding the New Language to the pstools.properties File

You need to add an entry for your language needs to the *pstools.properties* file, also located on your Web server. This file provides a mapping between the ISO locale and the three-letter PeopleTools language code that you selected for your language. The mapping is used by the PeopleSoft Internet Architecture to provide the signon screen in the user's preferred language, based on the Web browser's language preference settings.

Work with your Web server administrator to edit the `pstools.properties` file in the same directory as the `signin.html` file that you edited in order to modify the signon page.

The *pstools.properties* file includes a section that maps ISO locales to the equivalent PeopleSoft three-letter language code. Each PeopleSoft language code can be mapped to any number of ISO locales to deal with the case where multiple ISO locales are handled by a single PeopleSoft language. For example, by default, en, en_US, and en_GB are all mapped to the PeopleSoft language code ENG because separate English translations are not provided for these locales.

In our example, we will add mappings to the new SVE Swedish PeopleSoft language code for both the sv (Swedish) ISO locale and the sv_FI (Swedish - Finland) ISO locale. In the following example, **boldface** indicates the new line that was added:

```
#iso language and country code to psft language code cross reference

en=ENG

en_US=ENG

en_GB=ENG

es=ESP

es_ES=ESP

es_AR=ESP

es_CO=ESP
```

es_MX=ESP

fr=FRA

fr_FR=FRA

fr_BE=FRA

fr_CH=FRA

fr_CA=CFR

it=ITA

it_IT=ITA

it_CH=ITA

ja=JPN

ja_JP=JPN

nl=DUT

nl_BE=DUT

de=GER

de_DE=GER

de_AT=GER

de_AU=GER

de_CH=GER

pt=POR

pt_BR=POR

sv=SVE

sv_FI=SVE



For more information about settings in the pstools.properties file, see Understanding the Browser Language Preference.

Loading Translations

This topic describes the steps to follow if you have already installed a PeopleSoft database and you want to add translations that you have licensed from PeopleSoft.

If at all possible, you should install the PeopleSoft Global Multi-Language CD-ROM at the same time that you install the English release. This is an efficient approach because this process involves several steps, some of which may be time-consuming and involve some analysis of your database.

First, you need to determine how you will perform your upgrade. Choose the most appropriate option:

- Add the translations provided on the Global Multi-Language CD-ROM to your existing PeopleSoft database.

This option is available only if you have not made any customizations to existing PeopleTools-based objects in your existing database.

PeopleTools-based objects include any objects that you maintain using Application Designer or Tree Manager such as pages, fields and menus. If you have added only new PeopleTools objects to your database, without modifying existing definitions, then you can use this option.

Adding or customizing system data such as account codes, departments, and locations is not considered a PeopleTools object change. If you have performed only these types of transactions against your existing database, and you have not modified any objects using Application Designer or Tree Manager, then you can use this option.

- Upgrade your customizations to a multilingual database.

If you have made customizations to the PeopleTools objects in your database, you cannot simply add the translations contained on the Global Multi-Language CD-ROM to your existing database. Instead, you must create a new database containing all of your licensed translations and use the upgrade process to copy customizations from your existing database to your newly installed database.

Adding Translations to an Existing Database

This topic explains how to load translations by adding them to an existing database. This method of loading translations not be available to you, depending on the customizations you have performed on your existing database. Be sure to read the overview for this section before loading translations.

To add translations to an existing database:

1. Load your Multi-Language CD-ROM.

Follow the instructions in chapter 2, Setting Up the File Server, of your *PeopleSoft Installation and Administration Guide* to load the Global Multi-Language CD-ROM onto your file server.

2. Rerun the steps for setting up the batch environment.

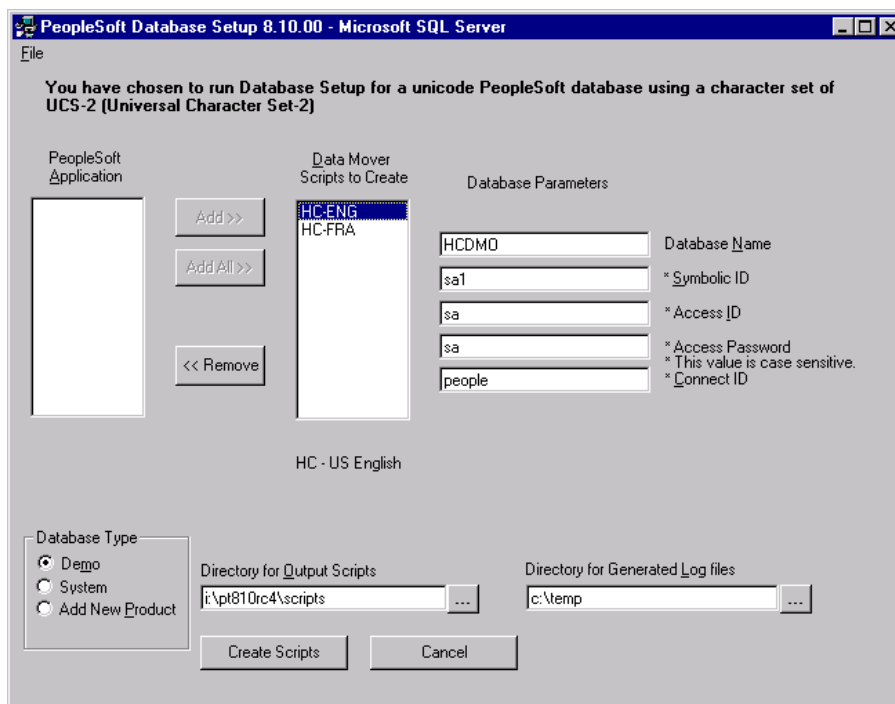
If you are running a Windows NT process scheduler (*PSNT*), then you need to rerun the steps for “Setting Up the Batch Environment” documented in your *PeopleSoft Installation and Administration Guide*.

Because the Global Multi-Language CD-ROM contains translations for your Crystal Reports and PS/nVision layouts, those translations need to be transferred to your Windows NT process scheduler.

If you run only non-Windows NT process schedulers, you do not need to complete this step; this step does not need to be performed for UNIX batch servers because the only objects that don’t contain translations in the database are Crystal and PS/nVision reports, both of which are supported only on Windows NT batch servers.

3. Create Data Mover import scripts.

Follow the instructions in chapter 5, Creating a Database, of your *PeopleSoft Installation and Administration Guide* to create the Data Mover import scripts using the PeopleSoft Database Setup program. You will see both the English database objects and your new translated objects in the **Data Mover Scripts to Create** area of the Database Setup program.



Database setup program

Be sure to enter the same database parameters that you entered the first time you ran the Database Setup program to install your English database.

In the **Data Mover Scripts to Create** list, select only the language that you are now adding to your database. Do not select English or any other language that already exists in your database.

Before clicking the **Create Scripts** button, select a **Directory for Output Scripts** that is different from the directory you specified when you ran this program to install your English-only database. This prevents the newly generated script from overwriting any existing scripts that you may have generated using Database Setup.

4. Run the Data Mover script.

Follow the instructions in the *PeopleSoft Installation and Administration Guide* to run your modified DMS script against your existing PeopleSoft database. Because you have already created your database and set up the appropriate database-specific objects, you need to perform only the task “Run Data Mover Import Scripts.”

Be sure to run Data Mover in bootstrap mode by logging in with your database’s Access ID.

5. Run VERSION.SQR to update your PeopleTools version numbers.

Run the PeopleTools VERSION.SQR against your database, as documented in the *PeopleSoft Installation and Administration Guide*. This will synchronize the PeopleTools version numbers, taking the new translations into account.

6. Swap your base language.

If you have not already swapped your base language, and you decide to run your database in a non-English base language, follow the steps described in *Swapping the Base Language* to change the base language of your database.

7. Rerun your database audits.

Your database should now be complete. To ensure that it is complete, rerun the database audits as documented under the heading “Check Database” in chapter 5, *Creating a Database*, of your *PeopleSoft Installation and Administration Guide*.

Upgrading Your Customizations into a Multilingual Database

This topic explains how to load translations by upgrading your customizations into a multilingual database. Be sure to read the overview for this section before loading translations.

To upgrade your customizations into a multilingual database:

1. Load your Multi-Language CD-ROM.

Follow the instructions in chapter 2, *Setting Up the File Server*, of your *PeopleSoft Installation and Administration Guide* to load the Global Multi-Language CD-ROM onto your file server.

2. Rerun the steps for setting up the batch environment.

If you are running a Windows NT process scheduler (*PSNT*), then you need to rerun the steps for “Setting up the Batch Environment” in the *PeopleSoft Installation and Administration Guide*.

As the Global Multi-Language CD-ROM contains translations for your Crystal Reports and PS/nVision layouts, those translations will need to be transferred to your Windows NT process scheduler.

If you run only non-Windows NT process schedulers, you do not need to complete this step. This step does not need to be performed for UNIX batch servers because the only objects that don't contain translations in the database are Crystal and PS/nVision reports, both of which are supported only on Windows NT batch servers.

3. Create and load your new Multi-Language database.

Follow the steps in chapter 5, Creating a Database, of your *PeopleSoft Installation and Administration Guide* to create and load a new database containing all the languages you have licensed. Keep in mind that you need to load the English objects provided by PeopleSoft, even if you don't plan to run your system in English. Follow all of the steps in Chapter 5 that are applicable to your installation, except the "Swap Base Language" step. You cannot swap the base language of your new database until you have copied your customizations into it.

4. Copy your customizations into your new Multi-Language database.

Using the Upgrade Copy utility in Application Designer, copy the objects you have modified from your existing database into your new Multi-Language database.

If you have maintained all of your customizations in Application Designer projects, you can easily use these projects to copy only the objects you have modified over to your new Multi-Language database.

If you have not maintained your customizations in a project, you may want to use the Application Designer Upgrade Compare process to compare your customized database against a newly installed, English-only database in order to identify your changes. Remember that, for this compare process, your source database is your existing database, and your target database is a newly installed, English-only database.

Once you have identified the objects you have changed, using the Compare process, use Upgrade Copy to copy the objects into your new Multi-Language database. Remember that during the Upgrade Copy process, your source database is your existing English-only database and your target database is the new Multi-Language database.



For more information, see *Upgrading with Application Designer*.

5. Run VERSION.SQR to update your PeopleTools version numbers.

Run the PeopleTools VERSION.SQR against your database. This will synchronize the PeopleTools version numbers, taking the new translations into account.

6. Swap your base language.

If you have not already swapped your base language and you decide to run your database in a non-English base language, swap the base language of your database.



For more information and instructions on how to swap the base language, see *Swapping the Base Language*.

7. Rerun your database audits.

Your database should now be complete. To ensure that it is complete, rerun the database audits, as documented under the heading “Check Database” in chapter 5, *Creating a Database*, of the *PeopleSoft Installation and Administration Guide*.

Understanding Related Language Tables

PeopleTools supports the use of multiple languages primarily through related language tables. A *related language table* is a special table that is used to maintain foreign language translations of description field values that are stored in a *base table*.

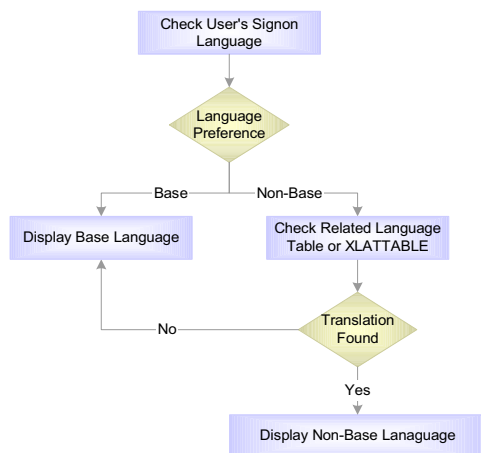
When the system displays a language-sensitive field value, it retrieves the text from either the base table or the related language table, depending on:

- The current language preference.
- Whether any translated rows for the field are found in the related language table.



The language preference refers to either the PeopleSoft Internet Architecture signon language or the Windows client language preference as determined by the Configuration Manager language preference. For more information about language preferences, see Controlling International Preferences.

If the current language preference is the system's base language, the text is retrieved from the base table. If the language preference is a non-base language, then the system looks for a translation of the text in the related language table. If it finds a translation, it displays the translated text; if no translation exists, the system uses the text in the base table. This enables developers to selectively translate portions of the system, while keeping the system fully functional at all times, even if not all rows have been translated. The following flowchart shows this flow of execution:



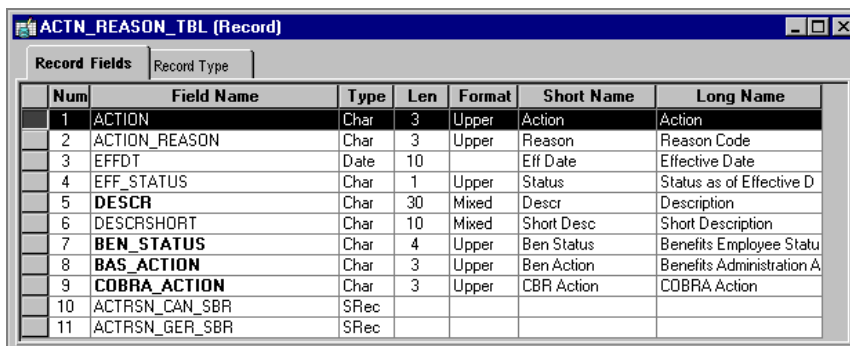
System display of base and non-base language text

The contents of related language tables are maintained by the system. When you translate a language-sensitive field, the system adds a new row to the related language table. When you delete a row from a base table, any child rows in the related language table are deleted. The primary responsibility of the application developer in the language architecture is to define and maintain the related language tables.

Related Language Table Structure

Related language tables store multiple translations of language-sensitive fields from an associated base table. For example description (DESCR) and short description (DESCRSHORT) fields are commonly language-sensitive.

The Action Reason table, ACTN_REASON_TBL, is one example of a base table. Note that it has the language-sensitive DESCR and DESCRSHORT fields. All other fields in the table are translate value or prompt table editable codes and are therefore not language-sensitive. But because this table includes at least one language-sensitive field, it is a language-sensitive table.



Num	Field Name	Type	Len	Format	Short Name	Long Name
1	ACTION	Char	3	Upper	Action	Action
2	ACTION_REASON	Char	3	Upper	Reason	Reason Code
3	EFFDT	Date	10		Eff Date	Effective Date
4	EFF_STATUS	Char	1	Upper	Status	Status as of Effective D
5	DESCR	Char	30	Mixed	Descr	Description
6	DESCRSHORT	Char	10	Mixed	Short Desc	Short Description
7	BEN_STATUS	Char	4	Upper	Ben Status	Benefits Employee Statu
8	BAS_ACTION	Char	3	Upper	Ben Action	Benefits Administration A
9	COBRA_ACTION	Char	3	Upper	CBR Action	COBRA Action
10	ACTRSN_CAN_SBR	SRec				
11	ACTRSN_GER_SBR	SRec				

ACTN_REASON_TBL

The base table must be explicitly associated with its related language table. This association is made on the Use tab of the base table's Record Properties dialog box, as shown here:

The image shows the 'Record Properties' dialog box with the 'Use' tab selected. It contains several sections for configuring record relationships and auditing.

General | **Use**

Set Control Field: [Dropdown]

Record Relationships

Parent Record: [Dropdown]

Related Language Record: ACTN_RSN_LANG [Dropdown]

Query Security Record: [Dropdown]

Record Audit

Record Name: [Dropdown]

Audit Options

- ☐ Add
- ☐ Change
- ☐ Selective
- ☐ Delete

OK Cancel

Record Properties dialog box - Use tab

A related language table must:

- Share all key record fields with its base table.
- Have an additional key record field, which must be LANGUAGE_CD.
- Have language-sensitive fields matching those in the base table (these are typically DESCR and DESCRSHORT or LONGNAME and SHORTNAME).
- Not contain any fields that are not present on the base table, except the LANGUAGE_CD key field
- Not contain any non-key, non-language-sensitive fields from the base table.

The related language table, ACTN_RSN_LANG, meets all of these requirements:

The image shows the 'ACTN_RSN_LANG (Record)' window with the 'Record Fields' tab selected. It displays a table with 7 columns: Num, Field Name, Type, Len, Format, Short Name, and Long Name. The table lists 6 fields.

Num	Field Name	Type	Len	Format	Short Name	Long Name
1	ACTION	Char	3	Upper	Action	Action
2	ACTION_REASON	Char	3	Upper	Reason	Reason Code
3	LANGUAGE_CD	Char	3	Upper	Lang Cd	Language Code
4	EFFDT	Date	10		Eff Date	Effective Date
5	DESCR	Char	30	Mixed	Descr	Description
6	DESCRSHORT	Char	10	Mixed	Short Desc	Short Description

ACTN_RSN_LANG

How Related Language Tables Store Translations

Base tables store language-sensitive descriptions in the database's base language. Related language tables store translations of these descriptions in non-base languages. For each row in the base table, there can be zero rows or one row in the related language table for each non-base language.

For example, the following table shows a row of data from the ACTN_RSN_TBL table.

<i>Action</i>	<i>Action_Reason</i>	<i>Effdt</i>	<i>Descr</i>	<i>DescrLong</i>
PAY	PRO	01/01/98	Promotion	Promotion

The following table shows child rows of data from the ACTN_RSN_LANG table. For each LANGUAGE_CD, there is a row with appropriate translations of the language-sensitive fields in the parent row.

<i>Action</i>	<i>Action_Reason</i>	<i>Effdt</i>	<i>Language_Cd</i>	<i>Descr</i>	<i>DescrLong</i>
PAY	PRO	01/01/98	ESP	Promoción	Promoción
PAY	PRO	01/01/98	FRA	Promotion	Promotion
PAY	PRO	01/01/98	GER	Beförderung	Beförderung

How Related Language Tables Are Used

Related language tables are the primary means of storing language translations in the PeopleSoft system. They store translations of descriptive text for both data (such as a standard list of expense categories) and system objects (such as descriptions of fields and records).

Related language tables that store descriptive text for data are generally associated with edit tables that store lists of values that are used for validation and prompting. When a page field prompts against a table that has an associated related language table, then, if the user's language preference is set to a non-base language, the description fields that have been translated into the current language appear in place of the base language descriptive text.

For example, the following page is displayed with the user's language preference set to Spanish (ESP). The Country field (País) prompts against the Country table (COUNTRY_TBL). The translated description (Francia) comes from the Country table's related language table (COUNTRYTBL_LANG).

Page with translated descriptive text



For more information about translating language-sensitive data fields, see Working With Language-Sensitive Application Data.

Many PeopleTools object definitions, including fields, records, and components, have associated related language tables. This makes maintenance of multiple language PeopleTools objects relatively straightforward: the system uses a single mechanism for maintaining translated data and for maintaining the system objects themselves. Field translation is particularly powerful, because translated field descriptions appear on search pages and on pages where the descriptive text for the field is used as the field label.

For example, all the fields on the ADDRESS2_SBP subpage are translated into Spanish:

PAGE (NONE)				
Target Lang = ESP; Project = ADDRESS_PFDU; Show = All				
Field	Base Long Name	Tgt Long Name	Base Short Name	Tgt Short Name
Label	Base Long Name	Tgt Long Name	Base Short Name	Tgt Short Name
Page Text	Base Label Text	Tgt Label Text		
Xlat Value Xlat Eff Date	Base Long Name	Tgt Long Name	Base Short Name	Tgt Short Name
ADDRESS2_SBP				
ADDRESS1	Address Line 1	Dirección1	Address 1	Dirección1
LN - ADDRESS1				
ADDRESS2	Address Line 2	Dirección2	Address 2	Dirección2
LN - ADDRESS2				
ADDRESS3	Address Line 3	Dirección3	Address 3	Dirección3
LN - ADDRESS3				
CITY	City	Ciudad	City	Ciudad
LN - CITY				
COUNTRY	País	País	País	País
LN - COUNTRY				
VIEW_RESUME				
ZIP	Postal Code	Cd.Postal	Zip	Cd.Postal
LN - ZIP				

Translation of fields in ADDRESS2_SBP

Because the fields are translated, the translated versions of field labels are displayed when a user whose language preference is Spanish accesses a page that shows the long name or short name of any of these fields. No additional development work on the part of the developer is required.

Of course, the translations of the object names must be maintained and updated whenever the base labels change.

Domicilio Particular	
Dirección1:	461 Haven Ct
Dirección2:	
Dirección3:	
Ciudad:	Moraga
País:	USA
Cd.Postal:	94556
Teléfono:	415/376-3848

Detail of page containing automatically translated address information



For more information about translating system objects, see Translating With Design Tools and Using the Translate Utilities.

Creating Related Language Tables

PeopleSoft applications provide related language tables for most edit tables that you are likely to want to translate, so in most cases you will not have to create your own related language tables. However, during a customization, you may create a new table that requires translation. If this happens, you will need to create your own related language table.

The following procedure assumes that you know how to design and build records in PeopleSoft Application Designer.



For more information, see [Creating Record Definitions](#).

To create a related language table:

1. Check the structure of the base table record, noting its key fields. Make sure that it has fields that can be language-sensitive (such as DESCR and DESCRSHORT).

If the base table has no fields that are language-sensitive, then it does not need a related language record.

2. Design the related language record.

The related language record must:

- Have all the same key fields as the base table.
- Have an additional key field, which must be LANGUAGE_CD.
- Have the language-sensitive fields from the base table.
- Not contain any fields that are not present on the base table except the LANGUAGE_CD key field
- Not contain any non-key, non-language-sensitive fields from the base table

By convention, the related language record name should reflect its relationship to the base table and end with *LANG*. For example, if the base table is MY_NEW_TBL, the related language table might be named MY_NEW_LANG.

3. Build the related language table.

An easy way to build the related language table is to copy the base record definition using the File, Save As menu command. Once you save the record with a new name, you can remove the fields that are not required on the related language table and add the LANGUAGE_CD key. By using this approach, you avoid having to remember the key structure and column names of the base table.



When using Save As to create the related language record, you do not need to also save the PeopleCode that is associated with the base record. PeopleCode programs on related language records are not executed; therefore, they are redundant and may be misleading. We recommend that you not maintain PeopleCode programs on related language records for this reason.



For more information about building SQL tables, see Building SQL Tables and Views.

4. Associate the related language table with the base table record.

Open the Record Properties dialog box for the base table, then set the **Related Language Record** property on the Use tab to the related language record.

The screenshot shows the 'Record Properties' dialog box with the 'Use' tab selected. The 'Set Control Field' dropdown is empty. Under 'Record Relationships', the 'Parent Record' dropdown is empty, the 'Related Language Record' dropdown is set to 'MY_NEW_LANG', and the 'Query Security Record' dropdown is empty. Under 'Record Audit', the 'Record Name' dropdown is empty. The 'Audit Options' section has four checkboxes: 'Add' (unchecked), 'Change' (unchecked), 'Selective' (unchecked), and 'Delete' (unchecked). The 'OK' and 'Cancel' buttons are at the bottom right.

Setting the related language record property of the base table record

Click **OK** to close the Record Properties dialog box.

5. Save the record.

Creating Related Language Views

Just as records with language-sensitive fields require related language records, views with language-sensitive fields require related language views. This is necessary because the view does not recognize when one or more tables it selects from has a related language table.

Related language views work the same way as related language records:

- You can create a related language view for any view that has language-sensitive fields. Anytime you create a view over a table that has a related language record, you will typically also need a related language view.
- The related language view consists of all the key fields from the base view, a language code, and the language-sensitive fields.
- You associate the base view and the related language view in the Record Properties dialog box of the base view.
- When a user logs on in a non-base language, PeopleSoft retrieves the data with the appropriate language code from the related language view.
- It's best to use the same naming convention for related language views as you do for related language records: append a *LANG* suffix to the view name.

Related language views have one additional issue: The select statement from the original view must be modified to select any language-sensitive fields from the appropriate related language tables.

The select statements behind the views vary in complexity, depending on how many tables are involved and how many of those tables have related language records.

The following topics provide examples of how you can construct the select statements for language-sensitive views.

One Base Table, One Related Language Table

When you have a view that selects data from a single table, the select statement for the language-sensitive view is straightforward: Simply select the data from the related language table, making sure to also select the `LANGUAGE_CD`.

For example, consider a view on the `CORP_SHIRT_TBL`, a table keyed by `SHIRT` and with one language-sensitive field, `DESCR`:

CORP_SHIRT_TBL	
SHIRT	DESCR
PSJ	PS-Japan
EUR98	European C...
PS007	PeopleSo...

CORP_SHIRT_LANG		
SHIRT	LANG_CD	DESCR
PSJ	FRA	PS-Japon...
EURO98	ESP	Conferenc...
EURO98	GER	PeopleSo...

One base table, one related language table

Let's assume you need to create a view that selects only the rows where the SHIRT key field begins with the letter *P*.

Here's the select statement for the base view:

```
SELECT SHIRT, DESCR FROM PS_CORP_SHIRT_TBL WHERE SHIRT LIKE 'P%'
```

Here is the select statement for the language-sensitive view:

```
SELECT SHIRT, DESCR, LANGUAGE_CD FROM PS_CORP_SHIRT_LANG WHERE SHIRT LIKE 'P%'
```

Because this view is very simple and selects only columns that exist both in the base table and in the related language table, the related language view is straightforward. It differs from the base view in only two ways:

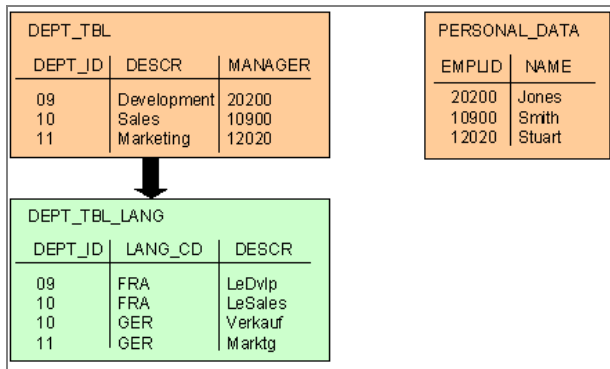
- The name of the table from which it selects.
- The addition of the LANGUAGE_CODE column.

If the base view also selected non-key, non-language-sensitive columns from the base table, there would be no need to select those columns in the related language view because non-language-sensitive fields don't need to be included in related language records.

Two Base Tables, One Related Language Table

When you join two tables, one of which has a related language record, the select statement becomes only slightly more complex.

For example, consider a view that joins DEPT_TBL, which has a related language record, with PERSONAL_DATA, which does not:



Two base tables, one related language table

Here's the select statement for the base view that selects the Department ID and description and the name of the manager for each department:

```
SELECT A.DEPT_ID, A.DESCR, B.NAME
FROM PS_DEPT_TBL A,
     PS_PERSONAL_DATA B
WHERE A.MANAGER = B.EMPLID
```

Here's the select statement for the language-sensitive view:

```
SELECT A.DEPT_ID, A.DESCR, C.NAME, A.LANGUAGE_CD
FROM PS_DEPT_TBL_LANG A,
     PS_DEPT_TBL B,
     PS_PERSONAL_DATA C
WHERE A.DEPT_ID = B.DEPT_ID
AND B.MANAGER = C.EMPLID
```

In this case, the related language view can't get all the information from the related language table. This is because the MANAGER field (which is used to get the manager's name from PS_PERSONAL_DATA) is not language-sensitive and is therefore not part of the related language table.

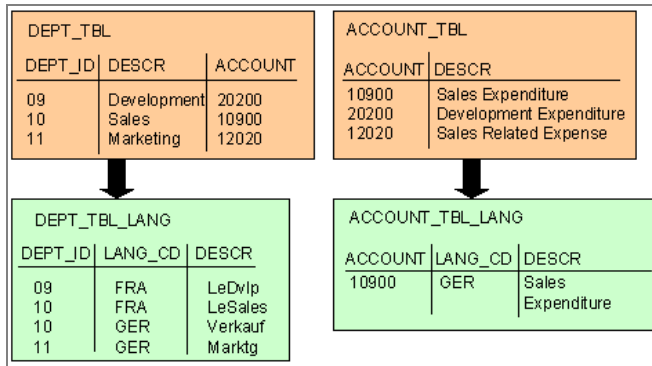
In order to retrieve the MANAGER field, PS_DEPT_TBL_LANG needs to be joined to PS_DEPT_TBL.

Because the PERSONAL_DATA record does not have a related language table, no special logic is required for fetching the manager's name, as it will be the same in all languages.

Two Base Tables, Two Related Language Tables

When you join two tables that both have related language records, your select statement becomes quite complex.

For example, consider a view that joins DEPT_TBL with ACCOUNT_TBL:



Two base tables, two related language tables

In this example, the base view joins DEPT_TBL with ACCOUNT_TBL in order to retrieve the description of the account. Here's the select statement for the base view:

```
SELECT A.DEPT_ID, A.DESCR, B.ACCOUNT, B.DESCR
FROM PS_DEPT_TBL A,
     PS_ACCOUNT_TBL B
WHERE A.ACCOUNT = B.ACCOUNT
```

The difficulty in creating the related language view comes from the fact that both tables referenced in the base view have related language records. Because the related language architecture does not require that all rows have translations, creating the related language view is not as simple as joining DEPT_TBL_LANG with ACCOUNT_TBL_LANG. Your related language view needs to take into account the following scenarios:

1. A translated department may reference a translated account
2. A translated department may reference an untranslated account.
3. An untranslated department may reference a translated account.

In this scenario, there will be no row in DEPT_TBL_LANG, but there will be a row in ACCOUNT_TBL_LANG. If you simply joined DEPT_TBL_LANG with ACCOUNT_TBL_LANG in the view, no translation would be retrieved by the related language view because the department didn't have a translation, even though the account did.

4. An untranslated department may reference an untranslated account.

This situation does not need to be addressed by the related language view because, in this scenario, no translation exists and it would be correct for the related language view to return no rows.

To address these scenarios, the SQL statement for the related language view needs to include logic for the first three scenarios—those where translations exist. Here's the select statement for the language-sensitive view:

```

SELECT A.DEPT_ID, A.DESCR, C.ACCOUNT, C.DESCR, A.LANGUAGE_CD

FROM PS_DEPT_TBL_LANG A,

      PS_DEPT_TBL B,

      PS_ACCOUNT_TBL_LANG C

WHERE A.DEPT_ID = B.DEPT_ID

      AND B.ACCOUNT = C.ACCOUNT

      AND A.LANGUAGE_CD = C.LANGUAGE_CD

UNION

SELECT E.DEPT_ID, E.DESCR, G.ACCOUNT, G.DESCR, E.LANGUAGE_CD

FROM PS_DEPT_TBL_LANG E,

      PS_DEPT_TBL F,

      PS_ACCOUNT_TBL G

WHERE E.DEPT_ID = F.DEPT_ID

      AND F.ACCOUNT = G.ACCOUNT

      AND NOT EXISTS (SELECT 'X'

                      FROM PS_ACCOUNT_TBL_LANG H

                      WHERE H.ACCOUNT = G.ACCOUNT

                      AND H.LANGUAGE_CD = E.LANGUAGE_CD)

UNION

SELECT I.DEPT_ID, I.DESCR, J.ACCOUNT, J.DESCR, J.LANGUAGE_CD

FROM PS_DEPT_TBL I,

      PS_ACCOUNT_TBL_LANG J

WHERE I.ACCOUNT = J.ACCOUNT

      AND NOT EXISTS (SELECT 'X'

                      FROM PS_DEPT_TBL_LANG K

```



```

WHERE K.DEPT_ID = I.DEPT_ID

AND K.LANGUAGE_CD = J.LANGUAGE_CD)

```

This view is really three separate SQL statements whose output is concatenated using the SQL UNION operator. Each select statement in the view addresses one of the first three scenarios described above. Let's examine this view, statement by statement:

```

SELECT A.DEPT_ID, A.DESCR, C.ACCOUNT, C.DESCR, A.LANGUAGE_CD

FROM PS_DEPT_TBL_LANG A,

     PS_DEPT_TBL B,

     PS_ACCOUNT_TBL_LANG C

WHERE A.DEPT_ID = B.DEPT_ID

      AND B.ACCOUNT = C.ACCOUNT

      AND A.LANGUAGE_CD = C.LANGUAGE_CD

```

This select statement addresses scenario 1; it retrieves the rows where translations for both the department and the account exist.

```

SELECT E.DEPT_ID, E.DESCR, G.ACCOUNT, G.DESCR, E.LANGUAGE_CD

FROM PS_DEPT_TBL_LANG E,

     PS_DEPT_TBL F,

     PS_ACCOUNT_TBL G

WHERE E.DEPT_ID = F.DEPT_ID

      AND F.ACCOUNT = G.ACCOUNT

      AND NOT EXISTS (SELECT 'X'

                      FROM PS_ACCOUNT_TBL_LANG H

                      WHERE H.ACCOUNT = G.ACCOUNT

                            AND H.LANGUAGE_CD = E.LANGUAGE_CD)

```

This select statement addresses scenario 2; it handles instances where the department translation exists, but the account translation does not exist.

The sub-select statement is required in order to prevent this statement from retrieving records that have already been selected by the statements that address scenario 1, where a translated account may exist.

```

SELECT I.DEPT_ID, I.DESCR, J.ACCOUNT, J.DESCR, J.LANGUAGE_CD

FROM PS_DEPT_TBL I,

     PS_ACCOUNT_TBL_LANG J

```

```

WHERE I.ACCOUNT = J.ACCOUNT

AND NOT EXISTS (SELECT 'X'

FROM PS_DEPT_TBL_LANG K

WHERE K.DEPT_ID = I.DEPT_ID

AND K.LANGUAGE_CD = J.LANGUAGE_CD)

```

This select statement addresses scenario 3, where translations exist for the account but not the department. Again, the sub-select statement is needed to avoid returning rows that have already been selected by the other select statements in the SQL statement where a translated department exists.

The Translate Table (XLATTABLE)

The Translate table (XLATTABLE) is unique in that both base and non-base data is stored in the same table. All PeopleTools designer and runtime tools understand this distinction and handle it appropriately; however if you create a view over the XLATTABLE or a view that joins in rows from XLATTABLE, PeopleTools no longer recognizes that special logic is needed.

Therefore, both base views and related language views that reference XLATTABLE require an extra clause to ensure that only the appropriate rows are selected.

Remember—this work is required only if you create a view that references XLATTABLE. Simply including XLATTABLE values on a page as related display fields does not require any special logic, as PeopleTools handles such instances automatically.

For base views, you'll select values where the language code is equal to the base language. Not including a clause in the base view to restrict values to the base language will result in your base language view returning rows in all the languages where translations exist.

For related language views, you do exactly the opposite: you select rows where the language code does not equal the base language.

To determine the base language in SQL, simply join or sub-select to the PSOPTIONS table. PSOPTIONS.LANGUAGE_CD is always set to the base language code of the database. Because PSOPTIONS contains only one row, there is no need to include a WHERE clause on the join or sub-select statement.

For example, consider the Translate table values for the Status field where both English and German translations exist:

XLATTABLE			
FIELDNAME	VALUE	LONGNAME	LANGUAGE_CD
STATUS	A	Approved	ENG
STATUS	D	Denied	ENG
STATUS	R	Recycle	ENG
STATUS	A	Genehmigt	GER
STATUS	D	Abgelehnt	GER

XLATTABLE

Here is the select statement for the base view, which is retrieving the translate values for the STATUS field:

```
SELECT VALUE, LONGNAME FROM XLATTABLE

WHERE FIELDNAME = 'STATUS'

      AND LANGUAGE_CD =

      (SELECT LANGUAGE_CD

      FROM PSOPTIONS)
```

Here is the select statement for the related language view:

```
SELECT VALUE FROM XLATTABLE

WHERE FIELDNAME = 'STATUS'

      AND LANGUAGE_CD <>

      (SELECT LANGUAGE_CD

      FROM PSOPTIONS)
```


CHAPTER 5

Working With Language-Sensitive Application Data

This section describes how to access and edit non-base language application data that is stored in related language tables. This section also explains and how the system maintains related language tables and base tables when users make changes to this data.

You can also use the translation method described in the chapter to translate names and descriptions of PeopleSoft Portal objects (content providers, folders, content references, and attributes). Note, however, that attribute definitions include an option to disallow translation of the attribute.

This section does *not* cover the maintenance of non-base language data that is stored in the Translate table, nor does it discuss the maintenance of related language tables that store non-base language descriptions of PeopleTools objects.



For more information about disallowing translations for specific folder attributes in the PeopleSoft Portal, see *Using Portal Administration Features*. For more information about translating and maintaining Translate table values and text descriptions of PeopleTools objects, see *Translating With Design Tools* and *Using the Translate Utilities*. **For information** about the structure of related language tables and how they work, see *Understanding Related Language Tables*.

Translating Data by Changing the Signon Language

To access language-sensitive data, you can simply log on to the database using the language in which you want to work. Remember, if a translation does not yet exist, you will see the base language data, even if you logged on using a non-base language. But when you enter data in this mode, the data is saved directly to the related language table.

This situation makes it easy to translate application data. Simply sign on using the target language. Go to the page with the data, where you will see the base language data. Overwrite the base language data with the translated data. When you save the page, the translation is saved in the related language table.

For example, suppose that your base language is English and you are translating description fields in the Action Reason table into Spanish. The base table in this case is ACTN_REASON_TBL; the related language table is ACTN_RSN_LANG. The following table shows relevant fields from a row of data in the base table:

ACTION	REASON	EFFDT	DESCR	DESCRSHORT
PAY	REC	01/01/95	Job Reclassification	Job Reclas

If this row has never been translated into Spanish, then there is no child row in the related language table with a LANGUAGE_CD of ESP. Once the row is translated, a new row is inserted into the related language table, similar to the following:

ACTION	REASON	EFFDT	LANGUAGE_CD	DESCR	DESCRSHORT
PAY	REC	01/01/95	ESP	Reclasificaci ón Trabajo	Reclas

If you are translating fields in an effective-dated base table (which is the case in the example), you need to use Correct History mode—the only mode that permits you to overwrite historical, current, and future rows.

If you insert a new row into a table with the signon language set to a non-base language, the system assumes that you want to update the information in the base table and adds new rows to both the base table and the related language table. This normally results in non-base language descriptions entering the base table. To maintain the consistency of the language in the base table fields, you need to translate the descriptions in the new row into the base language.



For more information, see Understanding Related Language Tables.

To translate data in related language tables by changing the signon language:

1. Sign on to PeopleSoft using the target language for your translation.

If you have already signed on, you can change your signon language on the International Preferences page.



For more information about signing on, see Choosing a Signon Language. For more information about the International Preferences page, see Changing the Signon Language While Signed On.

2. Open a page that provides access to the values you want to translate.
3. Click Correct History.

This puts you in correction mode. We use correction mode because the Action Reason table is effective-dated, which prevents the translator from making changes to the description fields in Update/Display mode.

4. Translate the description fields.

When the page appears, the description fields display the values that are stored in the base table.

The screenshot shows a web form titled "Action Reason Table". It contains several fields: "Action:" with value "PAY Pay Rate Change", "Reason Code:" with value "REC", "*Effective Date:" with value "01/01/1900", and "*Status:" with a dropdown menu set to "Active". Below these are two text input fields: "*Description:" containing "Job Reclassification" and "Short Description:" containing "Job Reclas". At the bottom left of the form are two small icons representing the United States and Spain.

Untranslated descriptions displayed in the base language

Enter the Spanish translations into the description fields.

This screenshot is identical to the previous one, but the text in the description fields has been translated into Spanish. The "*Description:" field now contains "Reclasificación Trabajo" and the "Short Description:" field contains "Reclas". The rest of the form, including the "Action:", "Reason Code:", "Effective Date:", and "Status:" fields, remains the same.

Descriptions translated into the target language

5. Save the page.

When you save the page, the system adds a new row to the related language table. No changes are made to the base table.

How the System Maintains Related Language Tables and Base Tables

When a user changes data on a language-sensitive page, the system responds by making changes to the base table or the related language table or both. This topic explains how the system responds when you translate fields or insert or delete rows in a language-sensitive table.

Action	Signon Language	Result
Translate a language-sensitive field (the base language field has never been translated).	Non-Base	Adds a new row to the related language table that is keyed to the current signon language.
Edit a language-sensitive	Base	Changes a field in the base table without

field.		affecting the related language table.
Edit a language-sensitive field.	Non-Base	Changes a field in a row of the related language table that is keyed to the current signon language. Has no effect on the base table.
Edit a non-language-sensitive field.	Base	Changes a field in the base table without affecting the related language table.
Edit a non-language-sensitive field.	Non-Base	Changes a field in the base table without affecting the related language table. As the related language table contains only language-sensitive fields, no changes to this table are necessary.
Add a row (new key)	Base	Adds a new row to the base table without changing the related language table.
Add a row (new key)	Non-Base	Adds a new row to the base table and a new row, keyed to the current signon language, to the related language table. This could introduce non-base language descriptions into the base language table.
Insert a row (effective-dated)	Base	Adds a new effective-dated row to both the base table; add a new effective-dated rows to the related language table for each language for which translations exist for the object. Any updates to language-sensitive fields are applied only to the base table.
Insert a row (effective-dated)	Non-Base	Adds a new effective-dated row to both the base table and the related language table for all languages for which translations exist for the object. Any updates to language-sensitive fields are applied to the current language in the related language table and to the base table.
Delete a row	Base or Non-Base	Deletes the base table row and all dependent rows in the related language table.

Editing Data in Multiple Languages Using MLS



The Multilingual Support (MLS) feature is not available in the PeopleSoft Internet Architecture in PeopleTools 8.12, but it is available in the Windows Client.

Multilingual Support (MLS) is a feature that enables users to enter or edit data in multiple languages during the same session without changing signon language. This feature makes it

much simpler and more intuitive for a multilingual user, such as a Canadian operator who is proficient in both French and English, to maintain data in several languages before saving a page. Additionally, MLS enables users to view all translations of a row of data in a single session, which may be important for multilingual implementations that need to describe data in a different language.

MLS is active if both of the following are true:

- The multilingual support option is enabled for the user operator.
- The current page contains a record that has a related language record.

To enable multilingual support:

1. Select PeopleTools, Maintain Security, Use, User Profiles.
2. Search for an existing **User ID**, or add a new one.

Use the standard search or add method to enter the User ID and access the ID page.

The General page appears.

The screenshot shows the 'General' tab of the 'User Profiles' page. The 'User ID' field contains 'PTDMO' and the 'Description' is 'Unger, Annette'. There is a checkbox for 'Account Locked Out?'. The 'Logon Information' section includes 'Symbolic Id' (sa1), 'Password', and 'Confirm Password' fields. The 'General Attributes' section includes 'Email ID', 'Language Code' (English), 'Multi Language Enabled?' checkbox, 'Currency Code', and 'Enable Expert Entry' checkbox. The 'Permission Lists' section includes 'Navigator Homepage', 'Process Profile', 'Primary', and 'Row Security' fields, each with a search icon and an 'Explain' link. At the bottom are buttons for 'Save', 'Return to Search', 'Add', and 'Update/Display'.

User Profiles - General page

3. Select Multi Language Enabled?
4. Click Save.

To enter data in multiple languages:

1. Navigate to the page on which the data is to be maintained.



In PeopleTools 8.12, you can use this feature only in the Windows client. Until this feature is enabled in the PeopleSoft Internet Architecture, you must navigate to the page in Windows.

Any language-sensitive fields in the panel are highlighted in yellow to make it easy to identify fields that permit data maintenance in multiple languages.

2. Choose a target language.

In the Windows client, choose a language from the Languages menu that appears.

Note that this does not change the current panel language: the language displayed on the menu and panel controls remains the same, and only the language-sensitive data is reloaded in the language selected.

3. Enter or edit data in any language-sensitive fields in the target language.

4. When you have finished editing the multilingual fields, save the panel group.

The system will update the base and related-language tables for the panel.

Translating With Design Tools

This chapter describes how to translate PeopleTools objects using design tools (PeopleSoft Application Designer, Tree Manager, and Process Scheduler). Before you can translate objects with design tools, you must have a good understanding of those tools. In particular, you need to understand the different types of objects, and you need to know how to open objects and access their properties.



For more information, see *Using Application Designer, Tree Manager, and Process Scheduler*.

Choosing a Translation Tool

Some objects, such as images, HTML areas, and icon labels in workflow maps, can be translated only with the design tools. For many objects, however, a second option exists: the translate utilities.

The translate utilities enable a translator to simultaneously view the base language and the translation of each object being translated, which can be extremely important for review and proofing purposes.

For most objects that can be translated with either design tools or translate utilities, choosing a translation tool is mostly a matter of personal preference. A translator who has good technical skills and who is familiar with PeopleTools may prefer to translate using the design tools. A translator who is primarily a linguist, with limited knowledge of PeopleTools, may prefer to use the translate utilities whenever possible.

When translating field values and xlat (translate) values, however, the translate utility knows as the Translation Workbench offers efficiencies that make it a significantly more powerful option than translating with the design tools.

The translate utilities are described in a separate chapter.



For more information, see *Using the Translate Utilities*.

Overview of System Object Translation

PeopleTools stores non-base language descriptions of language-sensitive PeopleTools objects in related language tables. This enables you to translate the description and other text of these objects (which include fields, field labels, messages, menus, workflow maps, and others) by setting your Configuration Manager language preference to the target language, logging in to PeopleSoft Application Designer, and then editing the description text in the object's definition or property settings. The system maintains these textual fields in base and related language tables in the same way that it maintains language-sensitive application data.

The Translate table is used as an alternative to related language tables for storing some types of language-sensitive application data. As with related language tables, you can translate the descriptive text in the Translate table into non-base languages. However, in this case, both base and non-base text are stored in the same table. The translate table (XLATTABLE) is an exception to the standard PeopleTools Related Language Table architecture.

Pages are also exceptional in that they require more than a simple translation of descriptive text: The translation of pages may require modification of the visual layout of the GUI elements (to account for differing text length in each language) or substitution of several page elements such as images. PeopleTools page definitions include a Language property that makes it possible to design multiple pages that share the same name, but differ by language. This enables you to design specific pages for each supported language. When a user signs in to PeopleSoft in a non-base language, an application displays the non-base language page if it exists; if it doesn't exist, the application displays the base language page.



For more information, see Related Language Tables.

Translating Object Descriptions and Labels Using PeopleSoft Application Designer

In many cases, the language-sensitive descriptions of PeopleTools objects are used as text in labels or other GUI elements (this is true for menu labels, business process icon labels, and some page control labels).

The topics in this section provide:

- An overview of description and label translation. This includes a summary table and a generic procedure for translating objects using design tools. If you are already experienced with PeopleTools, this section may provide all you need to know.
- A section about each object type, including a procedure for translating that object's description or label.

Overview of Description and Label Translation

The following table summarizes information about translating object descriptions and labels. It lists objects that have language-sensitive descriptions and labels, specifies where you need to go in the system to translate the descriptions, and specifies where the translated descriptions are displayed to users. Unless otherwise specified, the translations are done in PeopleSoft Application Designer.

Object	Where Displayed	Where Translated With Design Tools	Where Translated With Translate Utilities
Field	Field labels on pages, in search dialog boxes, and in descriptions in other PeopleTools such as PeopleSoft Query. Also can be used in PS/nVision and SQR reports, through a reference from the Strings Table.	Field definition	Translation Workbench
Record	Application Designer dialog boxes and some node labels in Tree Manager.	Record object properties dialog box.	PeopleTools, Translate, Use, Translate Records
Menu navigation	Menus (both PeopleSoft Internet Architecture navigation and Windows client menus).	Menu definition (Menu Group, Menu bar, and Menu Item Properties dialog box).	PeopleTools, Translate, Use, Translate Menu and PeopleTools, Translate, Use, Translate Menu Item
Folder tabs	Page folder tabs	Component	PeopleTools, Translate, Use, Translate Component
Workflow map icons	Icon labels in business process maps.	Business process properties	Not available
Process definitions, process type definitions, and job type definitions	List boxes in Process Request and Process Scheduler dialog boxes.	Definitions in Process Scheduler: the Process Definitions, Process Type Definitions, and Job Definitions components, all of which are found	Not available

		under PeopleTools, Process Schedule Manager, Use, Process Definitions.	
Tree	List boxes in Tree Manager dialog boxes.	Tree definition in Tree Manager.	Not available
Tree Node	Tree node label.	Pages accessed via tree node in Tree Manager.	Not available

The following generic procedure describes how to translate these object descriptions into non-base languages.

To translate the description fields of a defined object:

1. If necessary, set the Configuration Manager language preference to the target language.

The target language is the language into which to translate the description fields. Alternatively, if you have already signed in to PeopleTools, use the International Preferences page to change the current signon language.



For more information about the International Preferences page, see [Changing the Signon Language While Signed On](#).

2. Log in to the PeopleSoft system.
3. Open the object definition or properties.

This step is different for each object. For example:

- To translate a field description, open the field definition in PeopleSoft Application Designer.
 - To translate a record description, open the record definition in Application Designer, and then open the record's object properties dialog box under **File, Object Properties**.
 - To translate the description of a tree in PeopleSoft Tree Manager, open the tree, and then open its definition (press CTRL+D).
4. Translate the description into the target language
 5. Save the definition.

Translating Field Descriptions in PeopleSoft Application Designer

Field description translations impact large portions of an application. Although it's possible to translate fields using design tools as described in this chapter, the most efficient way by far is to use the Translation Workbench, part of the translation utilities.



For information about translating fields using the translation utilities, see Using the Translation Workbench.

To translate field descriptions in PeopleSoft Application Designer:

1. Change the Configuration Manager language preference setting to the target language.
2. Log in to PeopleSoft
3. Open Application Designer.
4. Open the field definition.
5. Translate the **Long Name** and **Short Name** fields into the target language.
6. Save the field definition.

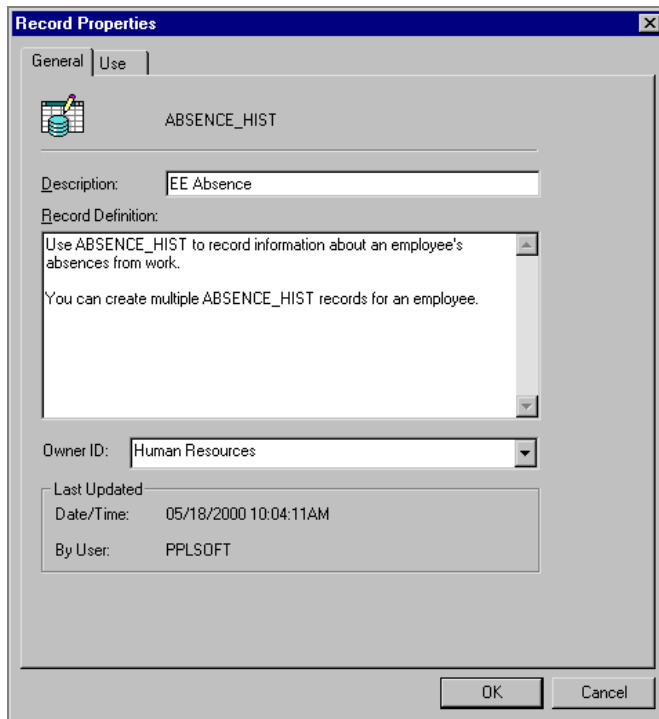
Translating Record Descriptions in PeopleSoft Application Designer

Record definitions have language-sensitive description fields that you can translate in PeopleSoft Application Designer. These translated descriptions appear in the database view in PeopleSoft Query and also provide a convenience for PeopleTools users who work in Application Designer in a non-base language.

To translate a record definition:

1. Change the Configuration Manager language preference setting to the target language.
2. Log in to PeopleSoft.
3. Open Application Designer.
4. Open the record definition that you want to translate.
5. Open the Record Properties dialog box
6. Translate the description fields into the target language.

Translate the **Description** field and, optionally, the **Comments** field, which is also language sensitive.



Record Properties dialog box

7. Click **OK** to close the dialog box.
8. Save the record definition.

Translating Menu Navigation in Application Designer

All visible parts of PeopleSoft menus are language-sensitive.

Although all levels of navigation are defined in menu objects, different levels are defined by different sets of properties

The menu properties define the first two levels of navigation:



Navigation controlled by the menu properties

Bar item properties define the next level of navigation, which are typically limited to some combination of Use, Setup, Process, Inquire, and Report.



Navigation controlled by bar item properties

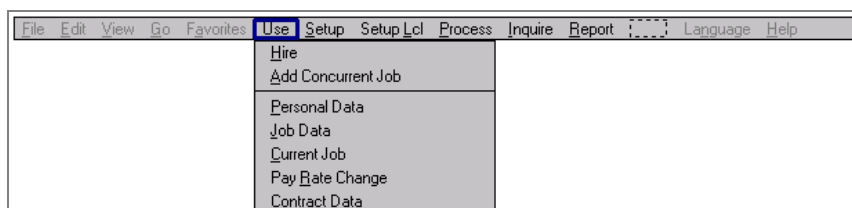
Menu item properties define the final level of navigation—the level that is associated with an individual component.



Navigation controlled by menu item properties

To translate menu navigation using PeopleSoft Application Designer:

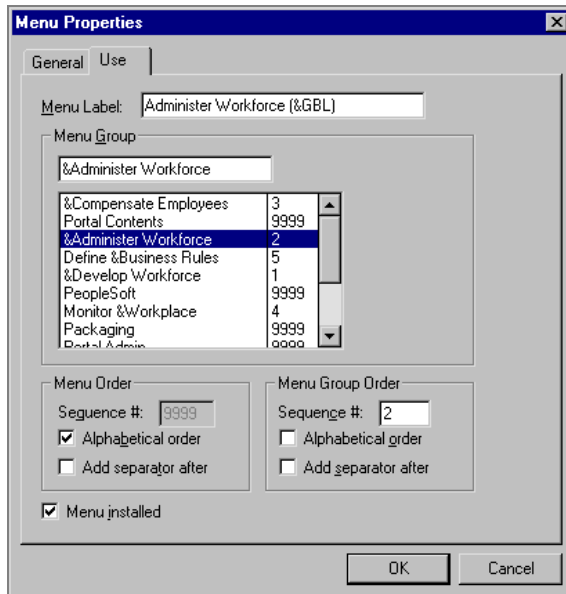
1. Change the Configuration Manager language preference setting to the target language.
2. Log in to PeopleSoft.
3. Open Application Designer.
4. Open the menu definition.



Menu Definition

5. On the Menu Properties - Use tab, translate the first two navigation levels.

To access the menu properties, click the Properties icon, or right-click the menu definition and select Menu Properties from the pop-up menu. In the Menu Properties dialog box, click the Use tab.



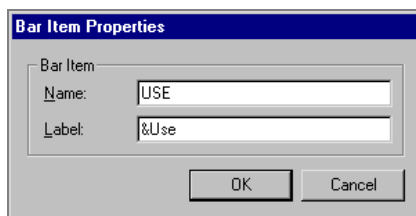
Use Tab in Menu Properties dialog box

The **Menu Group** is the first level of navigation.

The **Menu Label** is the second level of navigation.

6. In the Bar Item Properties dialog box, translate the next navigation level.

To access the Bar Item properties dialog box, double-click the menu bar label. The **Label** in the Properties dialog box controls this navigation level—the level that corresponds to the standard Use, Setup, Inquire, Process, and Report options.



Bar Item Properties dialog box

7. In the Menu Item Properties dialog box, translate the final navigation level.

To access the Menu Item properties dialog box, double-click the menu item label. The **Label** in the properties dialog box controls the final navigation level—the level that corresponds to the component that will be displayed.

Menu Item Properties dialog box

Translate only the **Label** field. *Do not* change the **Name** field.

8. Repeat steps 5 and 6 for each bar item and menu item.
9. Save the menu definition.

Translating Folder Tab Labels

Component definitions include two labels for each page in the component: an item label and a folder tab label. Both are language-sensitive and can be translated in PeopleSoft Application Designer or in the translation utilities.

In the PeopleSoft Internet Architecture, only one of these is visible to the user. If a folder tab label exists, it is used on the folder tab for the corresponding page. The item label is ignored. If no folder tab label exists, the item label is used as the folder tab label.

In the Windows client, item labels become part of the navigational structure, so if the item label and folder tab label are different, both should be translated. However, because folder tabs automatically use the item label when the folder tab label is missing, it is common to have item labels only.

To translate folder tab labels in PeopleSoft Application Designer:

1. Change the Configuration Manager language preference setting to the target language.
2. Log in to PeopleSoft.
3. Open Application Designer.
4. Open the component definition.

JOB_DATA.GBL (Component)						
Definition		Structure				
	Page Name	Item Name	Hidden	Item Label	Folder Tab Label	Allow Deferred Processing
1	JOB_DATA1	JOB_DATA1	<input type="checkbox"/>	&Work Location		<input checked="" type="checkbox"/>
2	JOB_DATA_JOBCO	JOB_DATA_JOBCODE	<input type="checkbox"/>	&Job Information		<input checked="" type="checkbox"/>
3	JOB_LABOR	JOB_LABOR	<input type="checkbox"/>	Job &Labor		<input checked="" type="checkbox"/>
4	JOB_DATA2	JOB_DATA2	<input type="checkbox"/>	&Payroll		<input checked="" type="checkbox"/>
5	JOB_DATA_SALPL	JOB_DATA_SALPLAN	<input type="checkbox"/>	&Salary Plan		<input checked="" type="checkbox"/>
6	JOB_DATA3	JOB_DATA_3	<input type="checkbox"/>	&Compensation		<input checked="" type="checkbox"/>
7	JOB_DATA_ERNDI	JOB_DATA_DISTRIBU	<input type="checkbox"/>	Job Earnings &Distri		<input checked="" type="checkbox"/>
8	JOB_DATA_BENP	BENEFIT_PROGRAM_	<input type="checkbox"/>	&Benefit Program Pa		<input checked="" type="checkbox"/>
9	EMPLOYMENT_DT	EMPLOYMENT_DATA	<input type="checkbox"/>	Employment &Inform		<input checked="" type="checkbox"/>
10	EMPLOYMENT_DT	EMPLOYMENT_DTA2	<input type="checkbox"/>	&Employment Dates		<input checked="" type="checkbox"/>
11	JOB_DATA1_WRK	JOB_DATA1_WRK	<input checked="" type="checkbox"/>	Job Data1 Wik		<input checked="" type="checkbox"/>
12	ENCUMB_TRIGGE	ENCUMB_TRIGGER	<input checked="" type="checkbox"/>	Encumb Trigger		<input checked="" type="checkbox"/>

Component

5. Translate the item labels and folder tab labels into the target language.

You can edit the labels directly in the cells in the **Item Label** and **Folder Tab Label** columns on the grid.

If the cell in the folder tab label column is blank, the item label will appear in the folder tab.

If there is both an item label and a folder tab label, be aware that only the folder tab label will be visible in the PeopleSoft Internet Architecture. The item label appears only on Windows menus.

For items that users access in Windows menus, you may want to include ampersands within the item label text to create accelerator keys.

6. Save the component definition.

Translating Icon Labels on Workflow Maps

All icon labels in workflow maps are language-sensitive. You can translate icon labels in PeopleSoft Application Designer through the icon definitions, which you can access by right-clicking the icon.

Because workflow maps are not visible to users except through activity guides, you'll normally translate only the icons that are in activity guides.



For more information about activity guides, see *Designing Activity Guides*.

To translate an icon label:

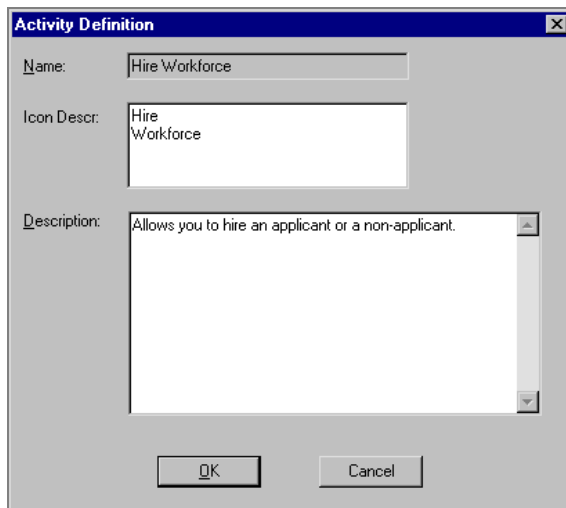
1. Change the Configuration Manager language preference to the target language.
2. Log in to PeopleSoft.
3. Open Application Designer.

4. Open the workflow map.

You can use this procedure for both business processes and activities.

5. Right-click the icon to access the component properties.

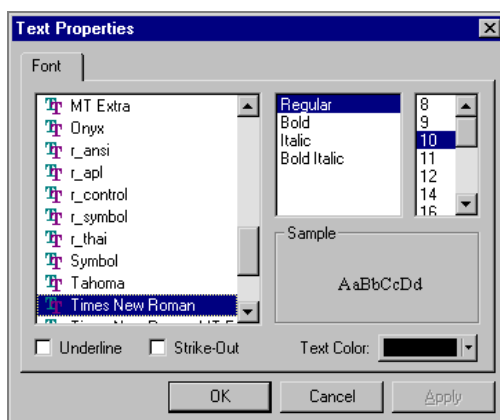
A definition dialog box for the type of icon you selected is displayed. The following dialog box is for an activity icon in a business process map.



Activity Definition dialog box

6. Translate the **Icon Descr** (icon description) field and, optionally, the **Description** field into the target language.

You can also change the font of the label by double-clicking the label itself and selecting Font from the pop-up menu. This may be necessary if you are translating the labels into languages that need language-specific fonts, such as Japanese.



Text Properties dialog box

Translating PeopleSoft Process Scheduler Definitions

PeopleSoft Process Scheduler has three language-sensitive definitions:

- Process Type definitions
- Job definitions
- Process definitions

You can translate the descriptive text for these definitions in Process Scheduler by changing the Configuration Manager language preference and then logging on and translating the descriptive text in the definition. The translated descriptive text appears in PeopleSoft Process Scheduler dialog boxes.

To translate a PeopleSoft Process Scheduler definition:

1. Sign on to PeopleSoft using the target language.



For more information about signing on to PeopleSoft in a particular language, see [Choosing a Signon Language](#).

2. Navigate to Process Scheduler, and open the definition that you want to translate.

To start Process Scheduler, choose **PeopleTools, Process Scheduler Manager**. To open a definition, choose one of the following:

- To open a Process Type definition, choose **Use, Process Type Definitions**.
- To open a Job definition, choose **Use, Job Definitions**.
- To open a Process definition, choose **Use, Process Definitions**.

3. Translate the descriptive text fields into the target language.

Translate the Description field for the Process Type and Job definitions. For the Process definition, translate the Description field and, optionally, the Long Descriptions field, which is also language-sensitive.

Process Definition | Process Definition Options | Override Options | Destination ▶

Process Type: SQR Process
 Name: MASSCHNG

*Description:
 Long Description:
 *Priority: Medium ▼

☒ API Aware
☒ Log client request
☐ SQR Runtime

Detail of Process Definition Page

4. Save the page.

Translating Tree Descriptions and Tree Node Labels

In PeopleSoft Tree Manager, you can translate the tree description and the labels of nodes that correspond to language-sensitive record fields.

The translated tree descriptions appear in Tree Manager list boxes.

Descriptive labels on tree nodes are derived from description fields in the record associated with the node. If you double-click the tree node, PeopleTools will transfer you to the page where the node's properties are defined. If the record is language-sensitive (that is, if it has an associated related language table), the tree nodes' descriptive text will be language-sensitive, and can be translated, using this page, in the same way that you translate other application data.



For more information about PeopleSoft Tree Manager, see Tree Manager.

To translate a tree description:

1. Change the Configuration Manager language preference to the target language.
2. Log in to PeopleSoft.
3. Start Tree Manager.

Choose **Go, PeopleTools, Tree Manager**.

4. Open the tree whose description you want to translate.

Choose **File, Open**, and select the tree that you want to open, using the Open Tree dialog box.

5. Open the tree definition.

Choose **Edit, Tree Definition**, or press CTRL+D, to open the tree definition.

Tree Definition properties

6. Translate the description field into the target language.

Translate the **Description** field, leaving all other fields unchanged.

7. Click **OK** to close the dialog box.
8. Save the tree.

To translate a language-sensitive tree node label:

1. Change the Operator Language Preference setting to the target language.

Use the International Preferences utility to set Operator Language Preference to the language into which to translate.



For more information about the International Preferences utility, see [Changing the Signon Language While Signed On](#).

2. Log in to PeopleSoft.
3. Start Tree Manager.

Choose Go, PeopleTools, Tree Manager.

4. Open the tree whose description you want to translate.

Choose **File, Open**, and select the tree you want to open, using the Open Tree dialog box.

5. Double-click the node you want to translate.
6. Translate the language-sensitive fields on the page into the target language.

7. Save the page.

Saving the page returns you to the tree.

Translating Non-Text Objects

You translate non-text objects—images and HTML objects—the same way that you translate text tables: by logging in to PeopleSoft Application Designer in your target language and modifying the object. The only difference is that you update the object itself rather updating a textual descriptions in a Properties dialog box.

Translating Images

You typically translate images that contain textual elements or elements that are specific to one language or culture.

To translate an image:

1. Change the Configuration Manager language preference to the target language.
2. Log in to PeopleSoft.
3. Open Application Designer.
4. Open the image.

Choose **File, Open**, and select the image you want to open.

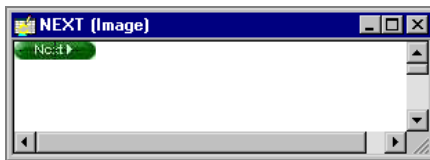


Image definition

5. Select Edit, Update Image.
6. Open the file with the translated image.

A standard Open dialog box appears.

The new graphic replaces the original graphic in the image.

7. Save the image.

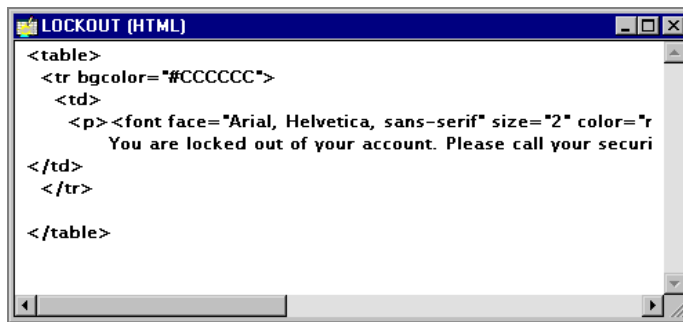
Translating HTML Objects

You typically translate HTML objects when they contain hard-coded text.

To translate an HTML object:

1. Change the Configuration Manager language preference to the target language.
2. Log in to PeopleSoft.
3. Open Application Designer.
4. Open the HTML object.

Choose **File, Open**, and select the HTML object you want to open.



HTML definition

5. Edit any language-sensitive text in the HTML object.
6. Save the HTML object.

Translating Translate Table Values



Don't be confused by the name *Translate table*. Although the Translate table plays an important role in globalization, it is not specifically associated with language translation. It is a common table that is used throughout PeopleTools and PeopleSoft applications.

The Translate table is used as an alternative to related language tables for storing some types of language-sensitive application and system data. Although it's possible to translate values in the translate table using design tools, the most efficient way is to use the Translation Workbench, part of the translate utilities.

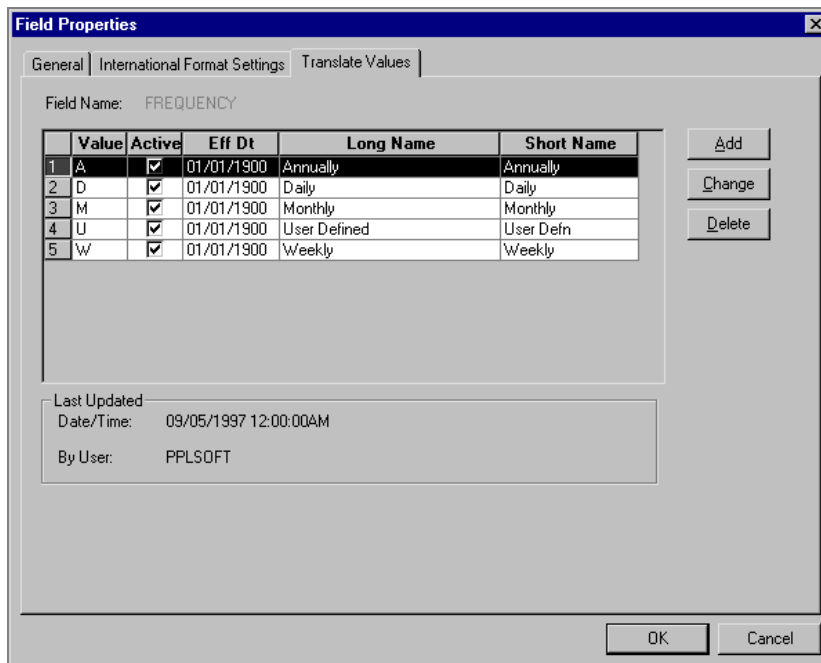


For information about using the translate utilities to translate values in the Translate table, see *Using the Translation Workbench*.

To translate values from the Translate table in PeopleSoft Application Designer:

1. Change the Configuration Manager language preference to the target language.
2. Log in to PeopleSoft
3. Open Application Designer.
4. Open the definition of the field whose translate values you want to access.
5. Access the Translate values in the Field Properties dialog box.

Click the Properties icon (or right-click in the field definitions and choose Field Properties); then click the Translate Values tab. If the descriptions have not been translated, they appear in the base language.



The image shows the 'Field Properties' dialog box with the 'Translate Values' tab selected. The 'Field Name' is 'FREQUENCY'. Below the field name is a table with columns: Value, Active, Eff Dt, Long Name, and Short Name. The table contains five rows of data. To the right of the table are buttons for 'Add', 'Change', and 'Delete'. At the bottom of the dialog are 'OK' and 'Cancel' buttons.

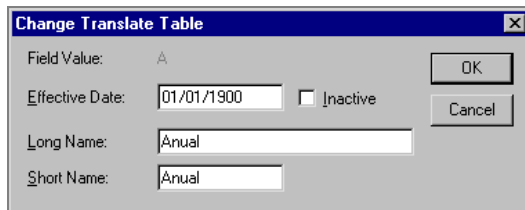
	Value	Active	Eff Dt	Long Name	Short Name
1	A	<input checked="" type="checkbox"/>	01/01/1900	Annually	Annually
2	D	<input checked="" type="checkbox"/>	01/01/1900	Daily	Daily
3	M	<input checked="" type="checkbox"/>	01/01/1900	Monthly	Monthly
4	U	<input checked="" type="checkbox"/>	01/01/1900	User Defined	User Defn
5	W	<input checked="" type="checkbox"/>	01/01/1900	Weekly	Weekly

Last Updated
Date/Time: 09/05/1997 12:00:00AM
By User: PPLSOFT

Translate values for the FREQUENCY field

6. Translate the descriptions into the target language.

For each translate value, click the **Change** button to display the Change Translate Table dialog box:



Change Translate Table dialog box

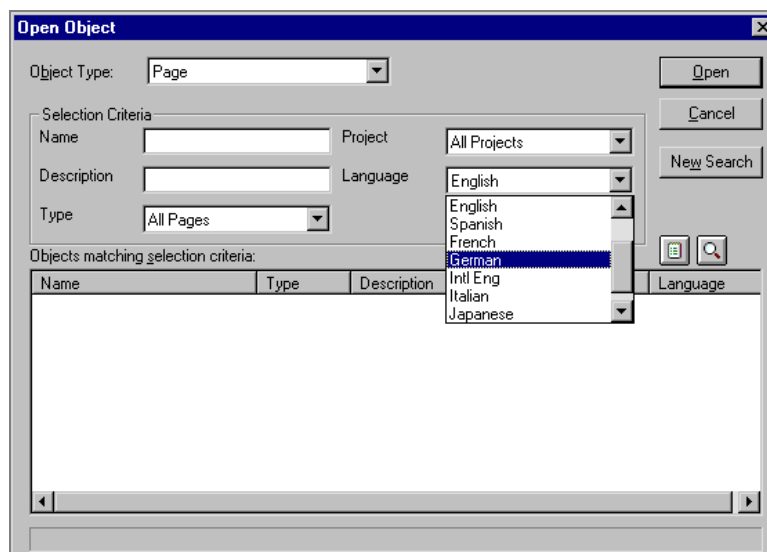
Translate the **Long Name** and **Short Name** fields and click **OK** to accept the changes. Don't change any other fields.

7. Click **OK** to close the Change Translate Table dialog box.
8. Click **OK** to close the Field Properties dialog box.
9. Save the field.

Globalizing Pages

Like other PeopleTools objects, pages have base-language and non-base-language variants. To the application user, pages behave like other language-sensitive objects. If the user logs on using the base language, the system displays the base-language page. If the user logs on using a non-base language, the system displays the non-base language page for that non-base language if the page exists; if there is no page for that non-base language, the system displays the base-language page.

However, unlike other related language objects, non-base-language pages share the same name as their corresponding base-language page, but are in most respects completely independent objects. In PeopleSoft Application Designer you can create, open, delete, modify, or save a specific language variant of a page without affecting other language variants of the same page.



Language drop-down list in Open Object dialog box

All properties of pages, including the number of fields and other controls and their positions and properties are language-dependent. This allows developers to adjust a page for the language in which it is presented to the user. Typical language-sensitive adjustments include:

- Realignment of fields to take into account the label length in the current language
- Rearrangement of field order to conform to each language's norms (for example, Japanese addresses typically display the city before the street name)

You usually create a local-language page by cloning the base-language page (that is, saving it with a different language code) and then modifying the resultant local-language page. The local-language page typically has translated text labels and differences in the arrangement and sizing of page controls to accommodate different-length character strings. The local-language page can also have additional fields, secondary pages, subpages, command pushbuttons, and so forth. Theoretically, this could allow developers to add functionality that is specific to a single language; however, in most cases it is preferable to keep the issues of language preference and locale-specific functionality distinct, by using market-specific components and by placing locale-specific functionality on base-language pages, rather than using the translated versions of pages.

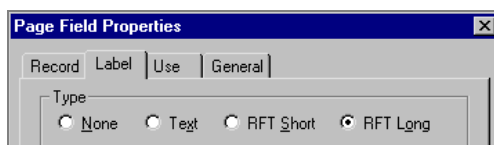
Designing Global-Ready, Base Language Pages

You can simplify a global development project and make a global system easier to maintain by making your base-language pages as translation-ready as possible. This minimizes the work that is required to translate and customize local-language pages. Typically, you should strive to reduce or eliminate textual elements that are maintained on the page definition and, instead, derive those text strings from other PeopleTools objects such as field labels or messages. This way, a translator needs to translate a text string only once, and the new translation will take effect across all pages where that string is referenced.

Following are some tips for designing global-ready base language pages.

- Use page control labels that derive from field descriptions.

On the Label tab of the Page Field Properties dialog box, use the RFT Short or RFT Long setting whenever possible. This cause the control labels, as well as the pushbutton tooltip text, to be derived from the language-sensitive descriptions that are stored in the field definition. Avoid using any other non-language-sensitive text on the page, such as page-based floating text labels.



Page Field Properties dialog box - detail of Label tab

- Associate group boxes with record fields

Using the Page Field Properties dialog box, you can associate group boxes with record fields. In this way, the label for the group box can be derived from field's **RFT Short** or **RFT Long**

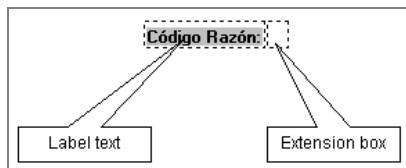
value. The only effect of this association is the label derivation. It has no other effect on the page's operation.

- As much as possible, complete and freeze the layout of the pages early in your development cycle.

Because the entire page layout is stored individually for each language, any changes you make to the page layout, such as adding or deleting fields, will have to be reapplied to each language version of the page. However, changes made to field properties and PeopleCode programs are automatically applied to all language versions of the page.

- Size and arrange the page controls so that there is enough space to accommodate data in non-base languages.

English strings for both labels and data tend to be shorter than strings in other languages. As you work with pages, you will notice that when you select a field, the dotted boundary box has two components—the surround box for the label, and an extension box to the right of the label. This extension box is typically 10 percent of the label size and is a useful guide as to the *minimum* amount of space that you should leave between a field and its label to allow for expansion during translation. As much as possible, try to ensure that there is enough space between the field and the label so that the extension box doesn't overlap the field or other page controls.



Page control label and extension box

Creating a Non-Base Language Page

You could create a non-base-language page from scratch, but it's more practical to create the base-language page first, then save it as a non-base-language page and make any necessary modifications.

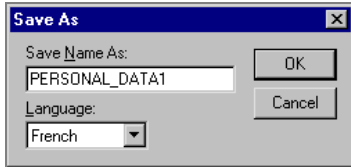
To create a non-base language page:

1. Open the base-language page.

When you open the page, any page fields with associated field descriptions that have already been translated will appear in the target language.

2. Save a copy of the page, setting Language to the target language.

Select **File, Save As** to display the Save As dialog box. Set the Language field to the target language. Save the page under the same name as the base-language page.



Save As dialog box for pages

3. Translate page control labels and other text.

If you translated objects on the base-language page using the Translation Workbench, all field labels and text on the page that are derived from the record definition will already be translated.

If there is any untranslated text, you can change the control's label properties so that the text is derived from the record definition. Once the text is coming from the record definition, you can use the Translation Workbench to complete the translation.

Alternatively, you can manually translate the page-specific text in PeopleSoft Application Designer.

4. Adjust the page control arrangement and sizing to accommodate character string length in the target language.

Once all label text is translated, you will see whether it is necessary to adjust the position of the page controls and labels to prevent crowding or overlap.

5. Make other modifications that your project requires.

Such changes might include changing bitmaps or images on pushbuttons or adding special fields that apply only within the target language.

CHAPTER 7

Using the Translate Utilities

The translate utilities provide an alternative to PeopleSoft Application Designer and other design tools for translating system object descriptions, translate value descriptions, system messages, and queries.

The translate utilities are the most powerful option if you are translating fields, translate (xlat) values, and override text on pages (that is, text that is not derived from the field description), because they allow you to view both the source language and the target language during translation, and they provide a project-based approach to translation.

The translate utilities are also a useful option for translating navigational objects such as menus, menu items, and folder tabs. For these objects, the translate utilities are generally the most convenient alternative if you are translating the entire system at once or if the translator is a linguist with limited experience using PeopleTools. Alternatively, you can translate these objects using the design tools. System messages are typically translated only with the translate utilities.

Not all system objects can be translated using the translate utilities: pages, icons on workflow maps, process scheduler objects, and trees must be translated with design tools.



For information about translating with design tools, see [Translating With Design Tools](#).

Using the Translation Workbench

The Translation Workbench provides an efficient mechanism for translating the highest volume objects and those that are most visible to users: the objects on pages. These objects include field labels, page text that is not derived from field labels, and translate values, which normally appear either as radio button labels or as values in drop-down lists. The Translation Workbench facilitates translations with an easy-to-use grid where you can view the base-language text and enter a translation simultaneously, enabling translators to review translations instead of simply overwriting text.



For more information about translating other types of objects, see [Using the Translate Pages and Translating With Design Tools](#).

Opening the Translation Grid

The translation grid is a Microsoft Excel-like grid showing all the translatable objects, along with their base-language and translated text.

The grid contents are based on the open Application Designer project. This association with Application Designer projects is a handy mechanism for organizing your translation effort. This also means that, to use the Translation Workbench, you must have a project open. If you modify the project, Translation Workbench will not pick up the change until you save the changes.

To open the translation grid:

1. Set your language preference to the target language.

Because you'll be using the Windows client to access the translation grid, you'll set your language preference using Configuration Manager.



For more information about setting your Windows client language preference, see Windows Client Language Preferences.

2. Log in to PeopleSoft Application Designer.
3. Create or open the project with the objects to be translated.

It makes sense to use projects with the non-base language versions of any pages. That way you're sure to include all object on the pages, even if the pages have been modified to include objects that aren't on the base page.



For more information about projects in PeopleSoft Application Designer, see Using Application Designer Projects.

4. Open the translation grid.

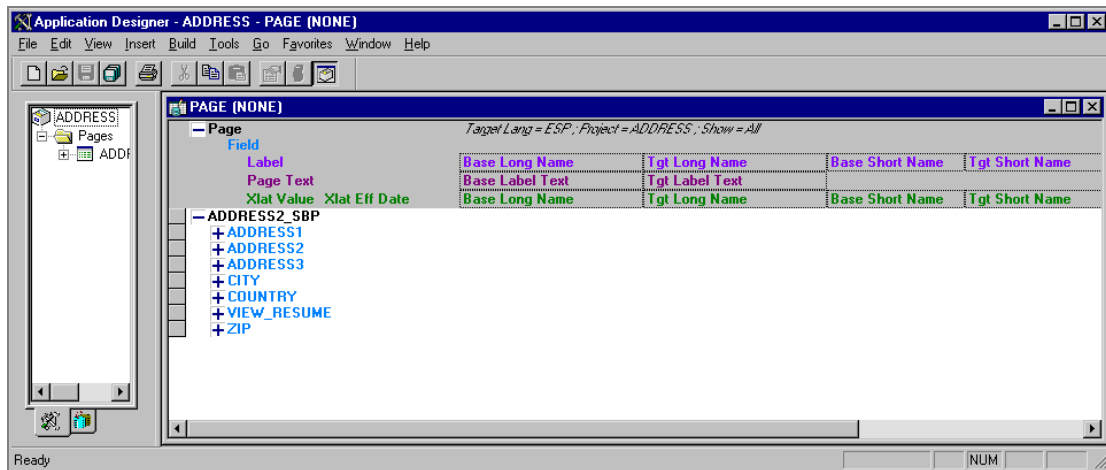
To translate the pages in your project, select **Tools, Translate, Translate Page Objects**. This is the most versatile option because it shows you all translatable objects that are associated with a page: you can use it to translate field labels, panel text (not derived from field labels), and translate values, all in one translation grid.

To translate only the fields in your project, select **Tools, Translate, Translate Fields**. This option lets you translate fields that are directly included as objects in your project. Fields that are part of records or pages in your project are not included unless they are also explicitly included in your project as fields.

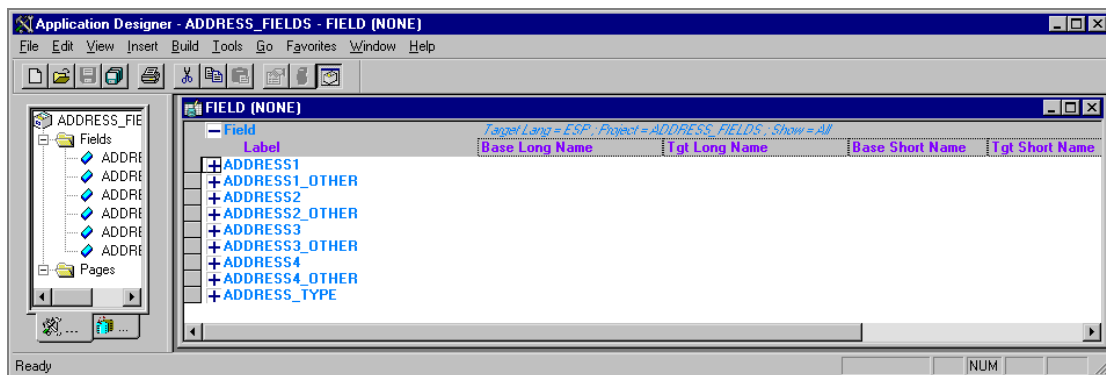
To translate only the xlat values in your project, select **Tools, Translate, Translate Xlats**.

You can also access all of these commands in the pop-up menu that appears when you right-click in the project workspace.

After you select one of the translate options, the appropriate translation grid appears in the object workspace:



Translate Page Objects grid



Translate Fields grid

Depending on which translation grid you opened, different translatable objects from the active project appear in the grid. All objects are arranged hierarchically: pages (if included) are at the top level, fields are at the next level, and field labels, panel text, and translate (xlat) values are at the lowest level. You can use standard tree controls to expand and collapse the view in order to display objects at different levels.



For more information, see [Expanding and Collapsing Nodes](#).

5. (Optional) Arrange the window to maximize the translation grid.

If you want more space for the translation grid, you can maximize the Application Designer window and then maximize the translation grid within the window. For even more space, you can hide the other frames in the Application Designer window. To toggle the display of

the project workspace, select **View, Project Workspace**, or press ALT-0 (zero). To toggle the display of the output window, select **View, Output Window**, or press ALT-1.

Translation Grid Display Options

Understanding the Information Bar

Across the top of the translation grid, an information bar displays helpful information about the project.

Page		Target Lang = ESP ; Project = ADDRESS ; Show = Untranslated			
Field					
Label		Base Long Name	Tgt Long Name	Base Short Name	Tgt Short Name
Page Text		Base Label Text	Tgt Label Text		
Xlat Value	Xlat Eff Date	Base Long Name	Tgt Long Name	Base Short Name	Tgt Short Name

Information bar

Across the top of the information bar, the target language and the current project appear in italics, along with the current viewing option: *All*, *Translated*, *Untranslated*, or *Modified*.

On the left side of the bar, you can see the hierarchical organization that is used to display objects in the grid: pages (at the highest level), fields (at the next level), and then the translatable objects—field labels, page text, and translate (xlat) values—at the lowest level. Each object is color coded; if you are ever unsure what type of object you're working with, the color provides a handy reminder. Fields are blue, field labels are light purple, page text (not derived from a field label) is dark purple, and xlat values are green.

PAGE (NONE)		Target Lang = ESP ; Project = ADDRESS ; Show = All			
Field					
Label		Base Long Name	Tgt Long Name	Base Short Name	Tgt Short Name
Page Text		Base Label Text	Tgt Label Text		
Xlat Value	Xlat Eff Date	Base Long Name	Tgt Long Name	Base Short Name	Tgt Short Name
ADDRESS2_SBP					
Page - Group Box					
ADDRESS1					
Lbl - ADDRESS1		Address Line 1	Dirección1	Address 1	Dirección1
ADDRESS2					
Lbl - ADDRESS2		Address Line 2	Dirección2	Address 2	Dirección2
ADDRESS3					
ADDRESS_TYPE					
Lbl - ADD_TYPE		Address Type	Address Type	Type	Type
Xlt - H 2000-01-01		Home	Home	Home	Home
Xlt - M 2000-01-01		Post Office Box	Post Office Box	PO Box	PO Box
Xlt - O 2000-01-01		Other	Other	Other	Other
Xlt - W 2000-01-01		Work	Work	Work	Work
CITY					
Lbl - CITY		City	Ciudad	City	Ciudad
COUNTRY					
Lbl - COUNTRY		Country	País	Countryt	País
ZIP					
Lbl - ZIP		Postal Code	Cd Postal	Zip	Cd Postal
Lbl - ZIP2		ZIP Code		Zip	

Color-coded objects

In this example, you can see that there is a group box with page text that is not derived from a field label. The ADDRESS_TYPE field has a set of associated xlat values. All other translatable objects are field labels.

Notice also that field labels that appear on the page are in italics. This is apparent when you look at the ZIP field, which has two labels.

Expanding and Collapsing Nodes

The left side of the translation grid is a hierarchical tree control that displays all the objects on the page. Pages are at the top level of the hierarchy, fields are at the next level, and the translatable objects—field labels, panel text, and xlat values—are at the lowest level.

To collapse a single object or to expand a single object by one level, click the + and – switches next to the node. Remember, double-clicking opens the object, so be sure to click only once.



Click to expand the object by one level.



Click to collapse object by one level.

To expand or collapse the entire tree structure, select **View, Expand All** or **View, Collapse All**. Right-clicking anywhere in the grid displays a pop-up menu that contains these options. Another option on the pop-up menu, **Expand Current Object**, expands the selected object by one level.


Choosing Which Rows to View

Right-click anywhere in the translation grid to see a pop-up menu that lets you choose which rows are displayed in the grid:

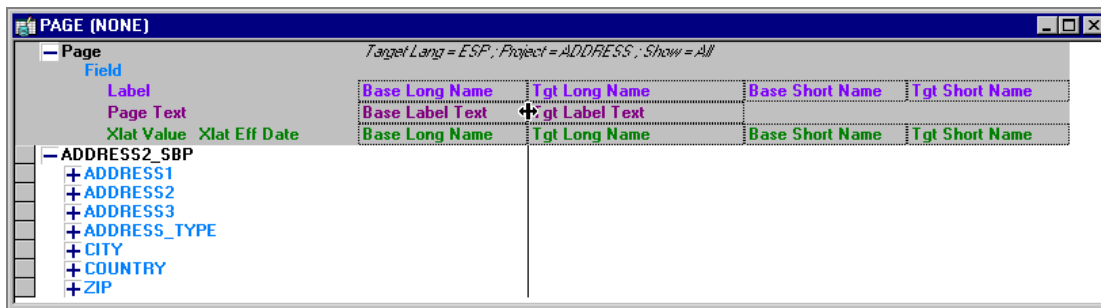
All	Shows all translatable objects,
Translated	Shows only the objects that have already been translated—that is, where the base language label and target language label do not match.
Untranslated	Shows only the objects that have not been translated—that is, where the target language label is blank or the base language label and target language label match.
Modified	Shows the objects that have been modified in the translation grid.
	The Translation Workbench defaults to this mode when it encounters a save conflict.

Keep in mind that the options you select in the pop up menu work together. If you select **Expand Current Object** while **Untranslated** is activated, you might not see anything. This simply means that everything in the current object is translated.

Resizing Columns and Rows

To change the column width, place the cursor over the column divider on the information bar. When the cursor changes to , drag the column divider to the desired position. Only translatable columns can be resized; you cannot resize the columns that display the hierarchical tree of page objects.

You can use the same technique to change row height.

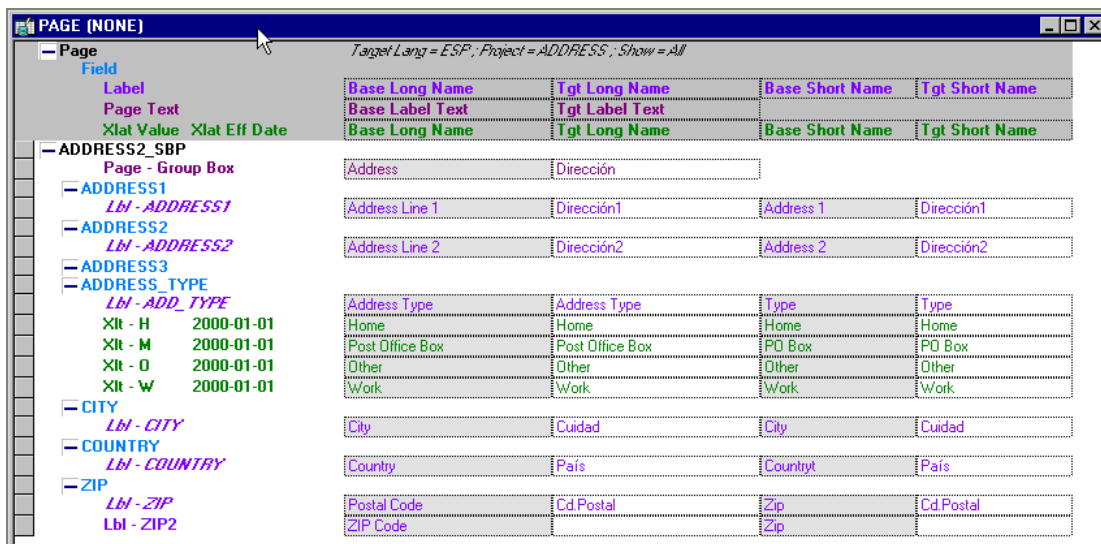


Resizing a column

Working in the Translation Grid

Entering Translations

The working area of the translation grid consists of four columns of data: Base Long Name, Tgt Long Name (target long name), Base Short Name, and Tgt Short Name (target short name).



Translation grid

As the column names imply, the two base language columns display the base language labels for the objects in the grid. Translating the labels is as easy as typing the translation in the **Tgt Long Name** and **Tgt Short Name** fields.

The grid incorporates standard text editing functionality. Cut, copy, and paste operations are available under the Edit menu. To force a line break within a label (so that the text wraps on the page), press CTRL-ENTER.

The first time that you translate a particular field label or xlat value, the same translation gets entered into any other occurrence of that field in the current grid. This ensures consistency, as well as being a time-saver.

The grid automatically limits the length of the text you enter, based on the maximum length of the fields. When you reach the maximum length, you cannot add any more characters.

As with any Application Designer object, changes are not permanent until you save them.



When related-language data is saved, the system requires values for all fields. You cannot partially translate a single PeopleTools object. For example, if you translate the long name but not the short name for a field, the short name defaults to the base-language short name, and this value is saved in the related language table. Similarly, if you translate some but not all xlat values for a particular field, any untranslated values pick up the base language text, which is then saved to the related language table.

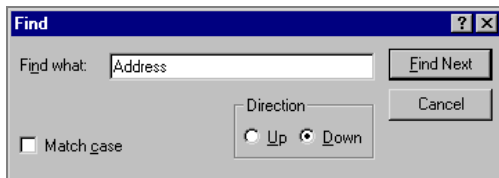
Navigating the Translation Grid

You can always click to move to a cell or node in the translation grid. However, you'll find that standard Windows keyboard controls are more efficient. The following tips will help you navigate in the translation grid:

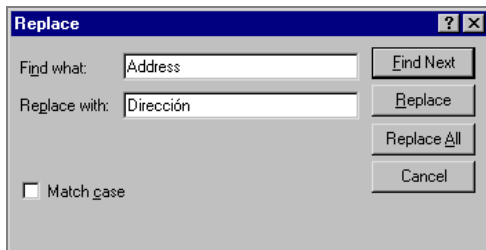
- Press TAB or ENTER to move to the right; press SHIFT+TAB to move to the left.
- Use any of the arrow keys to move from cell to cell. For example, the UP ARROW key moves you to the cell above the current cell.
- To move from the text cells to the tree, press the LEFT ARROW key. To move from the hierarchical tree to the text cells, you can always use TAB, but you can use the RIGHT ARROW key only when there are text cells in the same row—that is, if the current node is a translatable object.
- When moving around in the hierarchical tree, the UP ARROW and DOWN ARROW keys move you to the next item at that level or above—for example, from a field to an adjacent field or to a page. To move to a lower level, first press TAB, to move into the main translation grid, then move up or down to the desired node, and then use the LEFT ARROW key to move back into the tree.
- If there are several pages in your project, you can scroll directly to the node for a particular page by right-clicking that page in the project workspace and selecting Translate, Translate Page Objects from the pop-up menu.

Searching and Replacing

To search for a specific word within a single column, place the cursor anywhere in that column and select **Edit, Find and Replace In Current Column**. If you are searching a non-editable column, select **Edit, Find in Current Column**.



Find in current column



Find and replace in current column

The Find and Replace dialog boxes provide standard search and replace functionality, including the **Find Next**, **Replace**, and **Replace All** buttons. In either dialog box, select **Match case** if you want to enable case-sensitive searching.

The Find function gives you the option of whether to search **Up** or **Down** the column. When you reach the top or the bottom of the column, the search does not cycle back through the column. Therefore, if you want to search the entire column, be sure that you're in the top row when you press **Find Next**.

The Replace function always searches down the column. If you want to replace all occurrences in the column, be sure you're in the top row when you press **Replace**.

Using Application Designer Functionality

Because the translation grid is integrated into PeopleSoft Application Designer, you can use all standard Application Designer features while the translation grid is open. Some features that are particularly useful during translations are:

- Using **Edit, Find Object References** to research where in the system a particular object is used.
- Opening a translated page in order to realign translated objects.
- Opening objects to see the properties. Remember, as long as you're logged on using a non-base language, modifying labels in the object definition updates the related language tables, not

the base language tables.

There are several ways to open an object definition. Of course, you can use standard Application Designer functionality to open the object from the project window or the **File, Open** command. You can also open object definitions directly from the translation grid, either by double-clicking or right-clicking the object and selecting **View Definition** from the pop-up menu that appears. When you've selected page text that is not associated with a field definition, the page itself will open.

If there is a version of the page for the user's language preference, that version appears; if not, the base language page appears. Double-clicking in any row is equivalent to launching View Definition from the pop-up menu.



Access to different PeopleSoft Application Designer objects is controlled through security. A user who cannot open a particular type of object may not have the necessary level of access.

External Integration

Because the translation grid emulates a Microsoft Excel spreadsheet, you can use the standard Windows Copy and Paste tools to copy translations from the Translation Workbench into Excel or another Windows application. This is useful for performing operations such as a spelling check against your translations or for saving your translations in a spreadsheet for offline review.

To copy the entire contents of the currently open translation grid to the clipboard, first select the entire grid, either by selecting **Edit, Select All** or by clicking in the top left cell of the grid's border and selecting **Edit, Copy** from the menu.

With the Translation Workbench, you can copy an entire translation grid to the clipboard, but you can paste only individual field values back into the grid. You cannot paste the entire grid into the Translation Workbench after reviewing it in another application.

Using the Translate Pages

The Translation Workbench helps you translate fields and translate (xlat) values. Another set of utilities helps you translate other types of system objects.

Under **PeopleSoft, Translate, Use**, you'll find five pages to help you translate other system objects: menus, menu items, components, and messages.

To translate record descriptions with the Translate Records utility:

1. Start the Translate Records utility.

Select **PeopleTools, Translate, Use, Translate Records** to display the Translate Records page:

Translate Records page

The Translate Records page lets you select a set of record definitions and then translate the short and long descriptions for those records into a specific target language.


2. Select a target language.

Use the **Target Language** field to select the language into which to translate the record descriptions.

3. Use the **Record** field to select a set of record definitions.

Leave the **Record** field blank if you want to select all record definitions in the system.

If you want to work through the records alphabetically, type the first character(s) of the record name in the **Record** field.

If you want to translate a specific record, type the record name in the **Record** field or use the  **Search** button to prompt for a specific record.

4. Click the **Get Record** button to bring the record descriptions into the record list.

A set of record definitions appears in the **Record List**, based on the value in the **Record** field.

Translate Records

Selection Criteria

Record: Target Language:

Record List View All First 1 of 1 Last

Record Name / Base Description	Target Description
Record Name / Base Description Short Description: ABSENCE_HIST Long Description: EE Absence History Use ABSENCE_HIST to record information about an employee's absences from work. You can create multiple ABSENCE_HIST records for an employee.	<input type="text"/> <input type="text"/>

1 row selected.

Record list

- Translate the **Short Description** and **Long Description** into the target language.

The base-language descriptions for each record appear on the left; enter the translations in the fields on the right.

- Save the page.

To translate menus and menu group labels with the Translate Menus utility:

- Start the Translate Menus utility.

Select **PeopleTools, Translate, Use, Translate Menu** to display the Translate Menu page:

Translate Menu

Selection Criteria

Menu Name: Target Language:

Menu List View All First 1 of 1 Last

Translate Menus page

The Translate Menus page enables you to select a set of menu definitions and translate the labels for its menu and menu group into a specific target language.



For information about translating additional navigational labels—menu items and menu bar labels—see To translate menu bar and menu item labels with the Translate Menu Items utility.

2. Select a target language.

Use the **Target Language** field to select the language into which to translate the menu labels.

3. Use the **Menu Name** field to select a set of menu definitions.

Leave the **Menu Name** field blank if you want to select all the menus in the system.

If you want to work through the menus alphabetically, type the first character(s) of the menu name in the **Menu Name** field.

If you want to translate a specific menu, type the menu name in the field or use the **Search** button to prompt for a specific menu.

4. Click the **Get Menu** button to bring the menu descriptions into the Menu List.

A set of menu labels appears in the **Menu List**, based on the value in the **Menu Name** field.

Translate Menu

Selection Criteria

Menu Name: **Target Language:** **Get Menu**

Menu List View All First 1-4 of 4 Last

Menu Name:	ADMINISTER_HR_SECURITY	
Menu Label:	Administer HR &Security	<input type="text"/>
Menu Group:	Define Business &Rules	<input type="text"/>
Menu Name:	ADMINISTER_INV_SECURITY	
Menu Label:	Ad&minister &INV Security	<input type="text"/>
Menu Group:	Define Business &Rules	<input type="text"/>
Menu Name:	ADMINISTER_WORKFORCE_(GBL)	
Menu Label:	Administer Workforce (&GBL)	<input type="text"/>
Menu Group:	&Administer Workforce	<input type="text"/>
Menu Name:	ADMINISTER_WORKFORCE_(U.S.)	
Menu Label:	Administer Workforce (&U.S.)	<input type="text"/>
Menu Group:	&Administer Workforce	<input type="text"/>

4 rows selected.

Save

Menu list

5. Translate the **Menu Label** and **Menu Group** into the target language.

The base-language descriptions for each label appear on the left; enter the translations in the fields on the right.

If you want to include accelerator keys for use with Windows client navigation, position the accelerator key ampersand to the left of the character that will serve as the accelerator key for the menu or menu group.

6. Save the page.

To translate menu bar and menu item labels with the Translate Menu Items utility:

1. Start the Translate Menu Items utility.

Select **PeopleTools, Translate, Use, Translate MenuItem** to display the Translate Menu Items page:

Translate Menu Items page

The Translate Menu Items page enables you to select a set of menu definitions and then translate the labels for its menu bars and menu items into a specific target language.



For information about translating additional navigational levels—menu and menu group labels—see To translate menus and menu group labels with the Translate Menus utility.


2. Select a target language.

Use the **Target Language** field to select the language into which to translate the menu bar and menu item labels.

3. Use the **Menu Name** field to select a set of menu definitions.

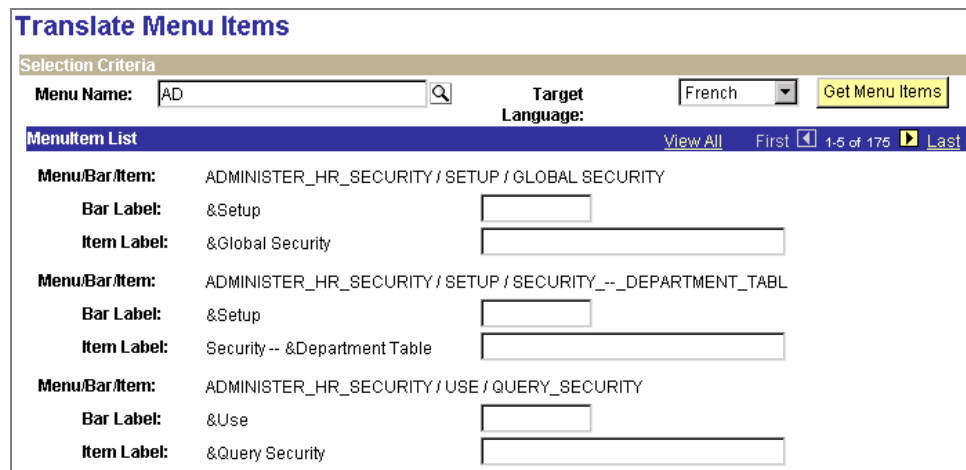
Leave the **Menu Name** field blank if you want to select all the menus in the system.

If you want to work through the menus alphabetically, type the first character(s) of the menu name in the **Menu Name** field.

If you want to translate a specific menu, type the menu name in the field or use the  **Search** button to prompt for a specific record.


- Click the **Get Menu Items** button to bring the menus into the MenuItem List.

A set of menu bar and item labels will appear in the **MenuItem List**, based on the value in the **Menu Name** field.



Translate Menu Items

Selection Criteria

Menu Name:  **Target Language:** **Get Menu Items**

MenuItem List View All First 1-6 of 175 Last

Menu/Bar/Item:	ADMINISTER_HR_SECURITY / SETUP / GLOBAL SECURITY	
Bar Label:	&Setup	<input type="text"/>
Item Label:	&Global Security	<input type="text"/>
Menu/Bar/Item:	ADMINISTER_HR_SECURITY / SETUP / SECURITY -- _DEPARTMENT_TABL	
Bar Label:	&Setup	<input type="text"/>
Item Label:	Security -- &Department Table	<input type="text"/>
Menu/Bar/Item:	ADMINISTER_HR_SECURITY / USE / QUERY_SECURITY	
Bar Label:	&Use	<input type="text"/>
Item Label:	&Query Security	<input type="text"/>

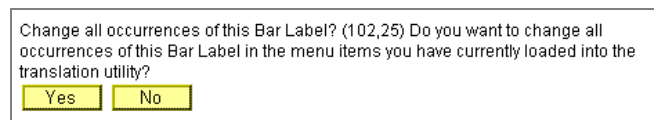
MenuItem list

- Translate the **Bar Label** and **Item Label** into the target language.

The base-language descriptions for each label appear on the left; enter the translations in the fields on the right.

If you want to include accelerator keys for use with Windows client navigation, position the accelerator key ampersand to the left of the character that will serve as the accelerator key for the menu or menu group.

After you translate a menu bar label and exit the field, you are asked if you want to use the same translation for all occurrences of this bar label:



Change all occurrences of this Bar Label? (102,25) Do you want to change all occurrences of this Bar Label in the menu items you have currently loaded into the translation utility?

Prompt following translation of menu bar label

If you click **Yes**, the translation you just entered is copied to each of the menu items in the list that share the same bar label. This is merely a data entry shortcut; no translations are saved until you explicitly save the page. Normally you click **Yes** in response to this prompt; you can click **No** if you want to leave some of the menu items untranslated.

The translated menu bar label will appear at runtime if:

- The user's language preference is set to the target language.
 - One or more menu items that appear in the menu bar have a menu bar label translated into the target language.
6. Save the Translate Menu Items page.

To translate folder tab labels:

1. Start the Translate Components utility.

Select **PeopleTools, Translate, Use, Translate Component** to display the Translate Components page:

Translate Component

Selection Criteria

Component Name: Target Language: English Get Component

Components List View All First 1 of 1 Last

Save

Translate Component page

The Translate Components page enables you to select a set of component definitions and then translate the labels for the pages that make up the component.

There are two translatable labels for each page in the component: an item label and a folder tab label.

In PeopleSoft Internet Architecture, only one of these labels is visible to the user. If a folder tab label exists, it is used on the folder tab for the corresponding page. The item label is ignored. If no folder tab label exists, the item label is used as the folder tab label.

In the Windows client, item labels become part of the navigational structure, so if the item label and folder tab label are different, they should both be translated. However, because folder tabs use the item label when the folder tab label is missing, it is common to have item labels only.


2. Select a target language.

Use the **Target Language** field to select the language into which to translate the component labels.

3. Use the **Component Name** edit box to select a set of component definitions.

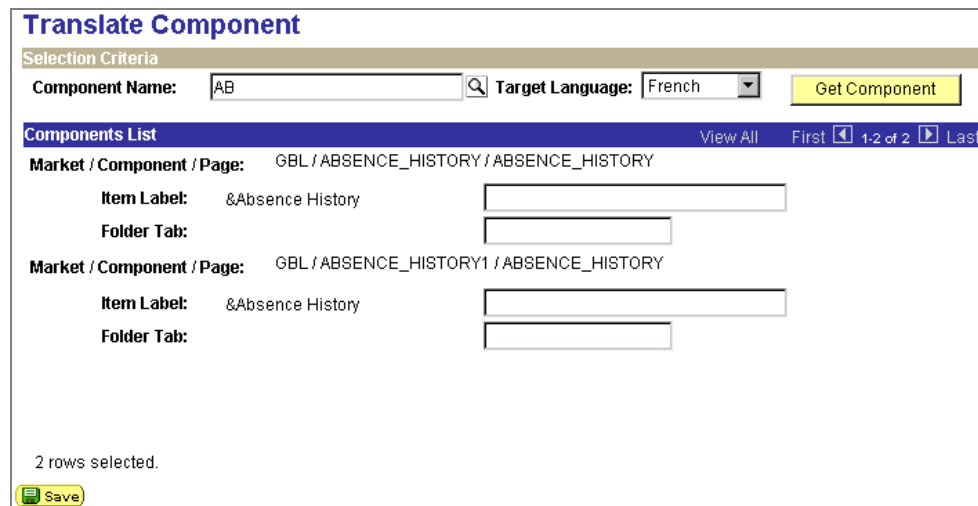
Leave the **Component Name** field blank if you want to select all the components in the system.

If you want to work through the components alphabetically, type the first character(s) of the component name in the **Component Name** field.

If you want to translate a specific components, type the component name into the edit box, or use the  **Search** button to prompt for a specific record.

4. Click the **Get Component** button to bring the components into the components list.

A set of components and their associated labels will appear in the **Components List**, based on the value in the **Component Name** field.



The screenshot shows the 'Translate Component' window. At the top, there's a 'Selection Criteria' section with a 'Component Name' field containing 'AB', a search icon, a 'Target Language' dropdown set to 'French', and a 'Get Component' button. Below this is the 'Components List' section, which has a header bar with 'View All', 'First', '1-2 of 2', and 'Last' buttons. The list contains two rows of data. Each row shows the 'Market / Component / Page' path, followed by 'Item Label' and 'Folder Tab' fields. The first row is for 'GBL / ABSENCE_HISTORY / ABSENCE_HISTORY' and the second for 'GBL / ABSENCE_HISTORY1 / ABSENCE_HISTORY'. Both rows show '&Absence History' as the item label. At the bottom left, it says '2 rows selected.' and there is a 'Save' button.

Components list

If a component item has no folder tab text, the item label appears in the folder tab. In these cases, you can leave the translation for the folder tab text blank.

5. Translate the **Item Label** and **Folder Tab** into the target language.

The base-language descriptions for each label appear on the left; enter the translations in the fields on the right.

If you want to include accelerator keys for use with Windows client navigation, position the accelerator key ampersand to the left of the character that will serve as the accelerator key for the menu or menu group.

6. Save the page.



To translate Message Catalog messages:



1. Start the Translate Messages utility.

Select **PeopleTools, Translate, Use, Translate Messages** to display the Translate Messages page:

Translate Messages

Selection Criteria

*Message Set Number:  Target Language:  **Get Messages**

Message List View All First  1 of 1  Last

Message Number: Base/Target Last Updated:

Base Message Text:

Base Explanation:

Target Message Text:

Target Explanation:

Save


Translate Messages page

The Translate Messages page enables you to select a message set from the Message Catalog and then translate the message set description, the messages, and the detailed explanations of the messages into a specific target language.

2. Select a target language.

Use the **Target Language** field to select the language into which to translate the message set.

3. Use the **Message Set Number** field to select a message set.

Type a message set number in the field, or use the  **Search** button to prompt for a specific message set number.

4. Click the **Get Messages** button to bring the messages into the message list.

A set of messages will appear in the Message List, based on the value in the **Message Set Number** field.

Translate Messages

Selection Criteria

Message Set Number: Target Language:

Message Set Description:

Message List View All First 1 of 7 Last

Message Number:	1	Base/Target Last Updated:	03/04/1998 9:53AM 11/10/2000 4:45PM
Base Message Text:	%1 has been updated by another user.		
Base Explanation:	Another user has modified an object you're editing. Their changes may conflict with yours. You must cancel your changes, then remake them after reviewing the modified object to ensure that your changes are compatible.		
Target Message Text:	<input type="text"/>		
Target Explanation:	<input type="text"/>		

7 rows selected.

Message list

- Translate the **Message Set Description** into the target language.

Click the **Update Message Set** button to display the Add Message Set page:

Add Message Set

Message Set Number: Language: Code:

Message Set Description:

Description:

Short Description:

Add Message Set page

Type a **Description** and **Short Description** for the message set in the target language; then click **OK** to accept the change and return to the Translate Messages page.

- Translate the **Message Text** and **Explanation** into the target language.

For each message in the message list, type translations for the messages and explanations in the **Target Message Text** and **Target Explanation** fields.

- Save the page.

To translate queries using the Translate Query utility:

- Start the Translate Query utility.

Select **PeopleTools**, **Translate**, **Use**, **Translate Query** to display the Translate Query page:

Translate Query

Selection Criteria

Query Name: Target Language:

Query List View All First 1 of 1 Last

Query Name / Base Description	Target Description
Short Description	
Long Description	

Translate Query page

The Translate Query page enables you to translate query descriptions, heading labels, and query prompt descriptions into a specific target language.

2. Select a target language.

Use the **Target Language** field to select the language into which to translate the query descriptions and labels.

3. Use the **Query Name** field to select a query.

Leave the **Query Name** field blank if you want to select all the queries in the system.

If you want to work through the fields alphabetically, type the first character(s) of the query name in the **Query Name** field.

If you want to translate a specific query, type the query name in the field, or use the **Search** button to prompt for a specific record.

4. Click the **Get Query** button to bring the query descriptions into the query list.

A set of query descriptions will appear in the **Query List**, based on the value in the **Query Name** field.

Translate Query

Selection Criteria

Query Name: Target Language:

Query List View All First 1 of 20 Last

Query Name / Base Description	Target Description
i PER006__EE_MAILING_LABELS Short Description PER006--EE mailing labels Long Description	<input type="text"/> <input type="text"/>

20 rows selected.

Query list

- Translate the base-language descriptions into the target language.

The base-language descriptions appear on the left; enter the translations in the fields on the right.

- Translate query field headings and prompt names.

Click  the **Show Item Details** button to display the Query Headings page:

Query Headings

Query Field Headings View All First 1-3 of 4 Last

Query Field Name	Field Heading
Department Name	Department Name
Effective Date	Effective Date
GL# Expense	GL# Expense

Query Prompts View All First 1 of 1 Last

Query Prompt Name	Prompt Text
-------------------	-------------

Query Headings page

By default, field headings in queries are derived from RFT Long or RFT Short field descriptions, which means that they are automatically language-sensitive (provided that the field descriptions have been translated). Query prompt names are similarly derived from the prompt table description. The Query Headings page displays any field headings labels and prompt descriptions that are *not* derived from field or table descriptions (that is, descriptions that have override text in the query definition).

Translate the description text for the field headings and prompt descriptions in the fields, and then click **OK** to accept the change and return to the Translate Query page.

7. Save the page.

Checking for Override Text in Pages

Most page text is derived from language-sensitive field descriptions. Text derived from field descriptions is translated automatically when you clone the base-language page definition (provided that the field descriptions have already been translated). However, page control definitions give you the option to override the field description with a text description.

If you use the Translation Workbench to translate pages, you can translate all such override text right along with labels that are derived from field descriptions. This feature of the Translation Workbench helps to ensure that the entire page gets translated.

PeopleSoft also provides a Page Text inquiry page that identifies override text in pages. You can translate the override text in the target-language page definition in PeopleSoft Application Designer.



For more information, see Globalizing Pages.


To check for override text in pages:


1. Start the Page Text utility.

Select PeopleTools, Translate, Inquire, Page Text to display the Page Text page:

Page Text


Selection Criteria

Page Name: 

*Language Code: 

[Get Page Text](#)

Num	Field Type	Record.Field	Label Text

 Save


Page Text page

The Page Text page lets you generate a list of all override text (that is, text that is not derived from field descriptions) in a set of page definitions.

2. Use the **Page Name** field to select a set of page names.

Leave the **Page Name** field blank if you want to select all the pages in the system.

If you want to work through the pages alphabetically, type the first character(s) of the page name in the **Page Name** field.

If you want to translate a specific page, type the page name in the field, or use the  **Search** button to prompt for a specific record.

3. Select the language code of the page set.

Use the **Language Code** field to select the language code of the page set. For example, if the **Page Name** field contains *B*, and the **Language Code** field contains *French*, the page set will consist of page definitions that begin with *B* and whose language code is *FRA*.

4. Click the **Get Page Text** button to display the overridden page text.

You will see a set of page definitions based on the settings in the **Page Name** and **Language Code** fields:

For each page that appears in the list, the **Label Text** is the override text on the page. Looking at the label text, you can tell whether it has been translated. If it hasn't been translated, you need to use the Translation Workbench to do the translation.



For more information about Translation Workbench, see [Using the Translation Workbench](#).

- 5. Save the page.**

CHAPTER 8

Translating Windows Resources

Strings referenced in the PeopleTools Windows client and application server are stored in Microsoft Windows resources and can include strings, dialog boxes, icons, cursors, and bitmaps. These Windows resource files (.RC and .RCX files) are compiled into dynamic link libraries (DLLs).

The Windows resources are distinct from PeopleTools objects, which are defined in PeopleTools database tables. Windows resources are used in PeopleTools applications and in the PeopleTools runtime environment (the common environment in which PeopleSoft applications are run).

Typically, PeopleSoft provides pre-translated Windows Resources on the PeopleTools CD-ROM, and no customization is required. However, if you are planning to translate selected PeopleTools objects into a new language, or if you want to modify the PeopleSoft-provided translation, this chapter provides you with the steps that are necessary to translate and compile these resources.

You can translate Windows resources by modifying the resource source files in a resource editor and then compiling the modified resources into an *alternate language DLL*. PeopleSoft supplies batch files to make it easier to compile these DLLs, using tools that are supplied with Microsoft Visual C++.

PeopleTools provides a full set of Windows Resource source files for all of our PeopleTools on the PeopleTools CD-ROM; however, these source files are loaded only if you choose to install the Language Development Kit during your installation. You can find them on your file server in the <PS_HOME>/SRC/RES directory. The translated, compiled resource DLLs are loaded from the PeopleTools CD-ROM when you select the Language Pack option during CD installation.



For more information, see the PeopleSoft Installation and Administration Guide for your platform.

You always need alternate language resources on application servers. Windows clients in a two- or three-tier installation also need alternate language resources on the client workstations.

A user's language preference determines which alternate language DLLs are used by the system at runtime and thus controls the language in which Windows resources are displayed. The language preference is established in different ways, depending on whether the user is using the PeopleSoft Internet Architecture or the Windows client.

<i>Environment</i>	<i>Language Preference for Windows Resources</i>
PeopleSoft Internet Architecture	Signon language

Environment	Language Preference for Windows Resources
Windows client	Configuration Manager language preference



For more information about language preference settings, see Controlling International Preferences.

Before You Begin

Before you begin the translation process, make sure that you have the required files and software tools.

Required Files

You can find these resources in the high-level PeopleTools directory under \SRC\RES. If this directory is missing from your PeopleTools installation, check to make sure that the optional Language Pack has been installed as described above.

PeopleSoft provides all the English resource files—and selected local language resources—as well as supporting header files that you need to produce your own alternate language DLLs. PeopleSoft also provides batch files to automate the resource compilation process.

Required Tools

The tools that you need to customize Windows resources are part of Microsoft's Visual C++ compiler package. You won't have to change or compile any C++ code, but you will need to use the resource compiler and linker; if you are translating any resources other than strings, you will need to use the Microsoft Developer Studio. See the *PeopleSoft Hardware and Software Requirements* guide for the exact version of Visual C++ supported.

Understanding Alternate Language DLLs

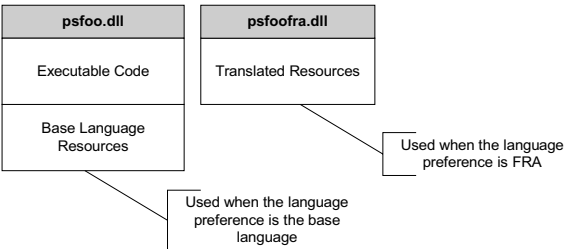
An alternate language DLL is a file that stores translated resources for an associated PeopleTools DLL or EXE file (module). At runtime the system uses the resources in the alternate language DLLs that correspond to the user's language preference.

PeopleSoft uses a naming convention that includes the name of the base executable (the executable with which the alternate language DLL is associated) and the language code that identifies the language of the alternate language DLL. An alternate language DLL is named:

PSZZZZXX.DLL

Where *ZZZ* is a 3-character code that identifies the PeopleTools module with which the alternate language DLL is associated, and *XXX* is a 3-character code that identifies the language of the alternate language DLL.

The alternate language DLL is used only to store translated resources. All code is executed from the base-language executable. The following diagram illustrates the structure of a base-language module, PSFOO.DLL, and its French alternate language, DLL, PSFOOFRA.DLL.



Base language DLL and alternate language DLL

The system uses the resources that are stored in the alternate language DLL that matches the user’s language preference, if the DLL exists. Otherwise, it uses the resources that are stored in the base executable file.

Locating Resource Directories

Alternate language DLLs must be present on all application servers. Windows clients in a three-tier installation also must have the alternate language DLLs present.

Windows NT application servers and Windows client machines use Windows resources from the base language and alternate language DLLs. UNIX application servers, however, store resources for all languages and modules in a single file, PSAPPSERV.RES. This file contains the same resources as the alternate language DLLs.



For more information about translating resources to a UNIX application server, see Using MAKEUNIX.BAT.

The following table shows the content of the directories that are used to support translated resources. These directories are distributed with PeopleTools and contain user-customizable resource files and other supporting files that are needed to compile, bind, name, and copy alternate language resource DLLs.

Directories and Files	Purpose
\SRC\RES	The root directory for alternate language resource DLLs. The files in this directory include some batch files that are used in constructing alternate language DLLs.

Directories and Files	Purpose
<code>\SRC\RES\ENG</code>	A prototypical alternate language development directory for the English language. There is one of these directories for each alternate language. This directory should be copied for each new alternate language that you create.
<code>\SRC\RES\ENG\INC</code>	Holds header (.H) files that are common to several DLLs and various icons and bitmaps.
<code>\SRC\RES\ENG\xxxxx</code>	<p>Within this directory, there is a directory for most PeopleTools DLL and EXE files found in <code>\SRC\BIN</code>. These directories contain the following file types:</p> <ul style="list-style-type: none"> *.RCX - string table resource files *.RC - menu and dialog box resource files *.CUR - cursors *.H - header files containing resource identifiers *.BMP - bitmap files used for icons and other graphical components *.ICO - icon files for Windows Explorer

Translating Resource Files

To make alternate language DLLs for PeopleTools, follow the procedure below, which outlines the process for translating the English PeopleTools resource files to another language (French, in this example).



Keep in mind that PeopleSoft provides pre-translated PeopleTools alternate language DLLs for many languages, including French, Spanish, Dutch, Portuguese, Japanese, German, and Canadian French. You need to follow these steps only if you want to translate PeopleTools into a language that is not provided by PeopleSoft or if you want to modify one of the translations that is provided.

To translate English resource files:

1. Determine the three-letter designation of the new language.

In this example, we'll use the FRA designation for French. For consistency in referring to languages throughout the system, use one of the standard code values stored in the Translate table for the LANGUAGE_CD field.

2. Copy the contents of `\SRC\RES\ENG` to the target resource directory.

For French, for example, the target resource directory is `\SRC\RES\FRA`.

3. Use Microsoft Developer Studio to convert the English resources to the alternate language.

A text editor is adequate to change string resources. The resource editor enables you to size and position elements in dialog boxes and other graphic components of the resource files.

4. Set up your environment variables.

Before compiling your translated resource files, you need to set the appropriate environment variables for the C++ resource compiler. Typically, Microsoft Visual C++ provides a batch file, VCVARS32.BAT which sets the variables for you. In addition to the variables set in this batch file, you should set the environment variable TOOLBIN to the <PS_HOME>\BIN\CLIENT\WINX86 directory of your file server, and append the SRC\RES\ENG\INC directory to the environment variable INCLUDE.

5. Compile alternate language DLLs using MAKEALTL.BAT.



For more information, see Using MAKEALTL.BAT.

6. Copy the resultant alternate language DLLs to the BIN\SERVER\WINX86 directory of your NT application server (NT application servers only).

7. Compile UNIX application server resources and transfer to the UNIX application server (UNIX application servers only).

If you are using a UNIX application server, you must run MAKEUNIX.BAT to compile the resource files into a format that is readable by the UNIX application server.



For more information, see Using MAKEUNIX.BAT.



If you have changed any menu item text, you have to update the Security settings to give users access to the menu items.

Using MAKEALTL.BAT

The MAKEALTL.BAT file calls the Microsoft Visual C++ Resource Compiler and Linker to compile your translated resources into a Windows Dynamic Link Library (DLL). Before running MAKEALTL.BAT, you must make sure that the environment variables required by the Resource Compiler and Linker are set correctly in your current DOS window. These environment variables are as follows.

<i>Environment Variable</i>	<i>Description</i>	<i>Example</i>
%path%	Must include your	C:\MSDEV\BIN

Environment Variable	Description	Example
	Microsoft Visual C++ executables directory.	
%include%	Must include your Microsoft Visual C++ INC directory and the Microsoft Foundation Classes (MFC) BIN directory. It must also include a copy of the PeopleTools resource include directory.	C:\MSDEV\INC C:\MSDEV\MFC\INC C:\PT810\SRC\RES\ENG\INC
%lib%	Must include your Microsoft Visual C++ library directory.	C:\MSDEV\LIB
%toolbin%	Must point to the directory containing your PeopleTools executables.	C:\PT810\BIN\CLIENT\WINX86

Once you have verified that the variables in the table above are correct, you can run MAKEALTL.BAT.

To run MAKEALTL.BAT:

1. In a DOS window, go to the SRC\RES directory.
2. Run the MAKEALTL batch file with <language> and <PeopleTools module> parameters.

This process compiles the resources into language-specific DLLs and copies the files to your PeopleTools executables directory.

The syntax of the command is:

```
MAKEALTL <language> <PeopleTools module>
```

<language> is the three-letter code for the language that you are compiling (such as FRA).

<PeopleTools module> is the name of the directory that contains the resource files that you have translated (such as PSSYS). Pass *ALL* in place of the directory name to compile all PeopleTools modules in the language that you specified.

Using MAKEUNIX.BAT

MAKEUNIX.BAT compiles your translated resources into a single file that is readable by the PeopleSoft UNIX application servers. It reads the same translated resources as MAKEALTL.BAT, but instead of producing a separate DLL for each language/module

combination, MAKEUNIX.BAT produces a single indexed resource file, called PSAPPSRV.RES.

To run MAKEUNIX.BAT, you must have the Microsoft Visual C++ development environment installed on your workstation. MAKEALTL.BAT calls the C++ compiler to parse the resource files from all languages and to create PSAPPSRV.RES. In addition to the environment variables that are required to run MAKEALTL.BAT, MAKEUNIX.BAT requires that the %PSVER% environment variable be set. This variable should be set to the base directory in which you installed PeopleTools (C:\PT810, in the preceding example).

If you have added any new languages as part of your PeopleTools translations, you have to edit MAKEUNIX.BAT to include the new languages so that they will be included in the PSAPPSRV.RES file.

To run MAKEUNIX.BAT:

1. Open a DOS command prompt window.
2. Change the directory to your \PT810\RES directory.
3. Run the MAKEUNIX batch file.

MAKEUNIX doesn't require that you pass the module or language name arguments; it compiles the UNIX application server resources for all languages and modules in one pass.

The output from MAKEUNIX.BAT, PSAPPSRV.RES, will be placed in \PT810\APPSERV\UNIX. You must transfer PSAPPSRV.RES to the BIN directory of your UNIX Application Server, using a network file transfer utility such as FTP.

Modifying Terminology

A specialized terminology management tool enables you to streamline certain language changes through the efficient use of text Search and Replace processing. This is particularly relevant to the task of adapting a language that is based on another language. For example, you can implement Mexican Spanish by adapting delivered Spanish translations using Search and Replace to substitute Mexican Spanish terms where appropriate.



In order to adapt an existing language to a new language, you must first define the new language in the PeopleSoft system. For more information about defining new languages, see *Creating New Languages*.

Even if you're not creating an entirely new language translation, you can use this functionality to adapt terminology according to your organizational standards. Does your organization use the words *Cost Center* instead of *Department*? You can now make this substitution throughout the database.

Terminology Management Overview

Searching and replacing terms is a multistep process, with several opportunities to back out the changes. The process breaks down into three phases: searching, replacing, and undoing. You can always use the search functionality on its own; it is entirely possible that you'll never use the replace or undo process.

Each phase—search, replace, and undo—consists of three separate steps: setting up the process, running the process, and reviewing the results.

Searching

Step	Description
Setting up the Search	On the Define Search Criteria page, specify which language(s) to search, which database objects to search, and what term to search for. If you want to replace the term with a new term, also specify the replacement text.
Processing the Search	Using PeopleSoft Process Scheduler, run the Search process.

Reviewing the Results	<p>You review the results in one of two places, depending on whether replace processing is enabled. Replace processing can be disabled for specific searches or for specific users. For example, you may be considering replacing terminology, and you may want to determine the potential exposure of such a change. Defining a Search with Replacement disabled, will ensure that there is no risk of accidental text replacement.</p> <p>If replace processing is enabled, review the results in the Search Results/Replace pages. If you don't want to replace the text, you can mark the process as complete.</p> <p>If replace processing is disabled, review the results on the Search Only Results page.</p>
-----------------------	--

Replacing

Step	Description
Setting up the Replace	<p>If your original search criteria specified replacement text, you'll see both the original text and the replacement text on the Search Results/Replace page. At this point, nothing has actually been replaced.</p> <p>Accept, reject, or modify the replacement text for each search result.</p>
Processing the Replace	Using PeopleSoft Process Scheduler, run the Replace process.
Reviewing the Replace	<p>Go to the Replace Results/Undo page to see the list of results, along with any replacement text. This page displays all results from the original search and indicates whether you replaced text.</p> <p>At this point, the replacement has happened, but you can still back out the changes by continuing to the Undo step.</p> <p>If you don't want to undo any of the replacements, you can mark the process as complete.</p>

Undoing

Step	Description
Setting up the Undo	As you review your replace results on the Replace Results/Undo page, specify whether to undo any of the replacements.
Processing the Undo	Using PeopleSoft Process Scheduler, run the Undo process.

Reviewing the Undo	<p>Go to the Inquiry page that displays all search results to see the final results of all your searching, replacing, and undoing.</p> <p>This page displays all results from the original search, regardless of whether you replaced the term or used Undo for any replacement.</p> <p>Although the search and replace is final at this point, this record of what you did is permanent. If you ever need to back out the changes, you can use this information to research what replacements took place.</p>
--------------------	--

Understanding Search Statuses

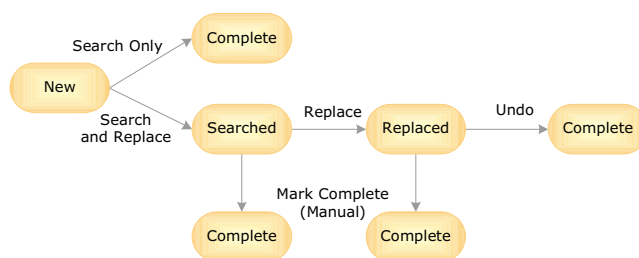
As you proceed through the search and replace process, you'll notice that all of the pages display a Status field. This status is updated automatically to reflect where you are in the process.

The default status for all new searches is *New*.

If you configure the search type to disable replacing (that is, if you choose the search-only option when you set up your search criteria), then no further action is possible after the search. Therefore, after the search, the status is updated to *Complete*. Otherwise, the status is updated to *Searched*.

Running the Replace process updates the status to *Replaced*, and running the Undo process updates the status to *Complete*.

If you don't want to run the full search-replace-undo process, you can manually mark the status *Complete* after searching or replacing. Once the status is *Complete*, you cannot change your mind. If you decide to make further terminology changes, you'll need to re-create the search criteria using the Copy Search Criteria feature on the Search Criteria Inquiry page.



Search statuses

Granting Search Privileges

Use the User Search Options page to grant search and view access to your users

To set user search options:

1. Select PeopleTools, Translate, Use, User Search Options.
2. Search for an existing user ID or add a new one.

Adding a user ID does not mean that you are creating new users; it means that you are adding a user to the list of those who have permission to use the search and replace functionality.

Add a user ID to grant access to someone who does not already have access. Search for an existing user ID to modify permissions for someone who already has access.

Use the standard search or add method to enter the user ID and access the User Defaults page.

User Defaults

User ID: PTDOCAG

Permitted Search Action

☐ Search Only

☒ Search, Replace and Undo

☒ View Other User's Activity

Save Return to Search Add Update/Display

User Search Options - User Defaults page

3. Set the user's search permissions.

Select **Search Only** to grant the user permission to search, but not to replace or undo. When a user with this level of access defines search criteria, the Search Only (No Replace) option is selected automatically.

Select **Search, Replace and Undo** to grant the user full access to all aspects of the PeopleSoft terminology management functionality.

4. Choose which searches the user can access.

Select **View Other User's Activity** to grant the user permission to see searches created by all users.

Leave this check box clear to deny access to all searches other than the user's own searches.

Only users with **Search, Replace and Undo** access can be granted access to other users' activity.

5. Save the page.

Searching

Defining Search Criteria

The first step in a search and replace process is to define your search criteria. To define your search, select **PeopleTools, Translate, Use, Define Search Criteria**.



In addition to using the standard add method to create a new search, you can copy an existing search and then modify the definition as necessary.



For more information about copying an existing search, see [Reviewing Search Criteria](#).

Define Search Criteria - Search Profile page



The Search Profile page displays only searches where the status is **New**. This means that you can edit the search criteria only while the status is **New**; you cannot edit the search criteria after running the search.



For more information about search statuses, see [Understanding Search Statuses](#).

To enable search processing:

1. Select **Search Ready**.

Selecting **Search Ready** enables the search processing to pick up this search. You can leave this check box clear until you're ready to run your search, but you must select it before you can search.

To choose which languages to search:

1. Enter the **Search Lang** (search language).

The **Base Lang** (base language) displays the base language of the database. This is a display-only field and cannot be edited.

The **Search Lang** determines which language tables are searched. If you select the base language as your search language, you can search and replace data on the base language tables only. If you select a non-base language as your search language, your basis for searching can be dependent on either the base language or the search language or both. However, you can still replace data only in the search language tables.

2. If the search language is different from the base language, select the basis for your search.

Select one of the following options:

Base Language

You can specify search text only for the base language tables. The system will search the base language tables, but the search returns results only when there's a match in the base language tables *and* a corresponding translation in the search language. That is, the search process will ignore occurrences of the word in the base language tables if there is no corresponding translation on the related language table.

Non-Base Language

You can specify search text only for the related language tables. The system will search the related language tables for the text that you've specified and return the results.

Both

You can specify search text for both the base language tables and the related language tables. In this case, the results returned will be only those instances where the search criteria for both the base language and related language have been met.

To disable the replace process:

1. Select **Search Only (No Replace)**.

Selecting **Search Only (No Replace)** ensures that the criteria will be used only for searching, not for searching and replacing. By selecting this option, you prevent the system from doing anything further after you run the search.

This setting also determines whether the status is set to **Complete** or **Searched** after you search. If you select **Search Only (No Replace)**, then no further action is possible after the search. Therefore, after the search, the status is updated to **Complete**. Otherwise, the status is updated to **Searched**.

To define your search scope:

1. Specify whether to search long text fields.

Select **Search Long Fields** to include the long description (Descr Long) field in your searches. This setting applies to any type of object that you're searching.

2. Specify which groups of database objects to search.

Searchable text exists in many parts of the database: fields, queries, menus, and even your application data. Before you search, you need to choose which types of database objects you want to search.

Use the **Search Objects** portion of the page to specify which types of objects are searched. Each check box represents a group of searchable objects. Select the groups that you want to search.



With certain PeopleSoft Application Designer objects, the translatable text shows up only in Application Designer dialog boxes and on property sheets. For example, you see page descriptions in the Application Designer Open dialog box, but end users don't see the descriptions when they look at the actual page. The important exception to this situation are fields, translate values, page text, and menus. The descriptions for these four types of objects are visible to end users.

Select one or more of these groups:

Records, Fields, Xlats	This includes record descriptions, field labels, and translate (xlat) values.
Business Processes	This includes the descriptions for Activities and Business Process Maps and the labels of the icons on those maps.
Queries and Strings	This includes the descriptions for queries and strings.
Pages	This includes descriptions of pages and page text that does not come from a field definition or translate value.

Menus

This includes all levels of navigation, including the component name.

Some menu descriptions include an extra ampersand to designate a hot key. For example, a menu that looks like *Administer Workforce* is actually stored in the system as *Administer &Workforce*. Most likely you would prefer to disregard the ampersand when you search; to do this, select **Exclude & from Menu Search**.

Other System Objects

This includes all other PeopleTools objects, such as application messages, business components, and process definitions.

Other Application Objects


This includes translatable application data—application tables with related language tables. There doesn't have to be a translation in your search language; the existence of the related language table is enough to qualify a table for inclusion.

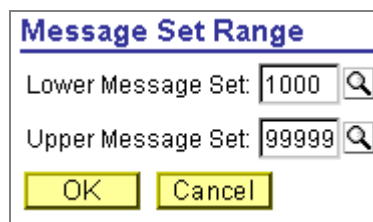
PeopleSoft provides a PeopleSoft Data Mover script, TSRECPOP.dms, that creates this list of translatable application tables. If you've customized the system with new translatable tables, you must run this script to re-create the table.

Messages


This includes both the messages and the longer explanatory text that is associated with each message.


When searching Messages, you can select the range of message to search. In PeopleSoft, message sets 1–999 are reserved for Tools Messages. Messages sets 1000–9999 are used by the applications. Selecting either **Application Messages** or **PeopleTools Messages** will narrow your search to the selected numeric range. You can also search **All Messages**.

Alternatively, you can narrow your search to a specific range of messages. To do this, select **All Messages**, and then click the  button to open the page where you can enter the desired message set range:



 To select all of the objects, click the arrow that curves to the right.

 To deselect all of the objects, click the arrow that curves to the left.

 To see a list of the tables that are associated with a group of objects, click the **Information** button adjacent to the group in question.

Text Search Records			
		View All	First 1-4 of 4 Last
PSDBFIELD	Database Field	PSDBFIELDLANG	Database Field(Alt Language)
PSDBFLDLABL	DBField Label Table	PSDBFLDLABLLANG	DBField Label Language Table
PSRECDEFN	Record Definition	PSRECDEFNLANG	Record Definition(Alt Lang)
XLATABLE	Translate Value	XLATABLE	Translate Value
Return			

Text search records for records, fields, and xlat objects.



To see the translatable fields in each of these records, look at the record on the Define Text Search Records page. For more information, see *Modifying Searchable Objects*.

To define your search text:

1. Enter the search and text for the base language or the search language or both.

There are three columns for search text. Different choices are available, depending on which language(s) you've selected to search and whether you've allowed replacing.

Search Text - Base	This option is available if you're searching the base language or both the base and non-base languages.
Search Text - Non Base	This option is available if you're searching the non-base language or both the base language and the non-base language.
Replacement Text	This option is available unless you selected Search Only (No Replace) in the top portion of this page.

For each language that you're searching, enter the text you want to search for. The system will search only the appropriate language tables.

Note that you can enter multiple rows in each column. This is useful if you want to search for several different terms, or if you want to search for several forms of the same word. For example, if you want to search for *mouse* and *mice* in the same search, simply create two search rows and use one for each word.

2. Make sure that your search text is in the proper case.

Searching is case sensitive. When you want to search for all occurrences of a word, regardless of case, create multiple search rows, and enter the word multiple times—once for each variation.

Use the three *abc* buttons to automatically put the text in the specified case. **ABC** capitalizes the text, **abc** lowercases the text, and **Abc** puts the text in title case (the first letter of each word is capitalized).

These buttons modify all text in the associated row. So if you are not using the same case in all three columns, take care not to inadvertently reset the case in one column when intentionally resetting the case in another.

3. Choose the matching method for the search.

Directly underneath each search text field, select one of the following matching methods to use when searching for that text:

Exact Text Match	Returns any instance where the text constitutes the entire contents of the field. For example, if you perform an exact text search on <i>Department</i> , your results won't include fields that contain <i>Department Code</i> .
Full Word Match	Returns all occurrences of the text string where the string is not embedded in another word. In this case, searching for <i>Department</i> will return occurrences of <i>Department Code</i> , but will not return occurrences of <i>Departments</i> or <i>Departmentalize</i> .
Like Text Match	Returns all occurrences of the text string

Cloning Existing Search Criteria

If you want to set up multiple similar searches, you can expedite data entry if you start by cloning an existing search. This enables you to start with most of your criteria already in place.

To clone a search definition:

1. Select PeopleTools, Translate, Inquire, Search Criteria.

You are prompted to identify a search.

Search Criteria

Find an Existing Value

User ID:

Text Search Identifier:

Search Status:

[Basic Search](#)

Find an Existing Value page

2. Enter the identifiers for the search definition.

Use the standard search method to access the Search Criteria page.

Search Criteria

User ID: PTDMO Identifier: EXAMPLE *Status: Complete

Base Lang: English ☒ Base Language ☐ Search Only (No Replace)

Search Lang: English ☒ Non-Base Language ☐ Search Long Fields

☒ Both

Search Objects

☒ Records, Fields, Xlats ☒ Menus ☒ Messages

☒ Business Processes ☒ Exclude & from Menu Search 1000 99999

☒ Queries and Strings ☒ Other System Objects

☒ Pages ☒ Application Objects

View All First 1 of 1 Last

Search Text - Base	Search Text - Non Base	Replacement Text
Department Full Word Match		Cost Center

Search Criteria page

3. Click the **Copy Search Criteria** button.

You are prompted for a unique search identifier.

Enter New Search Identifier

Text Search Identifier:

Enter New Search Identifier page

4. Enter a **Text Search Identifier** that will be the unique name for the new search.
5. Click **OK**.

The Search Profile page appears, and the cloned definition is saved. You can now modify the search criteria as necessary.

Sample Search Criteria

Searching Base Language Text

Search Profile

User ID: PTDMO Identifier: EXAMPLE2 *Status: New

Base Lang: English ☐ Base Language ☐ Search Ready

Search Lang: English ☐ Non-Base Language ☐ Search Only (No Replace)

☐ Both ☐ Search Long Fields

Search Objects

☒ Records, Fields, Xlats ☒ Menus ☒ Messages

☒ Business Processes ☒ Exclude & from Menu Search ☐ Application Messages

☒ Queries and Strings ☒ Other System Objects ☐ PeopleTools Messages

☒ Pages ☒ Application Objects ☐ All Messages

View All First 1 of 1 Last

Search Text - Base	Search Text - Non Base	Replacement Text
ABC abc Abc Department Full Word Match		Cost Center

Save Return to Search Add Update/Display

Base language search and replace

This illustrates a search and replace on the base tables only. This will be a search and replace because the **Search Only (No Replace)** check box is clear. Because the Search Lang matches the Base Lang, the **Search Text - Base** and **Replacement Text** sections of the page are enterable.

Notice that all of the check boxes are selected in the **Search Objects** area of this page. That means that the system will search all of the available search objects for the word *Department*. When the search results are returned, you will have the option to replace each instance of the word *Department* with the term *Cost Center*. The text replacement occurs in the base language tables.

Searching for Base Language Text With Existing Translations

Search Profile

User ID: PTDMO Identifier: EXAMPLE2 *Status: New

Base Lang: English ☐ Base Language ☐ Search Ready

Search Lang: Spanish ☐ Non-Base Language ☒ Search Only (No Replace)

☐ Both ☐ Search Long Fields

Search Objects

- ☒ Records, Fields, Xlats
- ☒ Business Processes
- ☒ Queries and Strings
- ☒ Pages
- ☒ Menus
- ☒ Exclude & from Menu Search
- ☒ Other System Objects
- ☒ Application Objects
- ☒ Messages
- ☐ Application Messages
- ☐ PeopleTools Messages
- ☐ All Messages

View All First 1 of 1 Last

ABC	abc	Abc	Search Text - Base	Search Text - Non Base	Replacement Text
			Product		
			Full Word Match		

Save Return to Search Add Update/Display

Base language search for objects with translations

In this case the system will search for the word *Product* in the base language tables and show all occurrences where *Product* has been translated into Spanish. You might do this as a preliminary effort, to see the different ways in which the word *Product* has been translated.

Searching Non-Base Language Text

Search Profile

User ID: PTDMO Identifier: EXAMPLE2 *Status: New

Base Lang: English ☐ Base Language ☐ Search Ready

Search Lang: Spanish ☒ Non-Base Language ☒ Search Only (No Replace)

☐ Both ☐ Search Long Fields

Search Objects

- ☒ Records, Fields, Xlats
- ☒ Business Processes
- ☒ Queries and Strings
- ☒ Pages
- ☒ Menus
- ☒ Exclude & from Menu Search
- ☒ Other System Objects
- ☒ Application Objects
- ☒ Messages
- ☐ Application Messages
- ☐ PeopleTools Messages
- ☐ All Messages

View All First 1 of 1 Last

ABC	abc	Abc	Search Text - Base	Search Text - Non Base	Replacement Text
				Producto	
				Full Word Match	

Save Return to Search Add Update/Display

Searching the non-base language

In this case the system searches for all instances of the word *Producto* in the Spanish language tables. Again, you might do this type of search in order to see the number of occurrences in the database. The result would show all occurrences where English text has a translation with the word *Producto*.

Replacing Non-Base Language Text

Search Profile

User ID: PTDMO Identifier: EXAMPLE2 *Status: New

Base Lang: English ☐ Base Language ☐ Search Ready

Search Lang: Spanish ☒ Non-Base Language ☐ Search Only (No Replace)

☐ Both ☐ Search Long Fields

Search Objects

☒ Records, Fields, Xlats ☒ Menus ☒ Messages

☒ Business Processes ☒ Exclude & from Menu Search ☐ Application Messages

☒ Queries and Strings ☒ Other System Objects ☐ PeopleTools Messages

☒ Pages ☒ Application Objects ☐ All Messages

View All First 1-3 of 3 Last

Search Text - Base	Search Text - Non Base	Replacement Text
	Producto	Articulo
	Full Word Match	
	PRODUCTO	ARTICULO
	Full Word Match	
	producto	articulo
	Full Word Match	

Save Return to Search Add Update/Display

Replacing non-base language text

In this case, the system searches for all instances of the word *Producto* and gives you the option to change each instance to the word *Articulo*.

By making use of the different cases, you make sure that you catch all occurrences.

Searching Base Language and Non-Base Language Text

Search Profile

User ID: PTDMO Identifier: EXAMPLE2 *Status: New

Base Lang: English ☐ Base Language ☐ Search Ready

Search Lang: Spanish ☐ Non-Base Language ☐ Search Only (No Replace)

☒ Both ☐ Search Long Fields

Search Objects

☒ Records, Fields, Xlats ☒ Menus ☒ Messages

☒ Business Processes ☒ Exclude & from Menu Search ☐ Application Messages

☒ Queries and Strings ☒ Other System Objects ☐ PeopleTools Messages

☒ Pages ☒ Application Objects ☐ All Messages

View All First 1-3 of 3 Last

	Search Text - Base	Search Text - Non Base	Replacement Text
ABC abc Abc	Product	Producto	Articulo
ABC abc Abc	Full Word Match	Full Word Match	
ABC abc Abc	PRODUCT	PRODUCTO	ARTICULO
ABC abc Abc	Full Word Match	Full Word Match	
ABC abc Abc	product	producto	articulo
ABC abc Abc	Full Word Match	Full Word Match	

Save Return to Search Add Update/Display

Searching base language and non-base language text

In the preceding example of the Search Profile page, we have selected the **Both** option, so the only occurrences that are returned are those where the search criteria matches on both the base table and the non-base table. Let's say, for example, that you have translated the word *Product* in two different ways in your database, *Producto* and *Articulo*. You then realize that, for consistency, you want *Product* to always be translated to *Articulo*. This search will ignore where *Product* has already been translated to *Articulo* and will search only for the instances where *Product* has been translated to *Producto*. This search will also enable you to change each instance of *Producto* to *Articulo*.

Running the Search Process

Once your Search Criteria is defined and saved, and the **Search Ready** check box is selected, you are ready to run a search.

To run a search:

1. Select PeopleTools, Translate, Process, Search.
2. Search for an existing run control ID or add a new one.

Use the standard search or add method to enter your run control ID and access the Search page.

Search

Run Control ID: RC1 [Report Manager](#) [Process Monitor](#) [Run](#)

User ID	Identifier
1 PTDMO	EXAMPLE

[View All](#) First 1 of 1 Last

[Save](#) [Return to Search](#) [Add](#) [Update/Display](#)

Search page

3. Verify that the appropriate search definitions will be processed.

By default, the search will include all previously unprocessed search definitions (that is, definitions with the status *New*) where the Search Ready check box is selected. If a search appears here that you do not want to run, go back to the Search Profile page and clear the Search Ready check box. If a search is missing, go back to the Search Profile page and select Search Ready.



Click the **View Search Criteria** button to display a read-only version of the search criteria. This is useful when you want to confirm that you're satisfied with the search criteria. Remember that running the search will not, in and of itself, affect your data at all, but it can still be inconvenient to run the wrong search, as you will not be able to go back and modify the search criteria.

Search Criteria

Base Lang: ENG English ☒ Base Language ☐ Search Only (No Replace)

Search Lang: ENG English ☒ Non-Base Language ☐ Search Long Fields

☒ Both

☒ Records, Fields, Xlats ☒ Menus ☒ Messages

☒ Business Processes ☒ Exclude & from Menu Search

☒ Queries and Strings ☒ Other System Objects

☒ Pages ☒ Application Objects

View All First 1 of 1 Last

Search Text - Base	Search Text - Non Base	Replacement Text
Department		Cost Center
Full Word Match		

Search criteria verification

- Click the **Run** button.

The Process Scheduler Request page appears.

Process Scheduler Request

User ID: PTDMO **Run Control ID:** RC1

Server Name: **Run Date:** 07/27/2000

Recurrence: **Run Time:** 5:58:57PM

Time Zone:

Select	Description	Process Name	Process Type	*Type	*Format
<input checked="" type="checkbox"/>	Translate Text Search	PTTSSRCH	SQR Process	Web	PDF

Process Scheduler Request page

- Complete the Process Scheduler Request page.



For more information about completing the Process Scheduler request page, see Submitting a Process Request.

- Click **OK**.

Clicking OK will run this search. Once the search is complete, you can navigate to the Search Results/Replace page to see the results.

Viewing and Replacing Search Results

After a search is completed, select **PeopleTools, Translate, Use, Search Results/Replace** to review the search results.

Replacing Non-Message Text

The Replace Data page displays the database object search results, but not the Message Catalog search results. Use this page to review search results, to verify replacement text, and to indicate when it's all right to run the replace process.

Base Text	Pages	Replacement Text	Too Long	Replace	Non-Base Text
1 Correction-Department	Q	Correction-Cost Center	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
2 Retiree Department	Q	Retiree Cost Center	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
3 Term Deferred Department	Q	Term Deferred Cost Center	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
4 Department of Personnel	Q	Cost Center of Personnel	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
5 Department of Transportation	Q	Cost Center of Transportati	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
6 Department of Planning	Q	Cost Center of Planning	<input type="checkbox"/>	<input checked="" type="checkbox"/>	

Search Results/Replace - Replace Data page

Field Name	Length	Record	RLang Rec
DESCR	30	ACTN_REASON_TBL	ACTN_RSN_LANG
DESCR	30	DEPT_TBL	DEPT_TBL_LANG
DESCR	30	DEPT_TBL	DEPT_TBL_LANG
DESCR	30	DEPT_TBL	DEPT_TBL_LANG
DESCR	30	DEPT_TBL	DEPT_TBL_LANG
DESCR	30	DEPT_TBL	DEPT_TBL_LANG

Search Results/Replace - Replace Data page (scrolled to the right)

To review search results:

1. Examine the search results in both the base language and the non-base language.

The search results appear in a grid, where you can see details about each occurrence of the search text.

Base Text displays the occurrences of the text that you searched for in the base language tables.

Non-Base Text displays the occurrences of the text you searched for in the non-base language tables. This is the text that will be replaced if you decide to run the replace process. If you searched the base language only, there is no non-base text, and the base text will be replaced.

The **Replacement Text**, **Too Long**, and **Replace** columns are used to finalize your replacement text.



For more information, see To verify replacement text.

2. Look to see which pages display the search text.

To help you analyze your search results, the **Pages** column indicates the number of pages on which this instance of the search text appears. This number appears only for objects that are placed on pages: fields, translate values, and (hard-coded) page text.

Click the **View Text Search Details** button next to the **Pages** column to display details about that occurrence of the search text.

Text Search Detail

Records and Field

ACTN_REASON_TBL	ACTN_RSN_LANG
DESCR	

Keys

View All First 1-3 of 3 Last

ACTION	DTA
ACTION_REASON	CDP
EFFDT	1900-01-01

Text


Correction-Department

Correction-Cost Center

Return

Text Search Detail page

This page displays the **Record** and **Field** where this instance was found, as well as the **Keys** for each occurrence of the text in the record. The page also displays the **Text** from the base language table, the corresponding translation from the related language table, and the suggested replacement text.

Clicking the  **View Text Found** button displays information about which search words were found. Remember that a single set of search criteria can include searches for many terms. For example, if you searched for both *Codigo* and *Departamento*, the Text Found page would show which word or words were found in this record.

Text Found		
View All First 1 of 1 Last		
Base Search Text	Non-Base Search Text	Replacement Text
1 Department	Department	Cost Center
Return		

Text Found page

3. If you like, look over the information about the record and field where the text was found.

The **Field Name** is the system name of the field where the text resides.

The length of this field appears in the **Length** column. This information is useful when you plan to replace text, because it tells you the maximum length of any replacement text.

The **Record** is the name of the record where this occurrence was found.

RLang Rec (related language record) is the name of the corresponding related language table.

4. If you like, inspect the original search criteria.



Click the **View Search Criteria** button to display a read-only version of the search criteria. This can help you understand your search results.

To verify replacement text:

1. Inspect the suggested replacement text.

The **Replacement Text** shows the suggested replacement text, based on your original search criteria. At this point no replacement has occurred.

Remember that the replacement text will replace the non-base text unless you searched only the base language, in which case it will replace the base text.

2. If necessary, modify the replacement text.

You can modify the replacement text if you're not happy with the suggested text or if the suggested text is too long for the field.

The system selects the **Too Long** check box when the suggested replacement text is too long for the field. This can occur when the replacement text is longer than the search text. When this happens, you must edit the replacement text (perhaps using an abbreviation) so that it fits in the field. It may help to look at the **Length** column to see the maximum allowable length.

Once the replacement text is an acceptable length, the system clears the **Too Long** check box.

You don't need to modify the text if you decide that you don't want the replacement to happen at all; the next step describes a different mechanism for disabling replacement for specific rows.



When the replacement text is too long, the search text will not be replaced during the replace process.

3. Selectively enable and disable replacement processing.

You can replace some occurrences of the search text without replacing all of them. You control which occurrences get replaced using the **Replace** check box.

Leave the **Replace** check box selected for the occurrences that you want to replace; clear the check box for the occurrences that you want left as they are. When you run the replace process, only the selected rows will be updated.

After you run a search, the **Replace** check box is automatically selected for all occurrences other than those where the suggested replacement text is too long for the field.



To clear the **Replace** check box in all rows, click the arrow that curves to the left.



To select the **Replace** check box in all rows (other than those where the replacement text is too long for the field), click the arrow that curves to the right.

To enable or disable replace processing:

1. To enable replace processing, select **Replace Ready**.

Once you are satisfied with all your replacement settings, select **Replace Ready** to enable replace processing. The replace process will only pick up searches where this option is selected.

2. To permanently disable replace processing for this specific search, click the **Complete** button.

This button will change the status of the results to **Complete**. Once this happens, no further processing is possible. If you've edited messages on the Replace Messages page, those changes become permanent.



For more information about editing messages, see Replacing Message Text.

3. Save the page.

Replacing Message Text

The Replace Messages page provides a focused view of all the message catalog entries that satisfy the search criteria. Use this page to review the messages and make replacement changes manually. This can be done at any time in the Search/Replace/Undo cycle.



You will update message text manually and, if you continue to the undo step, you'll back out your changes manually. The automatic replace and undo process for other database objects does not apply to message text.

Search Results/Replace - Replace Messages page

To review search results and update message text:

1. Click the  lightning bolt button to load the results into the page.

When you first access the Replace Message page, there are no visible search results. Clicking the lightning bolt button retrieves the search results.

Once the results are loaded, you can see each message that contained the search text from your search.

2. If you like, inspect the original search criteria.



Click the **View Search Criteria** button to display a read-only version of the search criteria. This can help you understand your search results.

3. Review the messages.

You can see only one message at a time on this page; to review the messages, page through them and look at each one individually.

The **Message Set Number**, **Message Number**, and **Language Code** identify the message that is visible.

The **Last Update Timestamp** indicates when the message was last modified.

4. Edit the messages as necessary.

As you review the messages, determine whether it is appropriate to replace the text or reword the message. When you decide to make changes, you can edit the text directly on this page. Saving the page updates the Message Catalog.



You must update message text manually; the automatic replace process for other database objects does not apply to message text.

5. Save the page.

Running the Replace Process

After you've finalized your replacement text and settings in the Search Results/Replace page, you need to run a replace process to actually make the replacements. Running the replace process updates the search status from *Searched* to *Replaced*.

After you run the replace process, you will still have a final opportunity to back out any changes made by the process.



Once you change the search status and update the search status, you can no longer view this search in the Search Result/Update pages. Be sure that you are finished with *both* pages (that is, you've finished manually editing your messages) before you change the status.



For more information about editing messages, see Replacing Message Text.

To run the replace process:

1. Select PeopleTools, Translate, Process, Replace.
2. Search for an existing run control ID or add a new one.

Use the standard search or add method to enter your run control ID and access the Replace page.

Replace

Run Control ID: RC1 [Report Manager](#) [Process Monitor](#) [Run](#)

User ID	Identifier	Search DateTime
1 PTDMO	EXAMPLE2	11/10/2000 10:25:34PM

View All First 1 of 1 Last

Save Return to Search Next in List Previous in List Add Update/Display

Replace page

3. Verify that the appropriate search definitions will be processed.

By default, the replace will include all search definitions that have the status *Searched* and have the Replace Ready check box selected. If a search appears here that you do not want to run, go back to the Search Results/Replace page and clear the Replace Ready check box. If a search is missing, go back to the Search Results/Replace page and select Replace Ready.



Click the **View Search Criteria** button to display a read-only version of the search criteria. This is useful when you want to confirm which set of criteria you're processing.

4. Click the **Run** button.

The Process Scheduler Request page appears.

Process Scheduler Request					
User ID: PTDMO		Run Control ID: RC1			
Server Name:	<input type="text"/>	Run Date:	<input type="text" value="07/27/2000"/>		
Recurrence:	<input type="text"/>	Run Time:	<input type="text" value="6:12:12PM"/>		
Time Zone:	<input type="text"/>	<input type="button" value="Reset to Current Date/Time"/>			
Process List					
Select	Description	Process Name	Process Type	*Type	*Format
<input checked="" type="checkbox"/>	Translation Text Replace	PTTSRPLC	SQR Process	<input type="text" value="Web"/>	<input type="text" value="PDF"/>
<input type="button" value="OK"/> <input type="button" value="Cancel"/>					

Process Scheduler Request page

5. Complete the Process Scheduler Request page.



For more information about completing the Process Scheduler request page, see [Submitting a Process Request](#).

6. Click **OK**.

Clicking OK will run this replace. Once the replace is complete, you can navigate to the Replace Results/Undo page to see the results.

Viewing and Undoing Replace Results

Undoing Data Replacements


After you run the replace process, select **PeopleTools, Translate, Use, Replace Results/Undo** to review your results and decide whether to undo any of the replacements.

Undo Data

Undo Messages




User ID: PTDMO







Identifier: EXAMPLE

*Status: Replaced 


Base Lang: English

Search Lang: English

☐ Undo Ready   

Non-Base Text	Base Text	Pages	Replacement Text	Replaced	Undo	Field Name
1	Correction-Department		Correction-Cost Center	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	DESCR
2	Retiree Department		Retiree Cost Center	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	DESCR
3	Term Deferred Department		Term Deferred Cost Center	<input type="checkbox"/>	<input type="checkbox"/>	DESCR
4	Department of Personnel		Cost Center of Personnel	<input type="checkbox"/>	<input type="checkbox"/>	DESCR
5	Department of Transportation		Cost Center of Transportation	<input type="checkbox"/>	<input type="checkbox"/>	DESCR
6	Department of Planning		Cost Center of Planning	<input type="checkbox"/>	<input type="checkbox"/>	DESCR

Save

 Return to Search

Replace Results/Undo - Undo Data page

View All First 1-6 of 75 Last		
Field Name	Length	Record RLang Rec
DESCR	30	ACTN_REASON_TBL ACTN_RSN_LANG
DESCR	30	DEPT_TBL DEPT_TBL_LANG
DESCR	30	DEPT_TBL DEPT_TBL_LANG
DESCR	30	DEPT_TBL DEPT_TBL_LANG
DESCR	30	DEPT_TBL DEPT_TBL_LANG
DESCR	30	DEPT_TBL DEPT_TBL_LANG


Replace Results/Undo - Undo Data page (scrolled to the right)

To review replacement text:

1. Review the results of the replace process.

The Replace Results/Undo page shows the same information as the Replace Data page.

The **Base Text** and, if applicable, the **Non-Base Text** show the original text.

You can see how many **Pages** display the text, and you can click the  **View Text Search Details** button to see details about these pages.

The **Replacement Text** field shows what was—or would have been—substituted for the original search text.

The **Replaced** check box indicates whether the original search text was replaced. At this point, the original text has been replaced, but the change has not been finalized.

The **Field Name**, **Length**, **Record**, and **RLang Rec** columns provide details about where the text is stored in the database.

2. Selectively enable and disable undo processing.

You can undo some of the replacements without undoing all of them. You control which occurrences are replaced, by using the **Undo** check boxes.

Select **Undo** for those rows where you want to back out your changes. Leave the **Undo** check box clear for the rows where you're satisfied with the replacement. When you run the undo process, only the selected rows will be affected.



To select the **Undo** check box in all rows, click the arrow that curves to the right.



To clear the **Undo** check box in all rows, click the arrow that curves to the left.

3. Save the page.

To enable or disable undo processing:

1. To enable replace processing, select **Undo Ready**.

Once you are satisfied with all of your undo settings, select **Undo Ready** to enable undo processing. The undo process will pick up only the rows where this check box is selected.

2. To permanently disable undo processing for this search, click the **Complete** button.

This button will change the status of the results to **Complete**. Once this happens, no further processing is possible. Any replacements that have been made, including changes to messages, are now finalized.



For more information about editing messages, see Replacing Message Text.

3. Save the page.

Undoing Text Replacements

The Undo Messages page displays the search results in the Message Catalog. If you edited the messages during the replace step, the text reflects those changes.

The Undo Messages page is identical to the Replace Messages page, and you can review and update messages, using the same procedures.



For more information about editing messages, see Replacing Message Text.



Just as you have to replace message text manually, so must you back out any changes manually. The automatic replace and undo process for other database objects does not apply to message text.

Running the Undo Process

To run the undo process:

1. Select PeopleTools, Translate, Process, Undo.
2. Search for an existing run control ID or add a new one.

Use the standard search or add method to enter your run control ID and access the Undo page.

Undo

Run Control ID: RC1 [Report Manager](#) [Process Monitor](#) [Run](#)

User ID	Identifier	Search DateTime	Replace DateTime
1 PTDMO	EXAMPLE	11/10/2000 6:14:22PM	11/10/2000 9:42:56PM

[Save](#)
[Return to Search](#)
[Next in List](#)
[Previous in List](#)
[Add](#)
[Update/Display](#)

Undo page

3. Verify that the appropriate search definitions will be processed.

By default, the undo will include all search definitions that have the status **Replaced** and the Undo Ready check box selected. If a search appears here that you do not want to run, go back to the Replace Results/Undo page and clear the Undo Ready check box. If a search is missing, go back to the Replace Results/Undo page and select the Undo Ready check box.



Click the **View Search Criteria** button to display a read-only version of the search criteria. This is useful when you want to confirm which set of criteria you're processing.

4. Click the **Run** button.

The Process Scheduler Request page appears.

Process Scheduler Request					
User ID: PTDMO		Run Control ID: RC1			
Server Name:	<input type="text"/>	Run Date:	<input type="text" value="07/27/2000"/>		
Recurrence:	<input type="text"/>	Run Time:	<input type="text" value="6:18:36PM"/>		
Time Zone:	<input type="text"/>	<input type="button" value="Reset to Current Date/Time"/>			
Process List					
Select	Description	Process Name	Process Type	^Type	^Format
<input checked="" type="checkbox"/>	Translate Text Undo	PTTSUNDO	SQR Process	Web	PDF
<input type="button" value="OK"/> <input type="button" value="Cancel"/>					

Process Scheduler Request page

5. Complete the Process Scheduler Request page.



For more information about completing the Process Scheduler Request page, see Submitting a Process Request.

6. Click **OK**.

Clicking OK will run this undo. Once the undo is complete, you can navigate to the All Search Results page to see the results.

Reviewing Search Information

Three read-only pages provide you with information about searches.

Reviewing Search Criteria

To see the search criteria for any search that's been defined, select **PeopleTools, Translate, Inquire, Search Criteria**.

Search Criteria page

This page is a display-only version of the Search Profile page.



For more information about the fields displayed on this page, see [Defining Search Criteria](#).



To transfer to the page that is appropriate to the current status of the search, click the **View Details** button. For example, if the status is **Searched**, then in order to continue the search-replace-undo process, you need to go to the Search Results/Replace pages. The following table explains which component you'll see, based on the status of the search:

Status	Component
New	Define Search Criteria (PeopleTools, Translate, Use, Define Search Criteria)
Searched	Search Results/Replace (PeopleTools, Translate, Use, Search Results/Replace)
Replaced	Replace Results/Undo (PeopleTools, Translate, Use, Replace Results/Undo)
Complete	All Search Results (PeopleTools, Translate, Inquire, All Search Results)

Complete	Search Results Only (PeopleTools, Translate, Inquire, Search Results Only)
----------	---



Click the **Copy Search Criteria** button to clone this search definition.



For more information about cloning a search definition, see Cloning Existing Search Criteria.

Viewing Results for Search Only Searches

When a search is defined as Search Only (No Replace), the status is set to **Complete**, which prevents you from viewing the search results on the Search Results/Replace pages. Instead, you must access the result using one of the Inquiry pages.

To see the search results for a completed Search Only (No Replace) search, select **PeopleTools, Translate, Inquire, Search Only Results**.

This page is a display-only version of the Search Results/Replace page.



For more information about the fields displayed on this page, see Viewing and Replacing Search Results.

Viewing Search Results for All Searches

To see the search results for any search, select **PeopleTools, Translate, Inquire, All Search Results**.

Review Data
Review Messages

User ID: PTDMO Identifier: EXAMPLE *Status: Complete
Base Lang: English
Search Lang: English

Non-Base Text	Pages	Replaced	Undone	Replacement Text	Base Text	Field Name	Length
1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Correction-Cost Center	Correction-Department	DESCR	30
2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Retiree Cost Center	Retiree Department	DESCR	30
3		<input type="checkbox"/>	<input type="checkbox"/>	Term Deferred Cost Center	Term Deferred Department	DESCR	30
4		<input type="checkbox"/>	<input type="checkbox"/>	Cost Center of Personnel	Department of Personnel	DESCR	30
5		<input type="checkbox"/>	<input type="checkbox"/>	Cost Center of Transportation	Department of Transportation	DESCR	30
6		<input type="checkbox"/>	<input type="checkbox"/>	Cost Center of Planning	Department of Planning	DESCR	30

Save
 Return to Search
 Next in List
 Previous in List

Review Data page - All Search Results

This page is a display-only version of the Replace Results/Undo page.



For more information about the fields displayed on this page, see Viewing and Undoing Replace Results.

Understanding Searchable Objects

When you define your search, you select which types of objects to include in your search. You do this by selecting from predefined groups of object types.

Search Objects

☒ Records, Fields, Xlats
☒ Menus
☒ Messages

☒ Business Processes
☒ Exclude & from Menu Search
☐ Application Messages

☒ Queries and Strings
☒ Other System Objects
☐ PeopleTools Messages

☒ Pages
☒ Application Objects
☒ All Messages

Groups of object types



PeopleSoft has already defined each group by associating particular tables in each group. You can see which tables are included in each group by clicking the **View Objects** button.

Text Search Records			
		View All	First 1-4 of 4 Last
PSDBFIELD	Database Field	PSDBFIELDLANG	Database Field(Alt Language)
PSDBFLDLABL	DBField Label Table	PSDBFLDLABLLANG	DBField Label Language Table
PSRECDEFN	Record Definition	PSRECDEFNLANG	Record Definition(Alt Lang)
XLATABLE	Translate Value	XLATABLE	Translate Value

Tables associated with the Records, Fields, Xlats group

You can use the Define Text Search Records page to view additional information about these records. Specifically, you can identify the key fields and the searchable fields.

You can use the same page to change the association between searchable fields and the groups of object types. However, this is not advisable. The only time you should modify the object groups is when you want to add newly created tables to the **Application Objects** group; PeopleSoft provides a PeopleSoft Data Mover script to do this for you.

Viewing Searchable Objects

You can use the Define Text Search Records page to view additional information about the searchable records.

To view the searchable fields in any searchable record:

1. Select PeopleTools, Translate, Use, Define Text Search Records.

You are prompted to identify the record you want to view.

Define Text Search Records - Search page

You can search for records either by **Record (Table) Name** or by **Object Type**. The object types are the same as the groups you use when defining a search.

2. Search for the record you want to view.

Use the standard search or add method to enter the user ID and access the Text Search Records page.

Text Search Records

Record: XLATTABLE ☒ Combined Language Table

Related Language Record Name: XLATTABLE

Object Type: Records, Fields, Xlats

Keys View All First 1-3 of 3 Last

EFFDT

FIELDNAME

FIELDVALUE

Searchable Text Fields View All First 1-2 of 2 Last

XLATLONGNAME

XLATSHORTNAME

Save Return to Search Next in List Previous in List

Text Search Records page

- Examine the information about this record.

The **Record** and **Related Language Record Name** identify the record you're looking at.

Combined Language Table is selected when the base record and the related language record are the same; you cannot change this setting.

The **Object Type** identifies which searchable group of objects includes this record.



Although you can modify the **Object Type** on this page, it is not advisable. As delivered, records are already logically organized into appropriately named groups; changing the association would cause the group names to no longer reflect the records in the group.

The **Keys** and the **Searchable Text Fields** fields display additional information about the record. All translatable fields (that is, the non-key fields in the related language record) are searchable.

Modifying Searchable Objects

If you create additional related language tables for delivered tables or develop a new translatable structure, you can make those tables searchable by adding them to the Application Objects group. To do this, run the PeopleSoft Data Mover script TSRECPOP.dms. This script refreshes the Application Objects group so that it includes all application tables that have related language tables.

Upgrade Considerations

Be aware that terminology updates are registered as changes to the affected database objects. This means that when you upgrade to a new PeopleSoft release, your object definitions are out of sync with those in the new PeopleSoft database.

This means that your upgrade reports, which identify changed objects, will include all the objects that have new or changed terms. Depending on the extent of your terminology changes, this can significantly impact the amount of time you spend analyzing the differences.

If you have implemented the PeopleSoft system with no customizations, you can deal with this issue by accepting the new PeopleSoft-delivered objects and then reapplying your terminology changes.

If you've customized your system and modified the delivered terminology, you may want to minimize the upgrade compare differences by doing one of the following:

- Reversing your terminology changes before the upgrade and then redoing the terminology changes after the upgrade.
- Applying your terminology changes to the PeopleSoft-delivered software before running the upgrade and compare reports.

Both of these methods cause the upgrade reports to disregard any terminology-only changes, which simplifies your analysis of these reports.



Upgrading becomes more complicated and involves extra steps when you've made terminology changes. Be sure to consider this cost when you decide whether terminology changes are necessary.

CHAPTER 10

Understanding Time Zones

PeopleSoft databases store times relative to a system-wide base time zone. If the base time zone is Universal Coordinated Time (also known as Greenwich Mean Time), a time entered as 10 a.m. U.S. Eastern Standard time is stored as 6 p.m. because of the time difference between the two locations. Although the time is stored as 6 a.m., it can still be displayed as 10 a.m. Eastern time. In fact, it can be displayed as the appropriate time for any time zone that you choose. You can also set up the system to enable users to choose the time zone for specific time or datetime displays.

To support display in alternate time zones, PeopleSoft maintains a global list of time zones, including information about daylight savings time observances. When a time zone begins or ends daylight savings time, both the description of the zone and the times that are associated with that zone are adjusted to reflect the change.

This approach ensures chronological consistency throughout an organization's geographically dispersed locations, while allowing for flexibility in the way that users see times.

Types of Time Zones

PeopleTools defines three types of time zones that are important to the operation of the system:

- Base time zone
- User time zone
- Specified time zone

Base Time Zone

The base time zone is the primary system-wide time zone in which PeopleTools processes dates internally and which is used for display when no other time zone is specified. The base time zone must match the time zone in which the database server is operating, specifically the time zone to which the MetaSQL %currentdatetime token returns. For example, in an Oracle database, the base time zone is the time zone in which the SYSDATE system variable is returned. You must enter this time zone on the PeopleTools Options page in order for your system to function correctly.



For more information, see [Setting the Base Time Zone](#).

Additionally, the base time zone is used for:

- All effective-date processing. Rows become effective when midnight passes on the effective date according to the base time zone of the system.
- Internal processing in PeopleCode and all batch processes.
- Display on pages and in reports, if no other time zone information is specified or available

User Time Zone

The user time zone is the time zone in which each system user is currently operating. It is read from the operating system of the user's Web Browser for PeopleSoft Internet Architecture and from the Windows Control Panel for the Windows Client. Although users cannot change their user time zone while they are logged into PeopleTools, they may be able to change it in their local operating system. Therefore, no validation or business rules being processed are based on the user time zone in PeopleSoft. The user time zone is used only for display and data entry purposes.

Specified Time Zone

A specified time zone is a time zone that a user or the system enters on a page as a fixed time zone for a specific row of data. This is useful for applications where all users must see the same time and time zone as the user who enters the transaction, such as in call processing systems. To implement a specified time zone field, your record must contain a time zone control field.



For more information about time zone control fields, see Controlling Time Zone Display.

Setting the Base Time Zone

The base time zone for PeopleTools is the same as the time zone used by the database. In order for PeopleSoft to properly manage time data, the system needs to know which time zone that is. You enter this information on the PeopleTools Options page. It is critical for the correct operation of the system that this time zone match the time zone in which your database is operating. Any discrepancy in the base time zone, as defined on the PeopleTools Options page, and the time zone in which the database system is operating will lead to inaccurate time processing.

To set the base time zone:

1. Select PeopleTools, Utilities, Use, PeopleTools Options.

The PeopleTools Options page appears.

PeopleTools Options page

2. Enter the Base Time Zone.

Valid values are maintained in the Time Zone component.



For more information about the Time Zone component, see [Maintaining Time Zones](#).

3. Save the page.



After selecting a base time zone during installation, avoid changing it. There is no automated processing to translate existing time information in the database to a new base time zone, and if you change the time zone defined on the PeopleTools Options page, no adjustment will be made to the existing time data in your database.

Controlling Time Zone Display

Although times are always stored in the base time zone, you can choose whether a page displays a time in base time or some other time zone.

Time zone display and processing functions operate only on time or datetime fields, not on date fields. This is because all time zone processing requires the knowledge of the time component of the field, which is not provided in date fields. Even if some databases store PeopleSoft date fields internally, as datetime fields, PeopleSoft does not use the time component of these fields.

You control the display using a combination of settings:

- Record field properties

In the record field properties, you can choose whether the time display is based on the base time zone or on another time zone. If you base the display on another time zone, you can specify a time zone control field: the field that is used to determine the time zone.

- Page field properties

Choose whether to automatically include the time zone abbreviation in a field that displays time or datetime values.

These two settings work together to control the time display on pages in PeopleSoft Internet Architecture.

If the record field properties specifies that time is displayed in the *base time zone*, setting the page field properties to show the time zone will help users interpret the time shown on the page.

If the record field properties specifies that the time zone of a particular time or datetime field is determined by the *time zone control field* then you might choose to put that control field directly on the page, to allow the user to select the time zone for a field manually.

If you put the time zone control field on the page where the users are permitted to enter a time zone, you will most likely want to disable the page field properties option that includes the time zone abbreviation in the time field, as it would be redundant to show the time zone in both places.

If you don't permit users to enter a time zone manually (for example, if you use the %clienttimezone or %basetimezone system variable to set the time zone), then you can display the time zone either by including the control field on the page or by setting the page field properties appropriately—but you needn't do both.

In the Windows client, users can display time values in their local time zone or the system's base time zone, regardless of how the record field properties and page field properties are set. By default, all Windows client time and datetime fields are displayed in the base time zone, but each user can override this option by selecting **View, Time Display, Local Time** or **View, Time Display, Database Time**.

Setting Time Zone Options in Record Field Properties

When a record includes a time or datetime field, choose whether PeopleSoft displays the time in the system's base time zone or in another time zone such as a time zone that corresponds to the user's location. Set this option for all time and datetime fields.

To set the time zone that controls the display time:

1. In the record, open the Record Field Properties dialog box for the time field.

Open the Record Field Properties dialog box by double-clicking the field that displays the time. The Use tab appears.

The image shows the 'Record Field Properties' dialog box with the 'Use' tab selected. The 'Field Name' is 'TIMEA'. The 'Keys' section includes checkboxes for 'Key', 'Duplicate Order Key', 'Alternate Search Key', 'Descending Key', 'Search Key', 'List Box Item', 'From Search Field', 'Through Search Field', and 'Default Search Field'. The 'Audit' section includes checkboxes for 'Field Add', 'Field Change', 'Field Delete', 'System Maintained', and 'Auto-Update'. The 'Record Field label ID' is set to 'Use Default Label'. The 'Default Value' section has 'Constant' and 'Record Name' fields. The 'Default Page Control' is set to 'System Default'. The 'Time Zone' section has 'Specified Time Zone' checked and 'Time Zone Control Field' set to 'TIMEZONE'. The 'Related Date' section has a 'Date Control Field'.

Record Field Properties dialog box- Use tab

2. Set the **Time Zone** property.

If you leave the **Specified Time Zone** check box clear, the time is displayed in the database's base time zone (unless overridden by the user to display in their local time zone, using the View, Time Display menu in the Windows Client).

If you select **Specified Time Zone**, the system converts the time according to the time zone that is specified in the **Time Zone Control Field**. The time zone control field must be a field in this record.

A single time zone control field can control multiple time and datetime fields on the record. Using a single control field causes all times to be displayed in the same time zone. If you want multiple time or datetime fields to be displayed in different time zones, you need separate control fields for each of them.

You may want to set the default value of the time zone field to %clienttimezone or %basetimezone. However, the default time zone would be used only when the transaction is created. Because the time zone is saved along with the transaction itself, future users who access the transaction will see the time relative to that time zone.

3. Click **OK**.

Setting Time Zone Options in Page Field Properties

When you place a time or datetime field on a page, you can choose whether to display the time zone in the same field where the time appears.

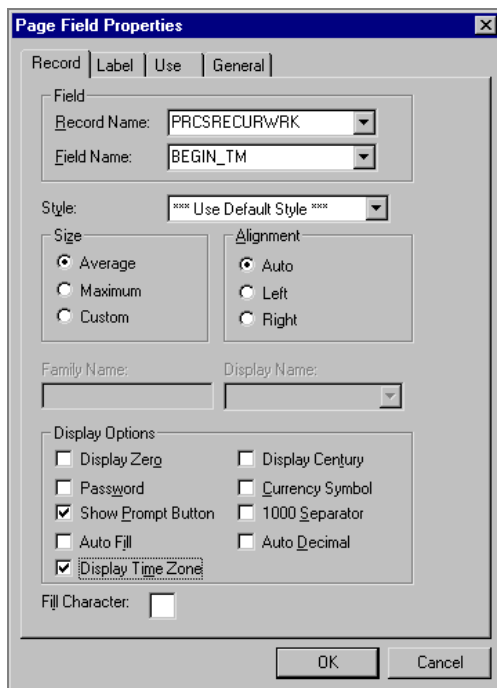
You would normally want to display the time zone this way when you want users to be aware of the time zone of a field without being able to change the time zone.

If, on the other hand, you want users to be able to set the time zone, don't display the time zone in the time field. Instead, design your page so that it includes a separate drop-down list box referencing the time zone control field. With this design, users can both see and change the time zone.

To show the time zone in the time field:

1. On the page, open the Page Field Properties dialog box for the time field.

Open the Page Field Properties dialog box by double-clicking the page control that displays the time. The Record tab appears.



Page Field Properties dialog box- Record tab

2. Select Display Time Zone.

When this check box is selected, the page displays the time zone in the same field as the time. Selecting this check box resizes the edit box to allow space for up to 10 additional characters.

Start Request	
Date:	03/18/2000 <input type="button" value="BT"/>
Time:	1:00:00AM PDT

Example of time field where Display Time Zone is selected

This option has no effect on data entry. Users cannot override the time zone. A user who sees the time in Greenwich Mean Time (GMT) must enter the time in GMT.

If the field that is associated with this page control is not a time or datetime field, the **Display Time Zone** check box has no effect.

3. Click **OK**.

PeopleCode Functions

There are a number of PeopleCode functions you can use to support the appropriate time zone display on pages. There are two functions that convert times to their base time zone equivalent: **ConvertTimeToBase()** and **ConvertDatetimeToBase()**. Another function, **IsDaylightSavings()**, establishes whether daylight savings time is in effect on a particular date.

Additionally, two system variables, `%clienttimezone` and `%basetimezone`, can be used in PeopleCode programs or as field default values. `%basetimezone` returns the base time zone of the database, and `%clienttimezone` returns the current user's local time zone.



All time zone functionality applies to displayed values only. Internally all values are stored and manipulated in the base time zone. This means that all time values in PeopleCode are in database time, even when the time is displayed to the user in another time zone. This allows for easy comparison and manipulation of times so that the PeopleCode developer does not have to worry about time zone differences. Additionally, all batch processes such as SQR and COBOL operate in the base time zone of the system.



For more information about PeopleCode date and time functions, see [Date and Time](#).

Maintaining Time Zones

You maintain a list of time zones on the Time Zone Data page. PeopleSoft delivers a comprehensive list of time zones, including information about daylight savings time observances. You can update the data if necessary; for example, to include a time zone for a locality with daylight savings observances that differ from other parts of its time zone.

Viewing Time Zone Definitions

To review or update time zone data, select **PeopleTools, Utilities, Use, Time Zones**, and view the Time Zone Data page.

Time Zone Data
Daylight Savings

Generate Query Offsets...

*Time Zone	Time Zone Standard Label	Time Zone Daylight Label	Universal Coord Time Offset	Observes Daylight Savings Time	Daylight Savings Offset	Daylight Saving Time Start	Daylight Savings Time End	Time Zone Description
9 CST	CST	CDT	-360	<input checked="" type="checkbox"/>	60	2FirstSunApr	2LastSunOct	Central Time
10 DST	DST		-720	<input type="checkbox"/>				Dateline Time, Eniwetok, Kwajalein
11 EKST	EKST	EKDT	300	<input checked="" type="checkbox"/>	60	2LastSunMai	3LastSunOct	Ekaterinburg Time
12 EST	EST	EDT	-300	<input checked="" type="checkbox"/>	60	2FirstSunApr	2LastSunOct	Eastern Time
13 GFTST	GFTST	GFTDT	120	<input checked="" type="checkbox"/>	60	2LastSunMai	3LastSunOct	GFT Time, Athens, Istanbul, Minsk
14 GMT	GMT	GMDT		<input checked="" type="checkbox"/>	60	2LastSunMai	3LastSunOct	GMT, London, Dublin, Lisbon, Edint
15 HST	HST		-600	<input type="checkbox"/>				Hawaiian Time
16 IRST	IRST	IRDT	210	<input checked="" type="checkbox"/>	60	2FirstSunMai	24thTueSep	Iran Time, Tehran

Save
Previous tab
Next tab

Time Zone Data | Daylight Savings

Time Zone Data page

Time Zone	A unique identifier for the time zone.
Time Zone Standard Label	The label that appears for the time zone during standard time.
Time Zone Daylight Label	The label that appears for the time zone during daylight savings time.
Universal Coord Time Offset	<p>The number of minutes the time is offset from Universal Coordinated Time (UTC), also known as Greenwich Mean time (GMT).</p> <p>A positive offset indicates a time zone east of UTC; a negative offset indicates a time zone west of UTC.</p> <p>For example the time zone for India, which is 5 hours 30 minutes east of UTC has an offset of +330. U.S. Pacific Standard Time, which is 8 hours west of UTC, has an offset of -480.</p>
Observes Daylight Savings Time	Indicates whether the time zone observes daylight savings time.
Daylight Savings Offset	The number of minutes by which the time is offset during daylight savings time.
Daylight Savings Time Start	The day that daylight savings time begins. This field prompts against values from the Daylight Savings page.

Daylight Savings Time End	The day that daylight savings time ends. This field prompts against values from the Daylight Savings page.
Time Zone Description	Description of the time zone.

The **Generate Query Offsets** button takes the information from the Time Zone Data page and populates the PSTZOFFSET table with offsets for all the time zones and their daylight savings time periods for a specified period of time. This makes the time zone information available in a format that can be easily accessed with SQL. This may be useful for the PeopleSoft Query tool.

PeopleSoft does not require that you populate this table; you need to do so only if you require access to this information. If you do use the information, be sure to regenerate the offsets anytime the underlying time zone data is changed.

To generate query offsets:

1. Click Generate Query Offsets.

You are prompted to enter a start date and an end date so that you can select a reasonable range of dates.

Time Zone Offset Generation page

2. Enter a Start Date and an End Date.

Select a reasonable range of dates. With the time zones delivered by PeopleSoft, more than 100 rows are generated for each year.

3. Click **OK**.

The data is generated for the table. Any existing data for the same range of dates is overwritten.

For example, the following data from the PSTZOFFSET table represents one time zone, EST (Eastern Standard Time) for a two-year period, from 01/01/1999 to 01/01/2001. It is based on a system with PST (Pacific Standard Time) as the base time zone. When you generate query offsets, the PSZOFFSET table contains similar data for all the time zones.

TIMEZONE	STARTDATETIME	ENDDATETIME	BASEOFFSET	TIMEZONE LABEL
EST	1999-01-01 00:00:00.000	1999-04-03 23:00:00.000	180	EST
EST	1999-04-03 23:00:00.000	1999-04-04 02:00:00.000	240	EDT
EST	1999-04-04 02:00:00.000	1999-10-30 23:00:00.000	180	EDT
EST	1999-10-30 23:00:00.000	1999-10-31 02:00:00.000	120	EST
EST	1999-10-31 02:00:00.000	2000-04-01 23:00:00.000	180	EST
EST	2000-04-01 23:00:00.000	2000-04-02 02:00:00.000	240	EDT
EST	2000-04-02 02:00:00.000	2000-10-28 23:00:00.000	180	EDT
EST	2000-10-28 23:00:00.000	2000-10-29 02:00:00.000	120	EST
EST	2000-10-29 02:00:00.000	2001-01-01 00:00:00.000	180	EST

PSTZOFFSET table

The first row of data shows that from 1999-01-01 at midnight until 1999-04-03 at 11 p.m. (base time—PST, in this case), the offset between EST and the base time zone is 180 minutes, meaning that EST is 180 minutes ahead of PST. The label for EST during this period is *EST*.

The second row shows that from 1999-04-03 at 11 p.m. until 1999-04-04 at 2 a.m., there is a three-hour time period during which EST is 240 minutes ahead of PST. This is because EST has changed to daylight savings time, but PST hasn't changed yet. The label for EST during this period is *EDT*: Eastern Daylight Time.

Viewing Daylight Savings Start and End Dates

On the Time Zone page, you specify the start and end days for daylight savings time by referencing a list of standard start and stop days. This list can include days in absolute format (for example, October 1) or in relative format (for example, Last Sunday in October). Entries on this list are not associated with any particular year.

To review or update time zone data, select **PeopleTools, Utilities, Use, Time Zones**, and view the Daylight Savings page.

Time Zone Data
Daylight Savings

View All
First
1-10 of 17
Last

*Daylight Savings Time ID	Absolute	Month	Day	Day of the Week	Hour	Minute	Description
1 0LastSunMar	<input type="checkbox"/>	Mar	5	Sunday	0	0	Last Sunday in March, midnight
2 1LastSunSep	<input type="checkbox"/>	Sep	5	Sunday	1	0	Last Sunday in Sept, 1:00am
3 22ndSunFeb	<input type="checkbox"/>	Feb	2	Sunday	2	0	Second Sunday in Feb, 2:00am
4 24thTueSep	<input type="checkbox"/>	Sep	4	Tuesday	2	0	Fourth Tuesday in Sept, 2:00am
5 2FirstFriMay	<input type="checkbox"/>	May	1	Thursday	2	0	First Friday in May, 2:00am
6 2FirstSunApr	<input type="checkbox"/>	Apr	1	Sunday	2	0	First Sunday in April, 2:00am
7 2FirstSunMar	<input type="checkbox"/>	Mar	1	Sunday	2	0	First Sunday in March, 2:00am
8 2FirstSunOct	<input type="checkbox"/>	Oct	1	Sunday	2	0	First Sunday in Oct, 2:00am
9 2LastSunMar	<input type="checkbox"/>	Mar	5	Sunday	2	0	Last Sunday in March, 2:00am
10 2LastSunOct	<input type="checkbox"/>	Oct	5	Sunday	2	0	Last Sunday in October, 2:00am

Save
Previous tab
Next tab

Time Zone Data | Daylight Savings

Daylight Savings page

Daylight Savings Time ID	A unique identifier for the date.
Absolute	Indicates whether the date is an absolute date. Leave this check box clear to indicate a relative date such as <i>Last Sunday in October.</i>
Month	The month.
Day	<p>For absolute dates, this is the day of the month.</p> <p>For relative dates, the numbers <i>1, 2, 3,</i> and <i>4</i> represent the first, second, third, and fourth occurrence of the day of the week; for example, to describe the second Sunday in February, you would enter <i>2.</i></p> <p>The number <i>5</i> for a relative date indicates the last occurrence of the specified day of the week, even if that day of the week occurs only four times in the month.</p>
Day of the Week	Day of the week.
Hour	The hour for the time of day.
Minute	The minutes for the time of day.
Description	A description of the date.

CHAPTER 11

Controlling Currency Display Format

This chapter describes the PeopleTools features that support currency display formats. These include the definitional settings that control the formatting and display of foreign currencies and an SQR process for resizing international currency fields.

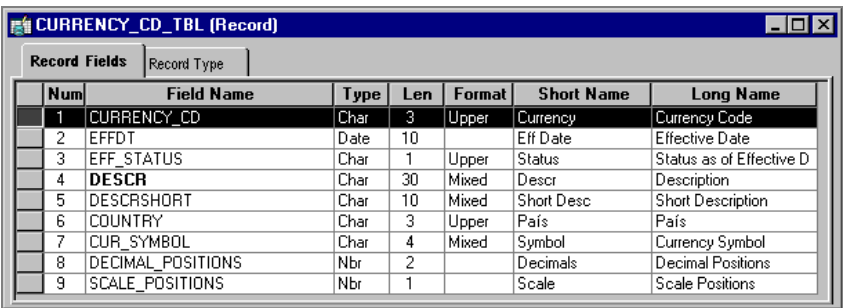
Maintaining Currency-Specific Settings

PeopleTools can automatically format a currency amount that is displayed on a page or a report based on both your numerical format preferences, as described in the chapter Controlling International Preferences, and on the currency that the amount represents.

Some display settings, such as the characters that are used for the thousands and decimal separators, are based on your language preference. Others, such as the currency symbol and the decimal precision, are based on the properties of the currency in which the amount is being displayed. Those properties include:

- A currency symbol, such as \$ for the Australian dollar or L for the Italian lira.
- A decimal precision. For example, there are two decimal positions for Australian dollars: \$5.00, but no decimal positions for the Italian lira: 500 L.

This information is stored in the Currency Code table along with other information about all currencies used in your implementation.



CURRENCY_CD_TBL (Record)						
Record Fields		Record Type				
Num	Field Name	Type	Len	Format	Short Name	Long Name
1	CURRENCY_CD	Char	3	Upper	Currency	Currency Code
2	EFFDT	Date	10		Eff Date	Effective Date
3	EFF_STATUS	Char	1	Upper	Status	Status as of Effective D
4	DESCR	Char	30	Mixed	Descr	Description
5	DESCRSHORT	Char	10	Mixed	Short Desc	Short Description
6	COUNTRY	Char	3	Upper	País	País
7	CUR_SYMBOL	Char	4	Mixed	Symbol	Currency Symbol
8	DECIMAL_POSITIONS	Nbr	2		Decimals	Decimal Positions
9	SCALE_POSITIONS	Nbr	1		Scale	Scale Positions

CURRENCY_CD_TBL (currency code table) record definition

The currency symbol is controlled by the **CUR_SYMBOL** field.

The decimal precision for a currency is controlled by the **DECIMAL_POSITIONS** field.



PeopleSoft provides updates to the CURRENCY_CD_TBL containing the latest values from the ISO Standard 4217, “Codes for the representation of currencies,” with each PeopleTools minor release. See your PeopleSoft Installation and Administration Guide for details on how to load these updates. Even though PeopleSoft provides these standardized values, we do not provide a full list of the symbols used by each currency, as this is not part of the ISO standard. Use the Currency Code page to check and set the appropriate currency symbol for each currency you plan to use in your implementation.



The navigation to the page where you maintain currency code information depends on which applications you are using. For more information about maintaining currency codes, see PeopleSoft Application Fundamentals for your product line.

Setting Up a Currency Amount Field

To set up a page field for displaying international currency amounts, you must create an association between the currency amount field and a currency control field in the same record. The currency control field holds a currency code value indicating the currency represented in the amount field and associating it with a row in the Currency Code table.

The currency amount field is a number field, the length of which can vary, depending on the decimal precisions of the currencies supported in your implementation. For example, PeopleSoft delivers currency amount fields in the PeopleSoft General Ledger application with lengths of 13.2 (that is, 13 integer positions and 2 decimal positions). You can resize these fields if your currency requirements exceed this size (for example, to 15.3).



For more information about resizing currency amount fields, see Resizing Currency Fields Using the International Field Size Report.

To set up a currency amount field:

1. Make sure that the record definition has an appropriate currency control field.

The currency control field must be in the same record as the currency display field. It must hold a valid Currency Code value, so it must be a *Char* (character) field with a length of 3, formatted in *Upper* (uppercase). You can use an existing field or define a new field. The field does not need to be named CURRENCY_CD; it can have any valid PeopleSoft field name.

For example, the CURRENCY_EXCHNG (currency exchange) record has two currency fields, FROM_AMT (from amount) and CONVERTED_AMT (converted amount), each of which has a currency control field: FROM_CUR (*from currency*) and TO_CUR (*to currency*), respectively.

Num	Field Name	Type	Len	Format	Short Name	Long Name
1	FROM_AMT	Nbr	15.3		From Amt	From Amount
2	FROM_CUR	Char	3	Upper	From Cur	From Currency Code
3	TO_CUR	Char	3	Upper	To Cur	To Currency Code
4	CUR_RT_TYPE	Char	5	Upper	Rate Type	Exchange Rate Type
5	EFFDT	Date	10		Eff Date	Effective Date
6	CONVERTED_AMT	Nbr	15.3		Conv Amt	Converted Amount

CURRENCY_EXCHNG record definition

- Make sure that the edit properties of the currency control field are set up correctly.

The currency control field must hold a valid currency code, so it must edit against the Currency Code table. Set up the edit properties in the Record Field Properties dialog box, as shown here:

Record Field Properties

Use | Edits

Field Name: FROM_CUR

☒ Required

Edit Type:

☐ No Edit ☒ Table Edit

Table Edit:

Type: Prompt Table Edit

Prompt Table: CURRENCY_CD_TBL

Set Control Field:

OK Cancel

Record Field Properties dialog box - Edits tab for the currency control field

- Set the Currency Control Field property of the currency amount record field.

You must associate the field containing the amount with the currency control field.

Open the Record Field Properties dialog box for the currency amount field by right-clicking the record field in the record definition and selecting Record Field Properties from the pop-up menu.

Record Field Properties

Use | Edits

Field Name: FROM_AMT

Keys

- ☐ Key
- ☐ Duplicate Order Key
- ☐ Alternate Search Key
- ☐ Descending Key
- ☐ Search Key
- ☐ List Box Item
- ☐ From Search Field
- ☐ Through Search Field
- ☐ Default Search Field

Record Field label ID

Use Default Label

Default Value

Constant: 1

or

Record Name:

Field Name:

Currency Control Field

FROM_CUR

Default Page Control

System Default

Audit

- ☐ Field Add
- ☐ Field Change
- ☐ Field Delete

☐ System Maintained

☐ Auto Update

OK Cancel

Record Field Properties - Use tab for the currency field

In the **Currency Control Field** drop-down list, select the appropriate currency control field for the currency amount field.

4. Click **OK** to accept the property settings.
5. Save the record definition.

Setting Currency Field Display Properties

When you place a currency amount field on a page, you can choose whether to display the currency symbol and whether to display a thousands separator.

To set currency field display properties:

1. Display the field's Page Field Properties dialog box.

To open the Page Field Properties dialog box for the currency field, right-click on the page field and select Page Field Properties from the pop-up menu.

Page Field Properties - Record tab

2. Set the currency display options.

Select the **Currency Symbol** check box to display the currency symbol (as defined for the currency in the Currency Code table). Clear the check box if you don't want to display the symbol.

Select the **1000 Separator** check box if you want the currency amount to be displayed using a 1000 separator character. The specific character used as the 1000 separator is determined by your language preference.



For more information about how your language preference controls the 1000 separator character, see [Applying Browser Language Preferences to Determine Formatting Rules](#).

3. Click **OK** to accept the dialog box settings.

4. Save the page definition.

Using System-Wide Multi-Currency Settings

The Multi-Currency check box on the PeopleTools Options page is a system-wide switch that enables automatic formatting of those currency amount fields that have associated currency control fields.

It also controls validation of user entry against the currency's defined decimal precision and will cause the system to issue an error if the user attempts to enter a decimal precision that is greater than the precision allowed by the currency code definition.

A third function of this setting is to globally display currency control fields. You can design an application so that fields that are specifically related to multiple currencies can be globally displayed or hidden in different implementations. If you implement the application in an enterprise that uses only a single currency, you can hide all the fields by clearing the **Multi-Currency** check box. If the enterprise later begins to use multiple currencies, you can redisplay the multi-currency fields by selecting the check box.

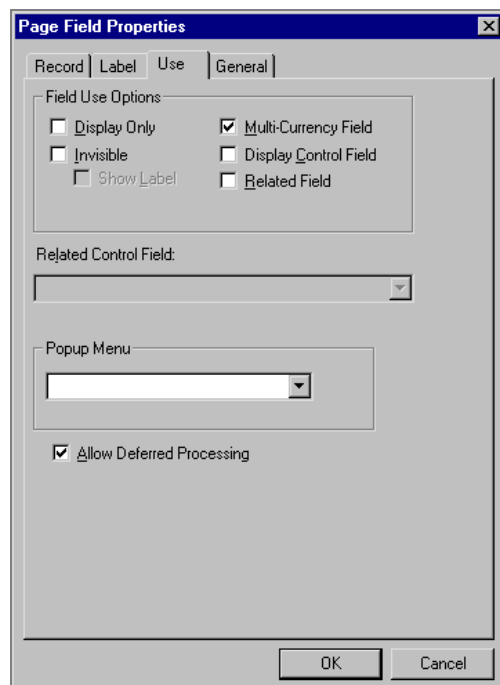
You should apply this technique only to fields that can be globally hidden without affecting the functionality of the application when working in a single currency. It will generally apply to user-operated currency control fields, but not to fields that actually display currency. You would also need to set the defaults of the currency display fields to the single currency used by the enterprise.

Under most circumstances, you will leave **Multi-Currency** selected.

To connect a currency control field to the system-wide multi-currency switch:

1. Display the field's Page Field Properties dialog box.

To open the Page Field Properties dialog box for the currency field, right-click on the page field, and select Page Field Properties from the pop-up menu. In the Field Properties dialog box, select the Use tab.



Page Field Properties - Use tab

2. Select the Multi-Currency Field option.

Select the **Multi-Currency Field** check box to specify that this field can be displayed or hidden from the PeopleTools Options page.

3. Click **OK** to accept the dialog box settings.
4. Save the page definition.

To enable or disable system-wide currency settings:

1. Select PeopleTools, Utilities, Use, PeopleTools Options.

This displays the PeopleTools Options page.

PeopleTools Options page

2. Select or clear the **Multi-Currency** check box.

Selecting the check box turns on automatic currency formatting; clearing the check box turns off automatic currency formatting.

If any page fields have the Multi-Currency Field option selected in the Page Field Properties dialog box, selecting the **Multi-Currency** check box displays those page fields; clearing the check box hides the page fields.

3. Save the PeopleTools Options page.

Resizing Currency Fields Using the International Field Size Report

If you process low-value currencies, which require numeric fields that are longer than those provided in your standard application, you can use the Set International Field Sizes (TLSINST1)

SQR report to increase the length and number of decimal positions of the fields throughout your database.

The Set International Field Sizes process updates the length and decimal position settings of field definitions stored in the PSDBFIELD PeopleTools table. To find out the new length and decimal-position settings for the field definitions, it uses values that are stored in the International Field Sizes table (PS_INTL_FLDSIZ_TBL).

The process does not update page field information (stored in the PSPNLFIELD table); however, the system automatically adjusts the sizes of any page fields whose Size property is set to Average or Maximum. If the page field Size is set to Custom, it will not be adjusted. If the page field is too small, the larger amount can still be entered; however, users may have to scroll to the right to see the full amount, and the truncated amount may be misleading.

Page field size is average

The Set International Field Size process shows all of the page fields that have been affected by the update. After running the process, you will need to check each page that was modified to make sure that the change in field size has not caused page fields to overlap. You will also need to use the Application Designer Build feature, with the Alter Tables option, to build (using SQL ALTER) any tables that have been affected by the changes in field size.

PeopleTools delivers the International Field Sizes table populated with suggested lengths for numeric fields that may require a large number of digits for certain currencies. You can edit the contents of this table to suit your own requirements, adding and deleting fields, and adjusting the new field sizes and decimal positions as necessary.

To resize international currency fields:

1. Start the PeopleTools International Field Size utility.

Select **PeopleTools, Utilities, Use, International Field Size** to displays the International Field Size page:

International Field Size				
List of Fields		View All First 1-15 of 23 Last		
*Field Name	Current Field Size	*Field Size - International		
ANNL_BENEF_BASE_R	15.3	12.2	+	-
ANNUAL_RT	15.3	12.2	+	-
APPL_EXPENSE_SUM	7.2	12.2	+	-
BUS_EXPENSE_SUM	7.2	12.2	+	-
CHANGE_AMT	12.6	12.3	+	-
COMPRATE	12.6	12.3	+	-
COST_PER_STUDENT	15.3	12.0	+	-
EXPENSE_AMT	15.3	12.2	+	-
HOURLY_RT	12.6	9.3	+	-
MAX_RT_ANNUAL	15.3	12.0	+	-
MAX_RT_HOURLY	12.6	9.3	+	-
MAX_RT_MONTHLY	15.3	11.2	+	-
MID_RT_ANNUAL	15.3	12.0	+	-
MID_RT_HOURLY	12.6	9.3	+	-
MID_RT_MONTHLY	15.3	11.2	+	-
Save				

International Field Size Page

The International Field Size page shows the name of each field that will be adjusted, the current size of the field, as stored in the PSDBFIELD table, and the new size of the field, stored in the International Field Sizes table.

2. Edit the data in the International Field Sizes table.

You can adjust the new field sizes to meet your own requirements. You can also insert or delete rows from the page.

3. Save your changes to the International Field Sizes table.
4. Run the Set International Field Sizes process.

Click **Run** to open the Process Scheduler Request page. Set the report options, and then click **OK** to run the TLSINST1 SQR.

The report updates the field definitions stored in the PSDBFIELD table and creates a report showing all page fields that have been impacted by the changes in field size.

5. Check all affected pages, and rearrange page fields as necessary to rectify overlap.

Check each page where the page field size has changed, and rearrange the fields on the page as necessary to prevent page fields from overlapping. If the page fields are too small to display the entire number, users will still be able to edit and view numbers by scrolling within the field.

6. Rebuild (SQL ALTER) any tables that have been affected by the changes in field sizes.

You will need to rebuild (SQL ALTER) any SQL tables that contain the fields whose size has changed. You can use the Find Object References features to determine which tables have been affected, or you can create a query against the PSRECDEFN, PSRECFIELD, and PS_INTL_FLDSIZ_TBL tables; for example:

```
SELECT DISTINCT A.RECNAME
FROM PSRECDEFN A, PSRECFIELD B, PS_INTL_FLDSIZ_TBL C
WHERE A.RECNAME = B.RECNAME
      AND B.FIELDNAME = C.FIELDNAME
      AND A.RECTYPE = 0;
```


CHAPTER 12

Market Rates and Currency Conversion

Although the implementation of market rates processing is, in the final analysis, application-specific, PeopleSoft offers a core set of objects (fields, tables, work records, pages, and PeopleCode functions), as well as a recommended set of standard techniques and formulas, to support a common approach to defining and storing market rates and to converting currency throughout PeopleSoft applications.

Market Rates Functional Requirements

The PeopleSoft approach to market rates and currency conversion is driven by the need to accommodate a number of important global-market business practices:

- Maintenance of international market rates that include interest rates, stock exchange indexes, and so forth, in addition to exchange rates.
- Use of direct and indirect quotation methods and avoidance of calculated reciprocal rates.
- Use of triangulated currency conversions, in which currencies are converted through a third Reference currency. This requirement is essential for converting between denominations of the euro.
- Tolerance checking during maintenance of market rates.

The following topics describe these functional requirements in greater detail.

Market Rates

Businesses use many types of market rates in addition to currency conversion rates. Some examples are:

- Interest rates
- Stock exchange indexes
- Economic indicators
- Commodity prices

You can capture data for each type of rate from many different sources. For example, you can download exchange rates from Reuters, the Financial Times, or the Federal Reserve.

PeopleSoft provides support for creating a common repository within applications for multiple types of market rate data.

Quotation Method

PeopleSoft supports both direct and indirect quotation of exchange rates.

To state the matter in practical terms, when you convert currency using a directly quoted exchange rate, you multiply the From currency by the exchange rate to get the equivalent amount in the To currency. When you convert currency using an indirectly quoted exchange rate, you divide the From currency by the exchange rate to get the equivalent amount in the To currency.

For example, the French franc (FRF) is most commonly quoted *directly* against the U.S. dollar (USD). That is, the conversion is stated in terms of how many French francs it takes to equal one U.S. dollar; for example, 5.8 FRF = 1 USD. Conversely, if the FRF-to-USD exchange rate is 5.8, then 1 USD = 5.8 FRF. When you convert from USD to FRF, you multiply by the exchange rate; when you convert from FRF to USD, you divide by the exchange rate.

With the introduction of the euro, all currencies are quoted directly against the euro. In order to achieve euro compatibility, PeopleSoft supports direct and indirect quotation of rates. When converting to the euro, you divide by the rate; when converting from the euro, you multiply by the rate.

Avoiding Calculated Reciprocal Rates

Calculated reciprocal rates enable an accounting system to use the direct quotation method for all currency conversions by calculating a reciprocal rate to use in the conversion formula when the From and To currencies are reversed. For example, the calculated reciprocal of 1 USD = 5.8 FRF would be 1 FRF = 0.172414 USD: in each case, you can multiply the From currency by the rate to get the equivalent amount in the To currency.

Use of calculated reciprocal rates introduces rounding errors and presents a user interface that is inconsistent with common business practice. PeopleSoft support for indirect and direct quotation methods allows applications to eliminate use of calculated reciprocal rates by using a single rate by which you either divide or multiply, depending on whether the conversion method is direct or indirect.

However, PeopleSoft continues to provide support for calculated reciprocal rates if your implementation requires them.

Quotation Units

PeopleSoft supports quotation units, sometimes called scaling factors, which are often used to preserve more decimal precision. For example, the exchange rate between Swiss francs (CHF) and Deutsche marks (DEM) can be stated as:

$$100 \text{ CHF} = 119.335 \text{ DEM}$$

instead of

1 CHF = 1.19335 DEM

The use of quote units is common business practice in several European countries.

Triangulation

Triangulation is the process by which a conversion between two currencies takes place through a third *Reference currency*. This process may be used in hyperinflationary environments, where all conversions to the local currency are done through a stronger, more stable currency such as the U.S. dollar. Also, this process may be used when a country is undergoing a currency revaluation.

With the introduction of the euro, any conversion between two participating currencies (denominations of the euro) must be triangulated through the euro. The exchange rates between the euro and the participating currencies are irrevocably fixed. The only officially published rates are between non-participating currencies and the euro. Triangulation is optional when you are converting between a participating and a non-participating currency (for example, when converting between French francs and U.S. dollars).

To support triangulation, PeopleSoft provides a means to define, for a currency pair, that it is triangulated through a fixed Reference currency. Then the actual conversion process is done in a two-step procedure, where the From currency amount is first converted to the Reference currency and then converted to the destination currency, using the appropriate exchange rates. Supporting triangulation also has an impact on the user interface, as there are now two, and possibly three, exchange rates that are relevant to the conversion.

For example, to convert from USD to FRF through the euro, PeopleSoft performs a two-step conversion, first converting the USD amount to euro, using the appropriate spot rate, and then converting the euro amount to FRF, using the fixed exchange rate.

Tolerance Checking

It is common for applications to support tolerance checking (against user-specified tolerances) in all places where exchange rates can be entered or overridden. With the introduction of indirect quotation methods and quote units, tolerance checking is even more critical to ensure data entry accuracy.

Understanding the PeopleSoft Approach to Market Rates and Currency Conversion

The following topics provide an overview of changes in PeopleSoft beginning with release 7.5 that facilitate the new approach to market rates and currency conversion. These changes are designed to provide continued support for existing application functionality, while also providing data structures to support development, implementation, and customization that addresses the new functional requirements.

RATE_MULT, RATE_DIV, and the Visual Rate

Support for both direct and indirect currency quotations would create a potential for currency conversion formulas in applications to become significantly more complex. The application that performs the conversion would need to look up the quote method. Then, for a direct quote, it would multiply by the rate; for an indirect quote, it would divide by the rate. For triangulated currency pairs, there would need to be two conversion steps, one to divide by one rate and a second to multiply by another rate.

To avoid all of this conditional logic in the conversion formula, PeopleSoft provides two fields to store the conversion factor, RATE_MULT and RATE_DIV. The rate that the user enters is called the *visual rate*. This visual rate is generally stored in either RATE_MULT or RATE_DIV, based on the quote method. The quote units are stored in whichever field doesn't contain the visual rate. As a result, the formula for currency conversion remains consistent:

$$(\text{from-currency} / \text{rate-div}) * \text{rate-mult} = \text{to-currency}$$

This formula is also used for currency conversion in PeopleCode programs for online processing, as well as in SQR and COBOL processes.

The following table shows a few basic examples of how visual rates are transformed into RATE_MULT and RATE_DIV, according to the quote method and quote units for the currency pair:

Currency Pair	Quote Method	Quote Units	Visual Rate	RATE_MULT	RATE_DIV
USD to GBP	Indirect	1	1.6	1	1.6
GBP to USD	Direct	1	1.6	1.6	1
DEM to CHF	Indirect	100	119.335	100	119.335
CHF to DEM	Direct	100	119.335	119.335	100
USD to Euro	Indirect	1	1.25	1	1.25
Euro to FRF	Direct	1	6.8	6.8	1
USD to FRF	Dir/Trng/Euro	1	5.44	6.8	1.25
FRF to Euro	Indirect	1	6.8	1	6.8
Euro to USD	Direct	1	1.25	1.25	1
FRF to USD	Indir/Trng/Euro	1	5.44	1.25	6.8

In all cases, the visual rate for a currency pair remains the same, regardless of the direction. This is consistent with business standards. For a direct quoted rate, you multiply by the visual rate; therefore, the visual rate goes into RATE_MULT, and 1 (or the quote units) goes into RATE_DIV. For an indirect quoted rate, you divide by the visual rate; therefore, the visual rate goes into RATE_DIV, and 1 (or the quote units) goes into RATE_MULT.

The following examples show indirect quotation, direct quotation with quote units, and triangulation:

100 USD to GBP (indirect) = (100 USD / 1.6) * 1 = 62.50 GBP

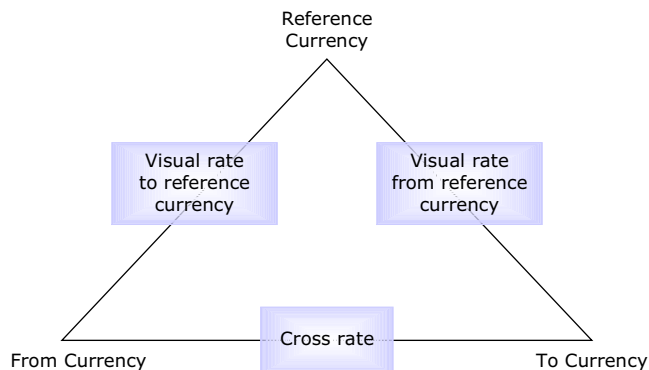
1000 CHF to DEM (direct w/ units) = (1000 CHF / 100) * 119.335 = 1193.35 DEM

100 USD to FRF (triangulate) = (100 USD / 1.25) * 6.8 = 544 FRF

Triangulated Rates

For a triangulated currency pair, such as USD to FRF in the preceding table, the values of RATE_MULT and RATE_DIV are determined by the rates of the From and To currencies in relation to the Reference currency. When viewing a triangulated conversion at the detail level, users access three visual rates:

- A rate for converting the From currency to the Reference currency.
- A rate for converting the Reference currency to the To currency.
- A *cross rate* indicating the rate that is required to convert the From currency directly into the To currency.



Visual rate components of a triangulated exchange rate

The visual rates are for display only and are not actually stored on the database. The cross rate is typically used as the *primary visual rate*; that is, the rate that the user sees and possibly overrides from the Market Rates main page. Users are given access to all three visual rates through the Rate Detail Page.

For performing the conversion, applications interpret the visual rates into RATE_MULT and RATE_DIV values, based on the quotation method for the exchange, and then use the RATE_MULT and RATE_DIV values that are stored in the Market Rates Data table in the currency conversion formula, either by accessing the values directly or by calling the PeopleCode ConvertCurrency function.



For more information about viewing visual rates, see [Market Rates Page](#) and [Rate Detail Page](#).

The cross rate in a triangulated conversion is not typically maintained directly. The system is designed to enable the user to maintain non-triangulated rates that are components of the triangulated rate and to run an SQR process to generate the triangulated exchange rate. (Users can, however, be given the option to override the cross rate, which causes one of the other exchange rate values to be recalculated to bring it in sync with the overridden cross rate.)

For example, suppose that an implementation was used triangulation to convert from U.S. dollars to French francs. The user would directly maintain the visual rate from U.S. dollars to euros (1.25 in the example table) and rate from euros to French francs (6.8 in the example table). The user could then run an SQR process to derive the triangulated rate for converting from USD to FRF. The results are shown in the following table:

Currency Pair	Quote Method	Quote Units	Primary Visual Rate	RATE_MULT	RATE_DIV
USD to Euro	Indirect	1	1.25	1	1.25
Euro to FRF	Direct	1	6.8	6.8	1
USD to FRF	Dir/Trng/Euro	1	5.44	6.8	1.25



The SQR process used to generate triangulated rates is the Cross-Rate Reciprocal SQR (EO9030.SQR). For information about how a specific application supports maintenance of triangulated exchange rates, see the documentation for that application.



For more information about triangulated conversion, see [Currency Quotation Method Page](#)). For more information about the PeopleCode `ConvertCurrency` function, see `ConvertCurrency`.

Market Rate Indexes

Market rate indexes, defined on the Market Rate Index table (RT_INDEX_TBL), provide the highest level of organization for market rates in the PeopleSoft system. These indexes define general categories of market rates, such as exchange rates, interest rates, commodity exchange rates, and so forth.

RT_INDEX_TBL (Record)						
Field Name	Type	Len	Format	H	Short Name	Long Name
RT_RATE_INDEX	Char	10	Upper		Index	Market Rate Index
RT_CATEGORY	Char	2	Upper		Category	Rate Category
DESCR	Char	30	Mixed		Descr	Description
DECIMAL_POSITIONS	Nbr	1			Decimals	Decimal Positions
DEFAULT_INDEX	Char	1	Upper		Default	Default Exchange Rate Index

Record definition of the Market Rate Index table

Users can define one index as the default exchange rate index on the Market Rate Index setup page. The logic on that page ensures that only one exchange rate index can be defined as the default exchange rate index.



The Market Rates Index page does not ensure that a default index has been defined, but it will be obvious if no default has been defined because the Market Rate Default view will return no data.



For more information about the Market Rates Index setup page, see [Market Rate Index Page](#). For more information about the Market Rate Default view, see [Market Rate Default View](#).

Market Rate Data

The Market Rate Data table is the common repository for all types of exchange rates, including exchange rates, interest rates, market indexes, and so forth. This table stores effective-dated information about each market rate, including its index, term, type, and From and To currencies, as well as RATE_MULT and RATE_DIV.

RT_RATE_TBL (Record)						
Field Name	Type	Len	Format	H	Short Name	Long Name
RT_RATE_INDEX	Char	10	Upper		Index	Market Rate Index
TERM	Nbr	5			Term	Term
FROM_CUR	Char	3	Upper		From Cur	From Currency Code
TO_CUR	Char	3	Upper		To Cur	To Currency Code
RT_TYPE	Char	5	Upper		Rate Type	Rate Type
EFFDT	Date	10			Eff Date	Effective Date
RATE_MULT	Sign	7.8			Multiplier	Rate Multiplier
RATE_DIV	Nbr	7.8			Divisor	Rate Divisor

Record definition of Market Rate Data table

For certain categories of market rates (that is, market rate indexes), such as interest rates, FROM_CUR and TO_CUR may have the same value.

During online maintenance of market rates, users don't view or change RATE_MULT and RATE_DIV directly; instead, they access the visual rate, which is calculated by the page logic, based on RATE_MULT, RATE_DIV, and the quotation method that is defined for the currency

pair. If no quotation method is defined for the currency pair, the page logic uses default quotation method values. The visual rate is stored temporarily on a page work record.

When maintaining triangulated rates, the user can access a primary visual rate (which is typically, but not necessarily, the cross rate), or they can access all three visual rates through a secondary detail page.



For more information about quotation methods and default quotation method values, see Currency Quotation Method.

Interim Exchange Rate View

The former Currency Exchange Rate table (CUR_RT_TBL) has been removed from the database as a physical table and replaced by a new Market Rate Data table (RT_RATE_TABLE). To ensure that the existing functionality in applications continues to work, an *interim* Currency Exchange Rate view, called CUR_RT_TBL, has been built over the Market Rates Data table to *replicate* the old Currency Exchange Rate table.

The following tables show the data structures of the old Currency Exchange Rate table, the new Market Rate Data table, and the interim Currency Exchange Rate view:

CUR_RT_TBL (Rel.7 table)	RT_RATE_TBL (Rel.7.5/ 8.1 table)	CUR_RT_TBL (Rel.7.5/8.1 SQL view)
K FROM_CUR	K RT_RATE_INDEX	SELECT
K TO_CUR	K TERM	FROM_CUR,
K CUR_RT_TYPE	K FROM_CUR	TO_CUR,
K EFFDT	K TO_CUR	RT_TYPE,
EFF_STATUS	K RT_TYPE	EFFDT,
CUR_EXCHNG_RT	K EFFDT	'A',
	RATE_MULT	RATE_MULT /
	RATE_DIV	RATE_DIV
		FROM
		PS_RT_RATE_TBL
		WHERE
		RT_RATE_INDEX =
		{sub-select to get
		default index}
		AND TERM = 0

The Currency Exchange Rate interim view (CUR_RT_TBL) supports backward compatibility with existing applications. To take full advantage of the new features of the redesign, new applications use either the Market Rate Data table or the Market Rate Default view (RT_DFLT_VW), which selects rows from the Market Rates Data table. Both the Market Rates Data table and the Market Rate Default view return the individual RATE_MULT and RATE_DIV fields, rather than returning a calculated single rate.

Market Rate Default View

The Market Rate Default view (RT_DFLT_VW) is used in place of the Market Rate Data table by applications that don't allow users to control market rate indexes. It selects, from RT_RATE_TBL, the rows where TERM is equal to zero and where the index (RT_RATE_INDEX) has been defined as the Default exchange rate index on the Market Rate Index page.

Market Rate Type

The Market Rate Type table (RT_TYPE_TBL) is an edit table that stores data about market rate types; that is, subcategories within a market rate index. For example, some common types of exchange rate are Official Rate, Spot Rate, and Free Market Rate. This table replaces the Currency Rate Type table (CUR_RT_TYPE_TBL) used in releases prior to PeopleSoft 7.5.

Interim View of Market Rate Types

To support backward compatibility with applications that were developed prior to PeopleSoft 7.5, a view called CUR_RT_TYPE_TBL has been constructed to select rows from the Market Rate Type table. The following table shows the data structures of the old Currency Rate Type table, the new Market Rate Type table, and the new Currency Rate Type view of the Market Rate Type table.

<i>CUR_RT_TYPE_TBL</i> <i>(Rel. 7 table)</i>	<i>RT_TYPE_TBL</i> <i>(Rel.7.5/81 table)</i>	<i>CUR_RT_TYPE_TBL</i> <i>(Rel.7.5/8.1 SQL view)</i>
K CUR_RT_TYPE	K RT_TYPE	SELECT
K EFFDT	DESCR	RT_TYPE,
EFF_STATUS	DESCRSHORT	%DateIn('1900-01-01'),
DESCR		'A',
DESCRSHORT		DESCR,
		DESCRSHORT
		FROM
		PS_RT_TYPE_TBL

In customizations you can continue to use CUR_RT_TYPE_TBL, but it may be more efficient to use RT_TYPE_TBL, because the new table is not effective-dated.

Currency Quotation Method

A currency quotation method that is defined for an exchange rate stores data that determines how the application interprets a visual rate (or multiple visual rates, in the case of triangulated exchange rates) that is entered by a user into the RATE_MULT and RATE_DIV fields on the Market Rate Data table. Conversely, it also determines how the stored RATE_MULT and RATE_DIV values are interpreted in the visual rate that is displayed to the user. The quotation method can be direct or indirect and it can be a non-triangulated conversion or a triangulated conversion through a third Reference currency. The currency quotation method also determines the quotation units of the From currency.

It is not necessary to define a currency quotation method for every exchange rate. If, during maintenance of market rates, no quotation method is found for an exchange rate, the [page](#) logic assumes these defaults:

- The exchange rate is *direct*.
- The quotation units are equal to *1*.
- The exchange rate is *not triangulated*.



This use of defaults supports backward compatibility with previous exchange rate data, including calculated reciprocal rates, if your implementation requires them.

Market Rate Definition

Tolerances for market rates are defined on the Market Rate Definition table. This table stores data that defines, for each market rate, the maximum degree of variance that is accepted when a user maintains or overrides a market rate. The Error Type setting determines how the system responds when the maximum variance is exceeded. If the Error Type is *Warning*, the system warns the user but permits entry of the new value; if the Error Type is *Stop*, the system alerts the user and does not accept the new value; if the Error Type is *None*, no tolerance checking is performed.

RT_RATE_DEF_TBL (Record)					
Field Name	Type	Len	Format	H	Short Name Long Name
RT_RATE_INDEX	Char	10	Upper		Index Market Rate Index
TERM	Nbr	5			Term Term
FROM_CUR	Char	3	Upper		From Cur From Currency Code
TO_CUR	Char	3	Upper		To Cur To Currency Code
MAX_VARIANCE	Nbr	3.2			Max Var Maximum Variance
ERROR_TYPE	Char	3	Upper		Error Type Error Processing Type
INT_BASIS	Char	2	Upper		Basis Interest Basis

Record definition of Market Rate Definition table

For interest rates, the Market Rate Definition table also stores an Interest Basis (in the INT_BASIS field). Some possible values are:

- 30/360
- 30E/360
- Actual/360
- Actual/365
- Actual/Actual

These values are maintained in the Translate table for the INT_BASIS (interest basis) field.

Market Rate Definition Default View

The Market Rate Definition Default view, RT_DEF_DFLT_VW, selects rows from the Market Rate Definition table that have a Term of 0 and an index that is defined as the Default Exchange Rate index and a Term of 0.

Common Pages for Market Rate Setup and Maintenance

The following topics describe the common pages that PeopleSoft provides to support the setup and maintenance of market rates and currency conversion methods in applications.

These pages appear under Define Business Rules, Define General Options. The exact navigation depends on which PeopleSoft database you're using. For example, in a PeopleSoft HRMS database, you'll find these pages under **Define Business Rules, Define General Options, Setup**. In a PeopleSoft Financials/Supply Chain Management database, you'll find these pages under **Define Business Rules, Define General Options, Use**.

Currency Code Page

The Currency Code page enables users to set up and maintain data in the Currency Code table (CURRENCY_CD_TBL). PeopleSoft preloads the table with a set of codes, based on the International Standards Organization (ISO) 4217 currency codes.

Currency Code

Currency Code: FRF

Definition Find | View All First 1 of 1 Last

*Effective Date: 01/01/1900 *Status: Active + -

*Description: French Franc

Short Description: Franc

Currency Symbol:

Country: FRA France

Decimal Positions: 2

Scale Positions:

Save Return to Search Next in List Previous in List Add Update/Display Include History Correct History

Currency Code page

Effective Date	The effective date of this currency code definition. The default is the current date.
Status	The effective status of the currency code definition. The default is <i>Active</i> .
Description	Language-sensitive description of the currency.
Short Description	Language-sensitive short description of the currency.
Currency Symbol	The symbol for the currency; for example, \$ for Australian dollar or <i>L</i> for Italian lira.
Country	The country code for the country where the currency originates.
Decimal Positions	The number of decimal positions in the notation for the currency. For example, there are two decimal positions for Australian dollars: \$5.00, but no decimal positions in the Italian lira: 500 L.
Scale Positions	Enables you to display currency amounts in a format that suppresses low-order digits. For example, to display a currency in thousands, set the scale positions to 3 . This causes 7,000 lira to be displayed as 7 lira, but the lira are stored in the database as the full amount (7000). This also affects data entry: if the user enters a value of 7, the value is stored in the database as 7000.

Market Rate Index Page

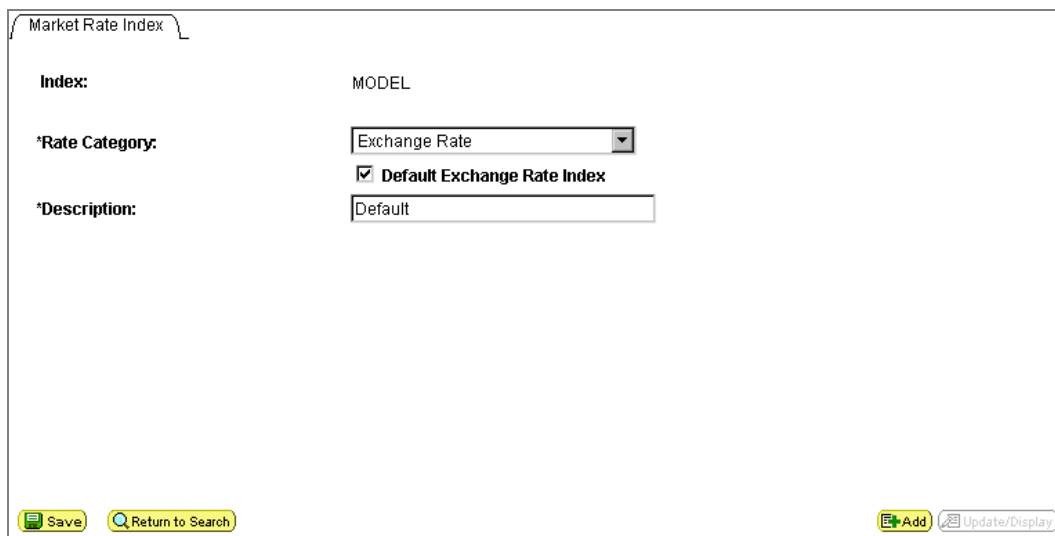
The Market Rate Index page enables users to set up and maintain data in the Market Rate Index table (RT_INDEX_TBL), which is an edit table that allows users to map indexes for market rates to broad categories, such as:

- Exchange rates
- Interest rates
- Economic indicators
- Commodity prices

These categories are maintained in the Translate (XLAT) table. Market Rate indexes are an application's highest level of organization for market rates.

This page also enables users to define a default index for exchange rates. This default setting is used by the Market Rate Default view and the Market Rate Definition Default view. These views are used by applications that do not allow user control of market rate indexes.

Here is a view of the Market Rate Index page:



Market Rate Index page

Index	Key term for the highest level of organization for market rates in the application.
Rate Category	General category for the market rate index, such as <i>Exchange Rate, Commodity Price, and Interest Rate.</i> These categories are maintained in the Translate (XLAT) table for the RT_CATEGORY field.

Default Exchange Rate Index	<p>This check box, when selected, specifies that the current Market Rate index is the default exchange rate index. This control is active only if the following are true:</p> <ul style="list-style-type: none"> • The Rate Category field is set to Exchange Rate. • No other index is currently defined as the Default Exchange Rate Index. (This rule ensures that only one index can be defined as the default.)
Description	A language-sensitive description for the Market Rate Index

Market Rates Page

Users can view and maintain market rates online using the Market Rates page.

The primary record for Market Rates page is the Market Rates Data table (RT_RATE_TBL). The Market Rates page also provides access to the Market Rate Definition table, and to currency quotation method and related data for the market rate in the Exchange Rate Work record (the data in this work record is largely derived from the Currency Quote Method table).

The following graphic shows the Market Rates page as it appears when the user is maintaining an exchange rate. With other rate categories (such as interest rates or commodity prices), the appearance of the page varies slightly.

Market Rates

Index: MODEL Default [Rate Definition](#)

Rate Category: Exchange Rate

Rate Type: OFFIC Official Rate

Term: 0

From Currency Code: USD US Dollar

To Currency Code: FRF French Franc

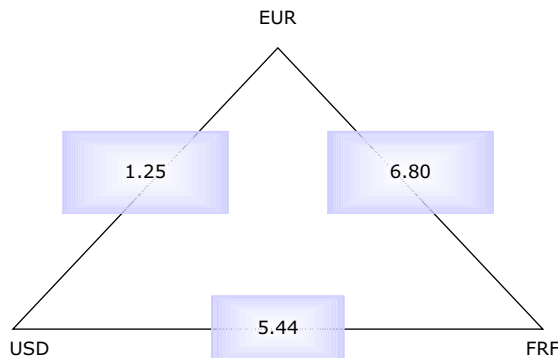
Rate		Find View All	First	1 of 1	Last
Effective Date:	01/01/1999				
Rate:	5.69061334				

Save Return to Search Add Update/Display Include History Correct History

Market Rates Page

The **Rate** field displays the *visual rate* or, in the case of triangulated exchange rates, the *primary visual rate*. Typically this is the *cross rate*, but it can also be one of the other component rates in the triangle. The following graphic represents the three component visual rates in an exchange

rate from FRF to USD, triangulated through the euro. The cross rate (the rate from USD to FRF) is used as the primary visual rate in the preceding screen shot.



Triangulated exchange rate from USD to FRF through the euro

The user can access all three visual rates of a triangulated exchange rate by opening the Rate Detail page.



For more information about the calculation of visual rates, see `RATE_MULT`, `RATE_DIV`, and the Visual Rate and Triangulated Rates.

The users can edit the **Rate** field, unless all of the following conditions are true:

- The rate is triangulated.
- The primary visual rate is the cross rate.
- The Allow Override check box is cleared for the exchange rate's quotation method on the Quotation Method page.



In typical implementations, users do not maintain triangulated exchange rates online. Instead, they maintain the rates of the From currency to the Reference currency, and the Reference currency to the To currency, and then run the Cross-Rate Reciprocal SQR (EO9030.SQR) to define the triangulated exchange rates.



For information about how to maintain triangulated exchange rates, see the documentation for your application.

If a quotation method has been defined for the currency pair, and if the Auto Reciprocate check box for the quotation method is selected, then creating or maintaining a rate for a currency pair automatically creates or updates the rate for the reciprocal currency pair. For example, if the user changes the USD-to-GBP rate, the GBP-to-USD rate is updated automatically. You can automatically reciprocate only currency pairs for which quotation methods have been defined.

If a rate definition does not exist for the currency pair, one will be automatically created with the default values of 2.5% maximum variance and *Warning* message processing. The user can use the pushbutton in the upper-right corner of the page to view or change the rate definition values.



The results of changing the rate definition do not take effect until you save, close, and then reopen the Market Rate page.

Rate Definition Page

The Rate Definition page, which users can access through the link in the upper-right corner of the Market Rates page, enables users to modify the Market Rate definition settings for the market rate. The following graphic shows the Rate Definition page for an exchange rate. The appearance of the page varies slightly for other market rate categories, such as interest rates.

Term	From Currency	To Currency	Maximum Variance	*Error Type
0	USD	FRF	2.50	Warning

OK Cancel

Rate Definition secondary page

Maximum Variance

The percentage of variance allowed when the user maintains the market rate. If the change exceeds the tolerance, an error results. The default value is **2.5%**.

Error Type

The type of error that results when the tolerance defined in the Maximum Variance field is exceeded during data entry. If this field is set to **Warning**, the system alerts the user but permits the change. If it is set to **Stop**, the system alerts the user and does not permit the change. If it is set to **None**, no tolerance checking is done. The default value is **Warning**.

Rate Detail Page



The Exchange Rate Detail page provides users with detailed information about an exchange rate. This page is accessible from the Market Rates main page through the Exchange Rate Detail button (although it can also be implemented using a pop-up menu).

The primary record for this page is the Exchange Rate Work record; however, the data is derived principally from the Quotation Method table. For triangulated rates, this page allows users to

update all three components of the triangulated rate, whereas in the main page, they can maintain only the primary visual rate (unless the Allow Override check box is cleared on the Quotation Method page for the exchange rate).

When the user is looking at a historic exchange rate, the rate is shown according to the *current quotation method*, which is not necessarily the quotation method that was originally used by the historic rate. This makes it convenient for the user to compare how the exchange rate has changed over time, using a consistent quotation method, even if the quotation method has changed. When the system determines that an exchange rate is inconsistent with the current quotation method defined for that exchange rate, it displays a historic quote in the Historic Quote field.

For example, if the user is viewing a historic rate, where French francs were converted to U.S. dollars directly, using a calculated reciprocal rate of 1 FRF = 0.1470588 USD, and the current quotation method for that currency pair is indirect, the conversion function recalculates the visual rate, based on indirect quotation; that is 6.8000001 FRF = 1 USD. This provides users with consistent visual rates when they are looking at historical data.

The following graphic shows the Exchange Rate Detail page as it appears for a triangulated currency pair. For non-triangulated currency pairs, the lower section of the page displays just a single visual rate:

Exchange Rate Detail
Rate Quotation Basis: Direct
Quote Units: 1
Triangulate: Y
Reference Currency: EUR
Current Quote:
 1.1527 USD = 1 EUR = 6.55957 FRF
Historic Quote:
 Not Applicable

From	To	Rate
USD	FRF	5.69061334
USD	EUR	1.15270000
EUR	FRF	6.55957000

Exchange Rate Detail page



For information about using this page in currency conversion processing, see Rate Detail Secondary Page.

Rate Quotation Basis

This read-only field displays the quotation basis for the exchange rate as defined on the Quotation Method page. If no quotation method is defined, the rate quotation basis is ***Direct***.

Quote Units

This read-only field displays the quote units for the exchange rate as defined on the Currency Quotation Method page. If no quotation method is defined, the quote units value is *I*.

Triangulate

This read-only field displays the triangulated setting for the exchange rate as defined on the Currency Quotation Method page. If no quotation method is defined, the Triangulate value is *N*.

Reference Currency

This read-only field, which displays for triangulated exchange rates only, shows the reference currency that is used in the triangulated exchange.

Current Quote

This read-only field displays the current exchange rate that is used to convert the From currency to the To currency.

A direct, non-triangulated rate shows quote units (or *I*) to the left of the equal sign and the visual rate to the right; for example:

$$1 \text{ USD} = 1.40000000 \text{ CAD}$$

An indirect, non-triangulated rate displays the visual rate to the left of the equal sign and quote units (or *I*) to the right; for example:

$$1.400000000 \text{ CAD} = 1 \text{ USD}$$

A triangulated rate displays two component rates of the triangle: the rate for converting the From currency to the Reference currency and the rate for converting the Reference currency to the To currency; for example:

$$1.25 \text{ USD} = 1 \text{ EUR} = 6.8 \text{ FRF}$$
Historic Quote

This read-only field displays a quote that is based on the quotation method that was originally used by a historic exchange rate, if the page logic determines that the exchange rate, as stored on the database, is inconsistent with the current quotation method. For example, the **Historic Quote** field displays a quote if the historic rate converted the From currency to the To currency directly, using a calculated reciprocal rate, but the current quotation method for the currency pair is now indirect. The field would also display a quote if the historic quote method were non-triangulated and the current quote method were triangulated. If the system determines that the exchange rate is inconsistent with the current quotation method, the **Historic Quote** field displays *Not Applicable*.



A historic quote is also displayed if a user overrides a cross rate and instructs the system to bypass triangulation, because the exchange rate being used is inconsistent with the current quotation method.

Exchange Rate

The **Exchange Rate** group box displays a single visual rate for non-triangulated exchange rates or all three component visual rates for triangulated exchange rates. The cross rate for triangulated exchange rates is editable only if the Allow Override check box is selected on the Quotation Method page for the exchange rate.



For more information, see Understanding the PeopleSoft Approach to Market Rates and Currency Conversion.

Currency Quotation Method Page

The Currency Quotation Method page enables the setup and maintenance of currency quotation methods for exchange rates. The currency quotation method for an exchange rate determines how the visual rate (the rate that is displayed on the Market Rates and Rate Detail pages) is converted into the RATE_MULT and RATE_DIV values stored on the database and, conversely, how the visual rate is derived from RATE_MULT and RATE_DIV.

The following screen shot shows the Currency Quotation Method page for a triangulated currency pair:

Currency Quotation Method

From Currency Code: USD US Dollar

To Currency Code: FRF French Franc

Quote Method Find | View All First 1 of 1 Last

Effective Date: 01/01/1999 Status: Active

Rate Quotation Basis

☒ Direct ☐ Indirect *Quote Units: 1

USD x.xxxx = EUR 1 = FRF y.yyyy Auto Reciprocate ☒

Triangulation Options

☒ Triangulate

Reference Currency: EUR

euro

Primary Visual Rate

☒ USD --> FRF

☐ USD --> EUR

☐ EUR --> FRF

Cross-Rate

☒ Allow Override

Recalculate

☒ USD --> EUR

☐ EUR --> FRF

Save Return to Search Add Update/Display Include History Correct History

Currency Quotation Method page



Users can view the currency quotation method for an exchange rate through the Rate Detail page, while working on the Market Rates page.



For more information about rates, see Rate Detail Page and Understanding the PeopleSoft Approach to Market Rates and Currency Conversion.

Effective Date	The effective date for the currency quote method; the default is the current date.
Status	The effective status of the currency quote method. The default value is <i>Active</i> .
Rate Quotation Basis	The Direct and Indirect options indicate whether the rates for this currency pair are direct or indirect quotes. For example, with USD to FRF, a <i>direct</i> quote means that 1 USD = x.xxxx FRF; an <i>indirect</i> quote means that x.xxxx USD = 1 FRF. Note that even currency pairs that are triangulated still have to be classified as either direct or indirect, for use in displaying the calculated cross-rate. The default is Direct .
Quote Units	This enables the user to apply a <i>scaling factor</i> to the exchange rate, as is common business practice for some currencies. This field can have any value, but usually is a power of 10 (10, 100, 1000, and so forth). The default value is <i>1</i> .
Auto Reciprocate	<p>When this check box is selected, the rate maintenance functions create or update the rate for the reciprocal currency pair whenever an exchange rate is added or updated. For example, when the user enters a new USD-to-GBP rate, the GBP-to-USD rate is updated automatically. You can automatically reciprocate only currency pairs for which quotation methods have been established. By default, this check box is selected.</p> <p>For more information, see Automatic Reciprocation of Quote Methods and Formula 9: Calculation of Reciprocal Rates.</p>
Triangulate	This check box indicates whether conversions between this currency pair must be triangulated through a third Reference currency.
Reference Currency	Specifies the Reference currency for a triangulated conversion.

Primary Visual Rate

With triangulated currency pairs, there are three exchange rates to consider:

- The rate between the From currency and the Reference currency
- The rate between the Reference currency and the To currency
- The calculated cross-rate between the From currency and the To currency

The **Primary Visual Rate** options enable installations to select which of the three rates they want to be the primary visual rate. The primary visual rate appears on primary pages and reports. In online applications, the other components of the rate can be viewed and modified with the Exchange Rate Detail page. By default, the rate between the From and To currencies (the cross rate) is the primary visual rate.

Cross-Rate - Allow Override

With triangulated currency pairs, this check box, when selected, enables installations to specify whether users can override the cross rates or can only change the components of the triangulated rate. By default, this check box is cleared.

Cross-Rate - Recalculate

If the user is allowed to override the cross-rate for a triangulated currency pair, this option tells the application which of the two other rates should be recalculated to bring the triangle back into balance again. By default, the **USD→EUR** option is selected.

Automatic Reciprocation of Quote Methods

The Currency Quote Method is designed to reciprocate itself automatically. For example, if you define USD to FRF to be direct, then the FRF-to-USD row is automatically created or updated to indicate a quote method of indirect.



Note: The Auto Reciprocate check box on the Currency Quote Method page pertains to the maintenance of the exchange rates on the Market Rates page. The quote method itself always reciprocates automatically.

The following table shows each possible field value and its reciprocal value:

Field	Value (for example, USD to FRF)	Reciprocal Value (for example, FRF to USD)
Quote Basis	Direct	Indirect
	Indirect	Direct
Quote Units	(any valid value)	(same value)
Auto-Reciprocate	Yes	Yes
	No	No
Triangulate	Yes	Yes
	No	No
Reference Currency	(any valid value)	(same value)
Primary Visual	From → To (for example, USD → FRF)	From → To (for example, FRF → USD)
	From → Ref (for example, USD → EUR)	Ref → To (for example, EUR → USD)
	Ref → To (for example, EUR → FRF)	From → Ref (for example, FRF → EUR)
Cross-Rate Allow Override	Yes	Yes
	No	No
Cross-Rate Recalculation	From → Ref (for example, USD → EUR)	Ref → To (for example, EUR → USD)
	Ref → To (for example, EUR → FRF)	From → Ref (for example, FRF → EUR)

Market Rate Definition Page

The Market Rate Definition page is used for the setup of tolerance checking and the interest basis for market rates.

Tolerance checking is performed when a user attempts to enter or override a rate. If a comparison of the new value and the previously stored value for the rate exceeds a defined Maximum Variance for the rate, the system can do one of the following:

- Warn the user, but permit the user to enter or override the rate with the new value.
- Warn the user and prevent the user from entering or overriding the rate with the new value.
- Do nothing (in this case the system does not perform a tolerance check).

Interest basis applies only to interest rates. The setting controls the type of year and month basis used to calculate interest. The values for this field are maintained in the Translate table; some common settings are:

- 30/360
- 30E/360
- Actual/360
- Actual/365
- Actual/Actual

Rows on Market Rate Definition table are automatically created (or updated) when a rate is added or maintained in the Market Rates page. These rows have a default Maximum Variance of 2.5% and a default Error Type of *Warning*. Users can view and change the Market Rate definition settings for a market rate through the Rate Definition page, which they can access from the Market Rates page.



For more information, see Rate Definition Page.

The appearance of the Market Rate Definition page varies, depending on the Rate Category. Exchange rates display From Currency and To Currency fields; interest rates display a single Currency field as well as an Interest Basis field. All other pages display a single Currency field. The following screen shot shows the Market Rate Definition page for an exchange rate:

Rate Definition

Index: MODEL Default

Rate Category: Exchange Rate

From Currency Code:

Term	From Currency	To Currency	Maximum Variance	Error Type	
0	ADP <input type="button" value="Q"/>	ATS <input type="button" value="Q"/>	2.50	Warning	<input type="button" value="+"/> <input type="button" value="-"/>
0	ADP <input type="button" value="Q"/>	BEF <input type="button" value="Q"/>	2.50	Warning	<input type="button" value="+"/> <input type="button" value="-"/>
0	ADP <input type="button" value="Q"/>	CAD <input type="button" value="Q"/>	2.50	Warning	<input type="button" value="+"/> <input type="button" value="-"/>
0	ADP <input type="button" value="Q"/>	DEM <input type="button" value="Q"/>	2.50	Warning	<input type="button" value="+"/> <input type="button" value="-"/>
0	ADP <input type="button" value="Q"/>	ESP <input type="button" value="Q"/>	2.50	Warning	<input type="button" value="+"/> <input type="button" value="-"/>
0	ADP <input type="button" value="Q"/>	FIM <input type="button" value="Q"/>	2.50	Warning	<input type="button" value="+"/> <input type="button" value="-"/>
0	ADP <input type="button" value="Q"/>	FRF <input type="button" value="Q"/>	2.50	Warning	<input type="button" value="+"/> <input type="button" value="-"/>
0	ADP <input type="button" value="Q"/>	GBP <input type="button" value="Q"/>	2.50	Warning	<input type="button" value="+"/> <input type="button" value="-"/>

Market Rate Definition page

Maximum Variance

The percentage of variance allowed when the user maintains the market rate. If the change exceeds the tolerance an error results. The default value is **2.5%**.

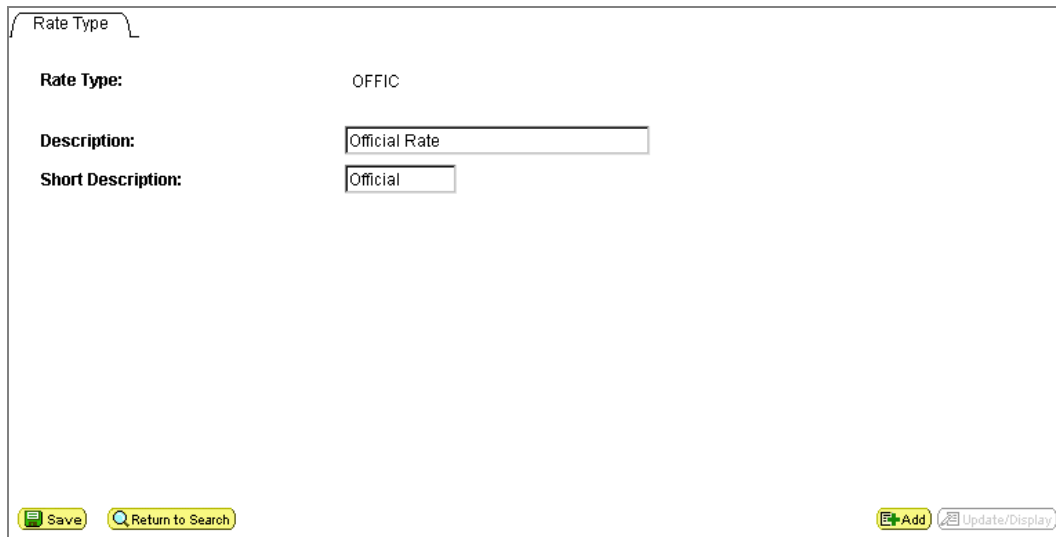
Error Type

The type of error that results when the tolerance defined in the **Maximum Variance** field is exceeded during data entry. If the field is set to **Warning**, the system alerts the user but permits the change. If it is set to **Stop**, the system does not permit the change. If it is set to **None**, the system does not perform tolerance checking. The default value is **Warning**.

Rate Type Page

The Rate Type page is used to set up and maintain data in the Rate Type table (RT_TYPE_TBL). The Rate Type table is an edit table that contains a list of valid market rate types and their descriptions.

The menu navigation for this page may be either Market Rate Type or Currency Rate type, depending on which PeopleSoft database you're using.



Rate Type

Rate Type: OFFIC

Description: Official Rate

Short Description: Official

Save Return to Search Add Update/Display

Rate Type page

Rate Type

Code for the rate type.

Description

Language-sensitive description of the rate type.

Short Description

Language-sensitive short description of the rate type.

PeopleSoft Common Objects for Currency Conversion

Storing the exchange rate in two fields, RATE_MULT and RATE_DIV, is ideal for conversion purposes and batch processing; however, it does add more overhead to online applications. To lessen the development burden, PeopleSoft provides a set of common functions, work records, and pages.

Exchange Rate Work Record

The central feature of currency exchange processing is the Exchange Rate work record, which refers to EXCH_RT_WRK or a copy of EXCH_RT_WRK.

This Exchange Rate work record, over which the Rate Detail page is built, stores all data related to exchange rate during the life of the Market Rate component. This helps to simplify and speed up processing. For example, once a quote method is retrieved, each time the user changes the visual rate, the application does not have to re-retrieve the quote method to process the visual rate.

The PeopleCode routines used for displaying and manipulating data on the page, as well as the common functions called by application-specific PeopleCode, are maintained on the work record.

The Exchange Rate work record holds the application-specific work record and field names for the visual rate, RATE_MULT, and RATE_DIV. The common PeopleCode functions retrieve and update the application specific fields, using the @ operator to convert the stored record and field name strings into record field references.



For more information, see @ Operator.

If an application has a page where more than one exchange rate appears at the same level, then duplicate versions of the work record will be required: one for each visual rate. To accommodate this, the Exchange Rate work record has been designed using a subrecord, EXCH_RT_WSBR. Because the actual work record needs only to contain the subrecord, you can create multiple copies of the work record. PeopleSoft provides two such copies: EXCH_RT_WRK2 and EXCH_RT_WRK3.

Rate Detail Secondary Page

The EXCH_RT_DTL page, described previously, provides the user with detailed information about the current exchange rate. Placing this page on an application-specific page enables PeopleCode access to all of the record fields on the Exchange Rate work record.



For more information, see Exchange Rate Work Record.

An inquiry version of this page, EXCH_RT_DTL_INQ, also provides access to the fields from the work record. You should use the inquiry version of the rate detail page whenever the corresponding Visual Rate field on the main page is defined as display-only in the page field properties. If the Visual Rate field is not defined as display-only, you should use the non-inquiry (update) version of the page. Then, when you initialize the detail page (using the InitializeDetailPage common function), you can set the input mode to either D (display) or U (update), depending on whether the Visual Rate field on the main page is currently enabled or disabled. If you follow these guidelines, you will keep the display mode of the page consistent with the display mode of the associated Visual Rate field.

PeopleSoft provides additional copies of these pages built over the additional copies of the work record (described in the preceding section). These copies are named EXCH_RT_DTL2, EXCH_RT_DTL2_INQ, EXCH_RT_DTL3, and EXCH_RT_DTL3_INQ.

The pages are comprised of two subpages. PeopleSoft provides one common subpage, EXCH_QUOTE_SBP, for the display-only quote information that appears on the top half of the page. The bottom half of the page uses either the update (EXCH_RT_UPD_SBP) or inquiry (EXCH_RT_INQ_SBP) version of the Rate subpage. The subpage design was implemented to facilitate the duplication that may be required if an application has a page with more than three exchange rates appearing at the same scroll level.

PeopleCode Built-in Functions

The PeopleCode ConvertCurrency and RoundCurrency functions provide easy ways to convert between and round international currencies.

ConvertCurrency

The PeopleCode ConvertCurrency built-in function provides the simplest means to convert currency on an online page because it does the work of retrieving values from the Market Rate tables.

Internally, ConvertCurrency uses the standard formula for performing currency conversion, using the RATE_MULT and RATE_DIV values that are stored on the Market Rate Data table:

$$(\text{from-currency} / \text{rate-div}) * \text{rate-mult} = \text{to-currency}$$

This function has an optional *rt_index* parameter to specify an exchange rate index. If this parameter is omitted, the function uses the index defined as the Default Exchange Rate index.



Online applications usually use the formula directly, rather than using the `ConvertCurrency` function. Using the formula directly gives developers more control over conversion processing; for example, it facilitates development of pages that permit users to override the rate or that store the exchange rate, in addition to the converted amount, in application tables.



For more information, see `ConvertCurrency`.

RoundCurrency

The `RoundCurrency` function rounds a currency to the decimal precision specified in its currency code definition in the Currency Code table.



For more information, see `RoundCurrency`.

Common PeopleCode Library Functions

The following topics provide reference information about the common PeopleCode functions used in currency conversion. These are PeopleCode *library functions*, rather than *built-in* functions, and they are stored either in the Exchange Rate work subrecord (`EXCH_RT_WSBR`, which is used on all copies of the Exchange Rate work record) or in the Currency Control function library (`FUNCLIB_CUR`).

These functions access and update values in an Exchange Rate work record, as well as an application-specific work record (which holds temporary values, such as formatted visual rates), and an application-specific record for holding values to be stored on the database, such as `RATE_MULT` and `RATE_DIV`.

When this document refers to the Exchange Rate work record, it means either `EXCH_RT_WRK`, or a copy, such as `EXCH_RT_WRK2`. The functions identify the record fields of the application-specific work record for `RATE_MULT`, `RATE_DIV`, and the visual rate using field names stored in the Exchange Rate work record.

Another essential function, which is not covered in this section, because it is application-specific, is `ProcessRateDetail`.



For more information about the `ProcessRateDetail` function, see [Function to Process the Detail Page](#). For more information about using PeopleCode library functions, see [Functions](#).

GetQuoteMethod

GetQuoteMethod is used to determine the quotation method in effect for a specific currency pair on the current (system) date.



All visual rates are calculated and displayed based on the *current quotation method* defined for the currency pair. Calculating the visual rate(s) based on the current method gives users a consistent view of exchange rates when they are looking at historical data.

The GetQuoteMethod function stores the From and To currencies, which are passed to it, on the Exchange rate work record. It performs a SQLExec to the Currency Quotation table and stores all the retrieved data on the work record.

If no quotation method exists for the currency pair, or if *from_currency* is equal to *to_currency*, GetQuoteMethod applies these defaults:

- Rate Quotation Basis = D
- Quote Units = 1
- Triangulate = N

Syntax

```
GetQuoteMethod(from_currency, to_currency)
```

Location

EXCH_RT_WRK.RATE_DIRECT FieldChange

Input Parameters

<i>from_currency</i>	The From currency code for the exchange rate.
<i>to_currency</i>	The To currency code for the exchange rate.

Return Values

None

Where Used

Called from RowInit for the Visual Rate field if From and To currency values are available. Also called whenever the From or To currency values change, either in the FieldChange event of a user-maintainable field or when a currency code value is manipulated internally (for example, in GL when the ledger changes, causing the base currency to change).

GetMaxVariance

Performs a SQLExec to the Market Rate Definition table (RT_RATE_DEF_TBL) or Market Rate Definition Default view (RT_DEF_DFLT_VW) to retrieve the MAX_VARIANCE and ERROR_TYPE values for the From and To currencies already stored on the work record. For triangulated currency pairs, data will be retrieved for all sides of the triangle. The MAX_VARIANCE and ERROR_TYPE values are stored on the Exchange Rate work record.

Syntax

```
GetMaxVariance(rate_index)
```

Location

EXCH_RT_WRK.RATE_DIRECT FieldChange

Input Parameters

<i>rate_index</i>	The rate index associated with the exchange rate. If the value passed is an empty string, the function queries RT_DEF_DFLT_VW, which selects only data for the default index. Otherwise, the function queries the Market Rate definition table (RT_RATE_DEF_TBL), using <i>rate_index</i> as a selection criterion.
-------------------	---

Return Values

None.

Where Used

Typically called when GetQuoteMethod is called and the application has the ability to override rates. In an inquiry, where the rates are display-only, GetQuoteMethod is called without GetMaxVariance.

DetermineVisualRate

DetermineVisualRate uses quotation method data stored on the Exchange Rate work record to calculate and format a visual rate based on the *rate_mult* and *rate_div* parameters.

Syntax

```
DetermineVisualRate(rate_mult, rate_div)
```

Location

EXCH_RT_WRK.RATE_MULT FieldChange

Input Parameters

rate_mult and *rate_div*

The values that you want to format into a visual rate, using the currency quote method values stored on the work record.

Return Values

A Number value equal to the formatted visual rate.

Where Used

This function is called only in cases where you don't want to store RATE_MULT and RATE_DIV on the work record. It is normally called only from within SetVisualRate.

SetVisualRate

SetVisualRate formats RATE_MULT and RATE_DIV on the Exchange Rate work record by accessing values from the application-specific work record. It identifies the application-specific work record using the record and field names stored on the Exchange Rate work record. It calls DetermineVisualRate, passing RATE_MULT and RATE_DIV as parameters, which returns the formatted visual rate.

Syntax

```
SetVisualRate()
```

Location

EXCH_RT_WRK.RATE_MULT FieldChange

Input Parameters

None.

Return Values

A Number value equal to the formatted visual rate.

Where Used

Called from RowInit for the Visual Rate field, if From and To currency values are available, and RATE_MULT and RATE_DIV have values. Also called within the UpdateRates function, which sends the visual rate back to the application-specific work record.

GetExchangeRate

GetExchangeRate calls the RetrieveExchangeRate function stored in FUNCLIB_CUR, which performs a SQLExec to the Market Rate Data table (or Market Rate Default view) to retrieve the RATE_MULT and RATE_DIV values. The returned values are stored on the Exchange Rate work record. GetExchangeRate also calls the UpdateRates function to format the visual rate and

then updates the RATE_MULT, RATE_DIV, and Visual Rate fields on the application-specific work record. It identifies application-specific work record fields, using the record and field names stored on the Exchange Rate work record.

Syntax

```
GetExchangeRate(rate_index, from_cur, to_cur, rate_type, effdt)
```

Location

EXCH_RT_WRK.RATE_MULT FieldChange

Input Parameters

<i>rate_index</i>	The Market rate index for the exchange rate. If no rate index is passed, the function queries the Market Rate Default view, which selects only rates that have the Default Exchange Rate index. If a rate index is passed, the function queries the Market Rate Data table, using the rate_index as a selection criterion.
<i>from_cur</i>	The exchange rate's From currency.
<i>to_cur</i>	The exchange rate's To currency.
<i>rate_type</i>	The exchange rate's rate type.
<i>effdt</i>	The exchange rate's effective date.

Return Values

None.



Although this function has no actual return value, it does update the RATE_MULT, RATE_DIV, and VISUAL_RATE fields on the application-specific record(s).

Where Used

Called whenever the currency codes, rate type, rate index, or rate effective date changes, provided all of the key values are available. Also called when the visual rate is blanked out, again provided that all of the key values are available.

RetrieveExchangeRate

RetrieveExchangeRate performs a SQLExec to the Market Rate Data table (or Market Rate Default view) to retrieve the RATE_MULT and RATE_DIV values, which are returned to the calling routine through the parameter list.

Syntax

```
RetrieveExchangeRate(rate_index, from_cur, to_cur, rate_type, effdt, rate_mult,
rate_div)
```

Location

FUNCLIB_CUR.RATE_MULT FieldFormula

Input Parameters

<i>rate_index</i>	The Market rate index for the exchange rate. If an empty string is passed in this parameter, the function queries the Market Rate Default view, which selects only rates that have the Default Exchange Rate index. If a rate index is passed, the function queries the Market Rate Data table, using the rate as a selection criterion.
<i>from_cur</i>	The exchange rate's From currency.
<i>to_cur</i>	The exchange rate's To currency.
<i>rate_type</i>	The exchange rate's rate type.
<i>effdt</i>	The exchange rate's effective date.

Output Parameters

<i>rate_mult</i> and <i>rate_div</i>	The output parameters, which can be either variables or record fields, are filled with the RATE_MULT and RATE_DIV exchange rate values retrieved from the Market Rate Default view or Market Rate Data table.
--------------------------------------	---

Return Values

None.

Where Used

This standalone function is used to retrieve exchange rate fields in cases where applications do not need to display the exchange rate. To display the exchange rate, applications should use the Exchange Rate work record and the GetExchangeRate function on that work record.

This function is called by the GetExchangeRate function.

CheckRateVariance

CheckRateVariance compares two rates, in visual rate format, to determine if the variance between the two rates exceeds the tolerance defined for the currency pair. It returns a Boolean value indicating whether the maximum value was exceeded.

It is the responsibility of the application to call the function, look at the return value, and display an error or warning message.



The function uses MAX_VARIANCE and ERROR_TYPE fields of the Exchange Rate work record to check the tolerance and error type defined for the exchange rate, rather than querying the Market Rate Definition table or view.

Syntax

```
CheckRateVariance(new_rate, old_rate, source)
```

Location

EXCH_RT_WRK.RATE_MULT FieldChange

Input Parameters

<i>new_rate</i>	The new entered rate that has been changed by the user or by some other means.
<i>old_rate</i>	The previously stored value for the visual rate.
<i>source</i>	When this function is called from applications, always pass an empty string in this parameter. Other values are for internal use in the Rate Detail page logic.

Return Values

Returns a Boolean value indicating whether the rate variance exceeds the tolerance specified in the MAX_VARIANCE field of the work record.

- It returns False if the new rate differs from the old rate by more than the defined maximum tolerance and the error type is either *Warning* or *Error*.
- It returns True if any of the following is the case:
 - The error type is *None* (in which case the function doesn't even check the tolerance)
 - The *old_rate* was 0, so there was no basis for comparison
 - The new rate differs from the old rate by an amount less than or equal to the maximum variance.

PeopleSoft suggest use of the variable name &RATE_OK to store the return value.

Where Used

Called from FieldEdit on the Visual Rate field.

ProcessVisualRateChange

ProcessVisualRate first determines the values of RATE_MULT and RATE_DIV, based on a newly entered visual rate, and then updates the fields on both the Exchange Rate work record and the application-specific record. The function identifies the application-specific record fields, using the record and field names that are stored on the Exchange Rate work record.

If the visual rate is a cross rate of a triangulated pair, the function prompts the user, asking whether the user wants to bypass triangulation and use the rate exactly as entered or recalculate the component rates, in which case the visual rate (that is, the cross rate) may be slightly altered because of rounding.

If a currency pair is triangulated, and one of the results has a value of 0 after processing, the function displays a warning message telling users that they must set the rate in the Rate Detail page.

Syntax

```
ProcessVisualRateChange(visual_rate)
```

Location

EXCH_RT_WRK.RATE_MULT FieldChange

Input Parameters

visual_rate The modified visual rate.

Return Values

None.



Although this function has no actual return value, it does update the RATE_MULT, RATE_DIV, and VISUAL_RATE fields on the Exchange Rate work record and on the application-specific record(s).

Where Used

Called from FieldChg (Field Change PeopleCode) on the Visual Rate field.

InitializeDetailPanel

InitializeDetailPanel retrieves the values of RATE_MULT and RATE_DIV from the application-specific work record and then populates those fields in the Exchange Rate work record that are specific to the Rate Detail page, using other values that are stored on the Exchange Rate work record.

InitializeDetailPanel uses the record and field names stored on the Exchange Rate work record to identify the RATE_MULT and RATE_DIV record fields on the application-specific work record.



This function makes it unnecessary to manipulate all of the detail-page specific fields each time the primary visual rate or other key values change in the main page. These fields can be updated only when the detail page is triggered.



For more information, see Rate Detail Page and Rate Detail Secondary Page.

Syntax

```
InitializeDetailPanel (mode)
```

Location

EXCH_RT_WRK.RATE_DIRECT FieldChange

Input Parameters

mode

The input mode in which the detail page will be displayed.
The following table shows the possible values.

Value	Descr	Comment
I	Inquiry	Use this mode when using the inquiry version of the page. This will cause none of the fields to ever be disabled. (They don't need to be disabled because they are set to display-only in the page definition.)
D	Display	Use this mode when using the update version of the page and you always want the rate fields to be grayed (that is, when the process is past the point where the rate can functionally be changed).
U	Update	Use this mode when using the update version of the page and you want the fields to be enabled when applicable. Note that the rate fields may still be disabled if From and To currencies are equal or if the Override Cross Rate flag is set to No for a triangulated currency pair.

Return Values

None.

Where Used

Called from application-specific ProcessRateDetail function.

UpdateRates

UpdateRates updates the application-specific RATE_MULT, RATE_DIV, and VISUAL_RATE fields with the RATE_MULT and RATE_DIV values from the Exchange Rate work record. It uses record and field names stored on the Exchange Rate work record to identify the record fields on the application-specific work record.

Syntax

```
UpdateRates ( )
```

Location

EXCH_RT_WRK.RATE_MULT FieldChange

Input Parameters

None.

Return Values

None.

Where Used

Called from FieldChange on the pushbutton that invokes the Exchange Rate Detail page, after the page is dismissed. It is also called from within GetExchangeRate.

Application-Specific Requirements for Currency Conversion

Each application that shows a visual rate on a page must have an application-specific work record to hold the visual rate and the PeopleCode associated with it; this can be an existing work record. The suggested name for the field is VISUAL_RATE. The work record should also have a field to store the original rate for purposes of tolerance checking.

The application will also typically provide an application-specific table to store RATE_MULT and RATE_DIV values that are stored on the database.

Application-specific PeopleCode needs to format work record fields and call the common functions in various circumstances, such as RowInit or FieldChange on the currency or visual rate.



For more information about common functions, see Common PeopleCode Library Functions.

Sample Code

The goal in developing this sample code was to illustrate visual rate manipulation and exchange rate detail processing. While a lot of *standard* currency processing was included (for example the use of rate types, effective dates, and hiding and showing fields when the transaction currency is the same as base), the sample code is not intended to provide a complete solution to the needs of currency processing in applications. The focus is on the manipulation of the visual rate and the detail page.

On the prototype page, there was an exchange rate on a *header* record at level 0 and a rate on the *detail* record at level 1. Both rates were able to use the same Exchange Rate work record, since they were at different scroll levels. All of the sample programs are taken from the level 0 header work record.

The sample code refers to two records in addition to the Exchange Rate work record: the TST_RATE, which is the application-specific database table, and TST_RATE_WRK, which is the application-specific work record.

The messages used in the sample code are from message set 13 in the Message Catalog; this is the PeopleSoft message set used for Market Rates processing.

RowInit on the Visual Rate

```

declare function GetQuoteMethod peoplecode EXCH_RT_WRK.RATE_DIRECT FieldChange;

declare function GetMaxVariance peoplecode EXCH_RT_WRK.RATE_DIRECT FieldChange;

declare function SetVisualRate peoplecode EXCH_RT_WRK.RATE_MULT FieldChange;

/* Format the application specific record and field names */

EXCH_RT_WRK.RECNAME_VISUAL_RT = RECORD.TST_RATE_WRK;

EXCH_RT_WRK.FIELDNAME_VISUAL = FIELD.VISUAL_RATE;

EXCH_RT_WRK.RECNAME_RATES = RECORD.TST_RATE;

EXCH_RT_WRK.FIELDNAME_RT_MULT = FIELD.RATE_MULT;

EXCH_RT_WRK.FIELDNAME_RT_DIV = FIELD.RATE_DIV;

/* If the From and To currencies are known, get quote method and max var */

if All(TST_RATE.FOREIGN_CURRENCY) and

    All(TST_RATE.CURRENCY_CD) then

    GetQuoteMethod(TST_RATE.FOREIGN_CURRENCY, TST_RATE.CURRENCY_CD);

    GetMaxVariance("");

end-if;

/* If rate mult and rate div have values, format the visual rate */

```

```

if All (TST_RATE.RATE_MULT) and
    All (TST_RATE.RATE_DIV) then
    VISUAL_RATE = SetVisualRate();
    /* Save the original rate for variance checking */
    RT_RATE = VISUAL_RATE;
end-if;

/* Check if the primary visual should be grayed */

if EXCH_RT_WRK.RATE_TRIANGULATE = "Y" and
    EXCH_RT_WRK.PRIMARY_VISUAL = "FT" and
    EXCH_RT_WRK.XRATE_OVERRIDE = "N" then
    Gray (VISUAL_RATE);
else
    UnGray (VISUAL_RATE);
end-if;

```

FieldEdit on the Visual Rate

```

declare function CheckRateVariance peoplecode EXCH_RT_WRK.RATE_MULT FieldChange;

/* If a rate was entered, call the function to check the rate variance */

if all (VISUAL_RATE) then

    &RATE_OK = CheckRateVariance (VISUAL_RATE, RT_RATE, "");

    if not (&RATE_OK) then

        if EXCH_RT_WRK.ERROR_TYPE = "ERR" then

            error (MsgGet (13, 7, "Exchange rate differs from previous value by more
than %1%.", EXCH_RT_WRK.MAX_VARIANCE));

        else

            warning (MsgGet (13, 7, "Exchange rate differs from previous value by
more than %1%.", EXCH_RT_WRK.MAX_VARIANCE));

        end-if;

    end-if;

end-if;

```

FieldChange on the Visual Rate

```

declare function ProcessVisualRateChange peoplecode EXCH_RT_WRK.RATE_MULT
FieldChange;

declare function GetExchangeRate peoplecode EXCH_RT_WRK.RATE_MULT FieldChange;

/* If a new rate was entered, call the function to process the visual rate
change */

if All(VISUAL_RATE) then

    ProcessVisualRateChange(VISUAL_RATE);

else

/* The rate was blanked out, so re-retrieve it */

    if All(TST_RATE.FOREIGN_CURRENCY) and

        All(TST_RATE.CURRENCY_CD) and

        All(TST_RATE.CUR_RT_TYPE) and

        All(TST_RATE.CUR_EFFDT) then

        GetExchangeRate("", TST_RATE.FOREIGN_CURRENCY, TST_RATE.CURRENCY_CD,
TST_RATE.CUR_RT_TYPE, TST_RATE.CUR_EFFDT);

        /* Save the market rate for tolerance checking */

        RT_RATE = VISUAL_RATE;

        if None(TST_RATE.RATE_MULT) then

            MessageBox(64, MsgGetText(13, 1, "Market Rate Processing"), 13, 6, "The
currency exchange rate is zero.");

            end-if;

        end-if;

    end-if;

end-if;

```

After the call to ProcessVisualRateChange, there may be additional application specific processing required, such as recalculating a base currency amount, because RATE_MULT and RATE_DIV may have been changed.

Function to Process the Detail Page

The prototype uses the ProcessRateDetail function to group all of the code that is required to initialize, launch, and wrap up the rate detail page processing. It is called from FieldChange on the pushbutton that opens the Rate Detail page.

This function calls InitializeDetailFunction and then executes the DoModal function to launch either the update version or the inquiry version of the detail page, depending on the currently

active component. When returning from the update version of the detail page, this function calls `UpdateRates`, which updates the application-specific `RATE_MULT`, `RATE_DIV` and `VISUAL_RATE` fields.

```

declare function InitializeDetailPanel peoplecode EXCH_RT_WRK.RATE_DIRECT
FieldChange;

declare function UpdateRates peoplecode EXCH_RT_WRK.RATE_MULT FieldChange;

function ProcessRateDetail

    evaluate %Component

    when COMPONENT.TST_RATE_PNL

        InitializeDetailPanel("U");

        DoModal(PAGE.EXCH_RT_DTL, MsgGetText(13, 2, "Exchange Rate Detail"), - 1,
- 1, 0);

        UpdateRates();

        Break;

    when COMPONENT.TST_RATE_INQ

        InitializeDetailPanel("I");

        DoModal(PAGE.EXCH_RT_DTL_INQ, MsgGetText(13, 2, "Exchange Rate Detail"), -
1, - 1, 0);

    end-evaluate;

end-function;

```

When the function returns from the page after calling `UpdateRates`, additional application-specific processing may be required. For example, the base currency amount may need to be recalculated because `RATE_MULT` and `RATE_DIV` may have been changed.

RowInit on the Foreign Currency Rate

```

/* If From and To currencies are the same, set rates to 1 and gray */

if FOREIGN_CURRENCY = CURRENCY_CD then

    Gray(CUR_RT_TYPE);

    Gray(TST_RATE_WRK.VISUAL_RATE);

    RATE_MULT = 1;

    RATE_DIV = 1;

    TST_RATE_WRK.VISUAL_RATE = 1;

end-if;

```


FieldChange on the Foreign Currency

```

declare function GetQuoteMethod peoplecode EXCH_RT_WRK.RATE_DIRECT FieldChange;

declare function GetMaxVariance peoplecode EXCH_RT_WRK.RATE_DIRECT FieldChange;

declare function GetExchangeRate peoplecode EXCH_RT_WRK.RATE_MULT FieldChange;

if FOREIGN_CURRENCY = CURRENCY_CD then

    GetQuoteMethod(FOREIGN_CURRENCY, CURRENCY_CD);

    SetDefault(CUR_RT_TYPE);

    RATE_MULT = 1;

    RATE_DIV = 1;

    TST_RATE_WRK.VISUAL_RATE = 1;

    Gray(CUR_RT_TYPE);

    Gray(TST_RATE_WRK.VISUAL_RATE);

else

    SetDefault(RATE_MULT);

    SetDefault(RATE_DIV);

    SetDefault(TST_RATE_WRK.VISUAL_RATE);

    SetDefault(TST_RATE_WRK.RT_RATE);

    UnGray(CUR_RT_TYPE);

    UnGray(TST_RATE_WRK.VISUAL_RATE);

    if All(FOREIGN_CURRENCY) and

        All(CURRENCY_CD) then

        GetQuoteMethod(FOREIGN_CURRENCY, CURRENCY_CD);

        GetMaxVariance("");

        if All(CUR_RT_TYPE) and

            All(CUR_EFFDT) then

            GetExchangeRate("", FOREIGN_CURRENCY, CURRENCY_CD, CUR_RT_TYPE,
CUR_EFFDT);

            /* Save the market rate for variance checking */

            TST_RATE_WRK.RT_RATE = TST_RATE_WRK.VISUAL_RATE;

            if None(RATE_MULT) then

```

```

        MessageBox(64, MsgGetText(13, 1, "Market Rate Processing"), 13, 6,
"The currency exchange rate is zero.");

        end-if;

    end-if;

    if EXCH_RT_WRK.RATE_TRIANGULATE = "Y" and

        EXCH_RT_WRK.PRIMARY_VISUAL = "FT" and

        EXCH_RT_WRK.XRATE_OVERRIDE = "N" then

        Gray(TST_RATE_WRK.VISUAL_RATE);

        end-if;

    end-if;

end-if;

```

FieldChange on the Rate Type or Effective Date

```

declare function GetExchangeRate peoplecode EXCH_RT_WRK.RATE_MULT FieldChange;

SetDefault(RATE_MULT);

SetDefault(RATE_DIV);

SetDefault(TST_RATE_WRK.VISUAL_RATE);

SetDefault(TST_RATE_WRK.RT_RATE);

if All(FOREIGN_CURRENCY) and

    All(CURRENCY_CD) and

    All(CUR_RT_TYPE) and

    All(CUR_EFFDT) then

    GetExchangeRate("", FOREIGN_CURRENCY, CURRENCY_CD, CUR_RT_TYPE, CUR_EFFDT);

    /* Save the market rate for variance checking */

    TST_RATE_WRK.RT_RATE = TST_RATE_WRK.VISUAL_RATE;

    if None(RATE_MULT) then

        MessageBox(64, MsgGetText(13, 1, "Market Rate Processing"), 13, 6, "The
currency exchange rate is zero.");

        end-if;

    end-if;

end-if;

```

Formulas

The following formulas are used in both the common functions, and in application-specific code, to process market rates.

Formula 1: Determining Primary Visual Rate

```

If the currency pair is triangulated then

    If the primary visual is From-Cur → Ref-Cur then

         $PV = RD$ 

    If the primary visual is Ref-Cur → To-Cur then

         $PV = RM$ 

    If the primary visual is From-Cur → To-Cur then

        If quote basis is Direct then

             $PV = (RM / RD) * U$ 

        If quote basis is Indirect then

             $PV = (RD / RM) * U$ 

If the currency pair is not triangulated then

    If quote basis is Direct then

         $PV = (RM / RD) * U$ 

    If quote basis is Indirect then

         $PV = (RD / RM) * U$ 

```

Where

```

PV = Primary Visual Rate for the From-Currency to To-Currency

RM = RATE_MULT for the From-Currency to To-Currency

RD = RATE_DIV for the From-Currency to To-Currency

U = Units for the From-Currency to To-Currency

```

Formula 2: Tolerance Check on Visual Rate Override

```

If the Error Processing type is "None" then

    Return True

```

```

Else if OLD = 0 then

Return True

Else if (Abs (NEW - OLD) / OLD ) * 100 > MAX then

Return False

Else

Return True

```

Where

OLD = Prior value of the exchange rate*

NEW = New value of the exchange rate

When a rate is retrieved from the rate table, OLD will be the initial saved value. Whenever a new value is entered that does not generate an exception, the new value will be saved. This is to account for the fact that we are not always retrieving a rate from the rate table to start with, either because a rate type was not entered or a rate was not found. However, if the newly entered rate generated an exception, warning or error, it is not saved as the OLD rate. Finally, in change mode, the value from the database (converted to a visual rate) is the initial value of the old rate.

Formula 3: Determining RATE_MULT and RATE_DIV After Visual Rate Entry or Override

If quotation method is direct then

RM = VR

RD = U

If quote method is indirect then

RM = U

RD = VR

Where

RM = RATE_MULT for the From-Currency to To-Currency

RD = RATE_DIV for the From-Currency to To-Currency

VR = Visual Rate for the From-Currency to To-Currency

U = Units for the From-Currency to To-Currency

Formula 4: Determining Quote Method Description for Pop-up Page

If the currency pair is triangulated then

$RD \text{ FROM_CUR} = U \text{ REF_CUR} = RM \text{ TO_CUR}$

If not triangulated, then

If quotation method is direct then

$U \text{ FROM-CUR} = [(RM / RD) * U] \text{ TO-CUR}$

If quotation method is indirect then

$[(RD / RM) * U] \text{ FROM-CUR} = U \text{ TO-CUR}$

Where:

$RM = \text{RATE_MULT}$ for the From-Currency to To-Currency

$RD = \text{RATE_DIV}$ for the From-Currency to To-Currency

$U = \text{Units}$ for the From-Currency to To-Currency

$\text{From-Cur} = \text{From-Currency Code}$

$\text{To-Cur} = \text{To-Currency Code}$

$\text{Ref-Cur} = \text{Reference-Currency Code}$

Formula 5: Determining Historic Quote Method Description for Pop-up Page

If $RD = 1$ and (Quotation Method not Direct or $U \neq 1$ or Triangulate = Y) then

$1 \text{ FROM-CUR} = RM \text{ TO-CUR}$

Else if

$RM = 1$ and (Quotation Method not Indirect, or $U \neq 1$, or Triangulate = Y)

Then

$RD \text{ FROM-CUR} = 1 \text{ TO-CUR}$

Else

Not Applicable

Where:

$RM = \text{RATE_MULT}$ for the From-Currency to To-Currency

$RD = \text{RATE_DIV}$ for the From-Currency to To-Currency

U = Units for the From-Currency to To-Currency

From-Cur = From-Currency Code

To-Cur = To-Currency Code

Formula 6: Determining Visual Rates for Pop-up Page

Line 1 (From-Currency to To-Currency):

If quote basis is Direct:

$$VR1 = (RM / RD) * U$$

If quote basis is Indirect:

$$VR1 = (RD / RM) * U$$

Line 2 (From-Currency to Reference-Currency, only for triangulation):

$$VR2 = RD$$

Line 3 (Reference-Currency to To-Currency, only for triangulation):

$$VR3 = RM$$

Where:

VR1= Visual Rate for the From-Currency to To-Currency

VR2= Visual Rate for the From-Currency to Ref-Currency

VR3= Visual Rate for the Ref-Currency to To-Currency

RM = RATE_MULT for the From-Currency to To-Currency

RD = RATE_DIV for the From-Currency to To-Currency

U = Units for the From-Currency to To-Currency

Formula 7: Calculation of Triangulation Rates

$$RM = (RM2 / RD2) * U$$

$$RD = (RD1 / RM1) * U$$

Where:

RM = RATE_MULT for the From-Currency to To-Currency

RD = RATE_DIV for the From-Currency to To-Currency

RM1 = RATE_MULT for the From-Currency to Reference-Currency

RD1 = RATE_DIV for the From-Currency to Reference-Currency

RM2 = RATE_MULT for the Reference-Currency to To-Currency

RD2 = RATE_DIV for the Reference-Currency to To-Currency

U = Units for the From-Currency to To-Currency

The goal of this formula is to provide a resulting RATE_MULT and RATE_DIV for the triangulated pair that independently retain the fundamental elements of the two exchange rates that contributed to the triangulation.

The following two examples show how this will work in triangulation against the euro and how it might work in other triangulation scenarios.

Example 1

This example shows clearly how this Formula works for triangulation through the euro, where we know that all currencies are quoted directly against the euro. The resultant RATE_MULT and RATE_DIV for the USD to FRF pair still retain the precise values of the rates that contributed to the triangulation. As a result, the calculation to convert USD to FRF will be an exact replica of a *true* two-step conversion (that is: divide by the USD-to-EUR rate and multiply by the EUR-to-FRF rate).

<i>From Curr.</i>	<i>To Curr.</i>	<i>Quote Meth.</i>	<i>Quote Units</i>	<i>Triang ?</i>	<i>Ref. Curr.</i>	<i>Primary Visual</i>	<i>Cross Rate</i>	<i>Rate Mult</i>	<i>Rate Div</i>
USD	EUR	I	1	N	--	--	1.25	1	1.25
EUR	FRF	D	1	N	--	--	6.8	6.8	1
USD	FRF	D	1	Y	EUR	5.44	5.44	6.8	1.25

Example 2

This example shows a hypothetical case where the previous assumptions about the quote methods/units of the contributing currencies do not hold true. While this will never occur with triangulation through the euro, it may occur in other triangulation scenarios. In this case, even though the value in RATE_DIV for USD to FRF is not exactly the same as the visual rate for USD to EUR, it is the equivalent of the visual rate, stated with a quotation method of indirect with quote units of 1 (that is, 1.25 USD = 1 EUR is equivalent to 10 USD = 8 EUR).

<i>From Curr.</i>	<i>To Curr.</i>	<i>Quote Meth.</i>	<i>Quote Units</i>	<i>Triang ?</i>	<i>Ref. Curr.</i>	<i>Primary Visual</i>	<i>Visual Rate</i>	<i>Rate Mult</i>	<i>Rate Div</i>
USD	EUR	D	10	N	--	--	8.0	8.0	10
EUR	FRF	D	1	N	--	--	6.8	6.8	1
USD	FRF	D	1	Y	EUR	FR	--	6.8	1.25

Formula 8: Calculation of Cross Rates

If quotation method is direct then

$$RM = [(RM1 * RD2) / (RD1 * RM2)] * U$$

$$RD = U$$

If quotation method is indirect then

$$RM = U$$

$$RD = [(RD1 * RM2) / (RM1 * RD2)] * U$$

Where:

RM = RATE_MULT for the From-Currency to To-Currency

RD = RATE_DIV for the From-Currency to To-Currency

RM1 = RATE_MULT for the From-Currency to Reference-Currency

RD1 = RATE_DIV for the From-Currency to Reference-Currency

RM2 = RATE_MULT for the To-Currency to the Reference-Currency

RD2 = RATE_DIV for the To-Currency to the Reference-Currency

U = Units for the From-Currency to To-Currency

If you compare this formula to Formula 7: Calculation of Triangulation Rates, you will observe that:

- To calculate the cross rate, we use (in RM2) the conversion factor for the To-Currency to the Reference-Currency, whereas in the triangulation formula, we use the conversion factor for the Reference-Currency to the To-Currency. In the example, to calculate the cross rate of USD -> FRF, we look at USD→EUR and FRF→EUR, whereas in triangulation, we look at USD→EUR and EUR→FRF.
- In calculating the cross rate, we are actually calculating a new, standalone rate for USD to FRF (which is 5.44 in this example), whereas in triangulation, we retain the elements of the contributing rates (which are 6.8 and 1.25 in the USD-to-FRF conversion factor).

The following example illustrates how this works:

From Curr.	To Curr.	Quote Meth.	Quote Units	Triang ?	Ref. Curr.	Primary Visual	Cross Rate	Rate Mult	Rate Div
USD	EUR	I	1	N	--	--	1.25	1	1.25
FRF	EUR	I	1	N	--	--	6.8	1	6.8
USD	FRF	D	1	N	--	--	5.44	5.44	1

Formula 9: Calculation of Reciprocal Rates

If resulting quote method exists then

RM = RD1

RD = RM1

If resulting quote method does not exist then

RM = 1 / (RM1 / RD1)

RD = 1

Where:

RM = RATE_MULT for the From-Currency to To-Currency

RD = RATE_DIV for the From-Currency to To-Currency

RM1 = RATE_MULT for the To-Currency to From-Currency

RD1 = RATE_DIV for the To-Currency to From-Currency

When the currency pair being reciprocated has a defined quotation method, we can simply reverse RATE_MULT and RATE_DIV, keeping the same visual rate. However, if the currency pair does not have a quotation method defined, then we have to assume direct quotation going in both directions. Thus, we need to calculate a new visual rate that is the reciprocal of the original visual rate.

Example 1

From Curr.	To Curr.	Quote Meth.	Quote Units	Triang?	Ref. Curr.	Primary Visual	X-Rate Override	Visual Rate	Rate Mult	Rate Div
USD	EUR	I	1	N	--	--	--	1.25	1	1.25
EUR	USD	D	1	N	--	--	--	1.25	1.25	1

Example 2

From Curr.	To Curr.	Quote Meth.	Quote Units	Triang?	Ref. Curr.	Primary Visual	X-Rate Override	Visual Rate	Rate Mult	Rate Div
USD	CAD	n/a	n/a	n/a	--	--	--	1.25	1.25	1
CAD	USD	n/a	n/a	n/a	--	--	--	0.8	0.8	1

CHAPTER 13

Character Sets and Language Input/Output

This chapter describes the function of character sets on PeopleSoft client workstations, application servers, and database servers. This chapter provides an outline of the decisions that need to be made when you are installing and configuring a multilingual PeopleSoft system.

Understanding Character Sets

A *character set* (also known as a *codepage*) is an ordered set of characters in which each character is mapped to a numeric index, called a *codepoint*, that is used to store character data in a computer system. Many hundreds of character sets exist. Some are international standards, sanctioned by the International Organization for Standardization (ISO), some are country-specific national standards, and others are not standardized at all but are specific to a particular computer system vendor. Given the number of separate computers that are involved in a typical PeopleSoft installation, it is likely that your system will use several different character sets.

Unfortunately, although there is general agreement on the content and arrangement of most character sets, especially those standardized by ISO, a multitude of names are used by different vendors and software packages for similar or identical character sets. ASCII is the most common character set used worldwide, and it encodes the basic characters and symbols that are needed to write the English language. However ASCII is limited to 127 characters and cannot represent the characters needed by Western European languages such as French and German, let alone complex scripts, such as Japanese. Many character sets, however, use ASCII as their base and include all ASCII characters in addition to their other characters.

The table below illustrates just a few of the common character sets that you are likely to encounter and some of the names that are used by different vendors to refer to them:

Character Set	Description and Comments	Type	PeopleSoft and SQR Name	Oracle DBMS Name	Microsoft Windows Name
ISO 8859-1	Western European "Latin-1." ASCII-based. Contains all characters required to represent Western European languages. Does not include the euro	ISO	LATIN1 or ISO_8859-1	WE8ISO8859 P1	CP28591

	symbol.				
Microsoft Codepage 1252	Microsoft Codepage 1252 - Western European. Very similar to ISO 8859-1, except for the inclusion of line drawing and other characters in place of some symbols. Includes the euro symbol.	Vendor (Microsoft)	CP1252	WE8MSWIN 1252	CP1252
ISO 8859-2	Central/Eastern European "Latin-2." ASCII-based. Contains characters that are required for Central European languages, including Czech, Hungarian, and Polish. Does not include the euro symbol.	ISO	LATIN2 or ISO_8859-2	EE8ISO8859P 2	CP28592
ISO 8859-15	Western European extended "Latin-9." ASCII-based and similar to ISO 8859-1, but contains the euro symbol and several characters that are required for Icelandic.	ISO	LATIN9 or ISO_8859-15	WE8ISO8859 P15	CP28605
Shift-JIS	Japanese PC-based, double-byte character set. ASCII-based.	National (Japan)	SJIS	JA16SJIS	CP932
IBM CCSID 37	IBM Coded Character Set ID 37. Western European Multilingual EBCDIC-based character set.	Vendor (IBM)	EBCDIC	WE8EBCDIC 37	CP1140

Some of these character sets, such as ISO 8859-1 and IBM CCSID 37 are single-byte character sets (SBCS), meaning that they require only 1 byte to represent each character. For example, in ISO 8859-1, the hexadecimal number 61 represents the lowercase Latin letter *a*. However, for

complex languages such as Japanese, more characters exist in the language than the 256 values that can be represented by a single byte. Therefore, 2 bytes need to be used for each character. These character sets, such as Shift-JIS, are known as double-byte character sets (DBCS).

As you will need to make several decisions about character sets when you install and administer your PeopleSoft system, it is important that you understand the basics of the character sets used by PeopleSoft, and their properties, in order to make good configuration decisions.

The Unicode Standard

The most important consideration for dealing with character sets across your system is ensuring that all characters that you plan on representing with in your PeopleSoft system exist in the character set used by each component of the system. For example, if you are planning on maintaining Japanese characters in your employee names, you need to ensure that the character set used by your database system includes Japanese characters, that each external system feeding in to or out of PeopleSoft expects data in a character set that includes Japanese characters, and that your workstations and printers are installed with fonts that include those characters.

For example, the Japanese Shift-JIS character set contains all ASCII characters; therefore, it is sufficient for encoding both English data and the primary characters required in Japanese. However, it does not include the accented Latin characters that are needed for French and German, so it is not a suitable character set for implementations that encompass Western European countries.

Given the sample list of character sets in the table above and the number of languages required by a typical global PeopleSoft implementation, the work of selecting a character set can be a daunting task, especially when you are dealing with a large list of languages that you plan to support.

To simplify this type of situation, a universal character set was devised by an industry consortium of vendors in the late 1980s. The vision of the consortium was to create a single, universal character set that contains unique codepoints for all the characters used by the world's languages and for this new character set to be widely used across systems, in place of national and vendor standards.

This vision has been realized in the creation of The Unicode Standard, an internationally recognized character set that encodes every character that is required to write virtually every written language. The Unicode Standard, whose character repertoire is maintained jointly by The Unicode Consortium and ISO, is synchronized with ISO Standard 10646: Universal Multiple-Octet Coded Character Set (UCS).

PeopleSoft uses Unicode throughout its PeopleTools products, to simplify language handling. PeopleSoft allows the use of Unicode within PeopleSoft databases to enable you to maintain a single database with characters from virtually any language.

Unicode Encodings

The Unicode Standard and ISO 10646 provide for more than 1 million character encodings. Of course, only a tiny percentage of this encoding space is actually used today. The latest version (3.0) of The Unicode Standard encodes approximately 47,000 characters.

To manage such a large repertoire of characters, the characters are arranged in 16 *planes*. Each plane consists of 65,533 character positions. As of The Unicode Standard 3.0, only plane 0, known as the Basic Multilingual Plane (BMP), actually contains encoded characters. The other planes are planned for use in the near future, but they currently contain no characters.

Based on the concept of 16 planes, each Unicode character has a unique 4-byte code. As you can imagine, using 4 bytes of storage per character seems very wasteful, especially considering that most applications today use a small collection of characters and that, of the 16 planes defined, only 1 plane contains characters. Therefore, several different ways of encoding Unicode characters have been standardized and adopted by vendors who are implementing the standard.

Four encodings of the Unicode standard are widely used. All of them are fully compatible with each other and share the same repertoire of characters. They differ, however, in how each character is represented at the byte level. These encodings are:

- UTF-32 (formally known as UCS-4)
- UCS-2
- UTF-16
- UTF-8

Of these, only UCS-2 and UTF-8 are supported by PeopleSoft; however, following is a brief overview of each:

Encoding	Description
UTF-32 (Universal Transformation Form - 32-bit)	UTF-32 is the full 32-bit (4-byte) encoding of Unicode. Each Unicode character is represented by a 4-byte number. For example, the character LATIN SMALL LETTER A (a) is represented in hexadecimal as 0x00000061. UTF-32 was formally called UCS-4. PeopleSoft systems do not use UTF-32.
UCS-2 (Universal Character Set - 2-byte)	UCS-2 uses 2 bytes (16 bits) to represent each Unicode character. As such, it can reference only 65,535 codepoints and is limited to characters on the Basic Multilingual Plane (BMP). Because all characters currently encoded in the Unicode Standard, as of September 2000, are on plane 0, UCS-2 is sufficient to reference all Unicode characters. PeopleSoft uses UCS-2 in memory on the Windows client, the application server, and in Unicode Microsoft SQL Server databases.

UTF-16 (Universal Transformation Form - 16-bit)	<p>UTF-16 is an extension of UCS-2 that allows the application to reference characters on planes other than the BMP by combining two UCS-2 code units to form a single character. UCS-2 is upward-compatible with UTF-16 in that each UCS-2 character is also a valid character in UTF-16. However, UTF-16 allows characters outside the BMP to be referenced. These additional characters are known as surrogate characters because they require two UTF-16 code units to be represented: a low surrogate and a high surrogate. When no surrogate characters are present, UTF-16 is identical to UCS-2.</p> <p>PeopleSoft plans to use UTF-16 in future versions of PeopleTools, when surrogate characters are encoded and need to be supported, but does not use UTF-16 in PeopleTools 8.1.</p>
UTF-8 (Universal Transformation Form - 8-bit)	<p>UTF-8 is a transformation of Unicode that encodes each character as 1 to 4 bytes, depending on which character is being encoded. All ASCII characters are encoded in UTF-8 as 1 byte, and this byte is identical to the encoding in ASCII. UTF-8 can therefore be seen as backward-compatible with ASCII data. All characters in the BMP are encoded as 1, 2, or 3 UTF-8 bytes. Characters in other planes (none of which are encoded today) are encoded as 4 bytes in UTF-8. UTF-8 has three main advantages: it is fully ASCII compatible, ASCII data can be considered as UTF-8 data, and it does not require that all characters use 2 or more bytes of storage.</p> <p>PeopleTools uses UTF-8 for serving HTML pages in PeopleSoft Internet Architecture (PIA). PeopleSoft also uses UTF-8 in Oracle Unicode databases.</p>

The table below summarizes these four Unicode encodings and their use in PeopleSoft:

Unicode Encoding	Minimum Bytes/Character	Maximum Bytes/Character	PeopleSoft Usage	Example Encodings
UTF-32	4	4	None	LATIN SMALL LETTER A (a) 0x00000061 LATIN SMALL LETTER N WITH TILDE (ñ) 0x000000f1 HIRAGANA LETTER KA (か) 0x0000304b
UTF-16	2	4	None; planned for	LATIN SMALL LETTER A (a) 0x0061

Unicode Encoding	Minimum Bytes/ Character	Maximum Bytes/ Character	PeopleSoft Usage	Example Encodings
			future	LATIN SMALL LETTER N WITH TILDE (ñ) 0x00f1 HIRAGANA LETTER KA (か) 0x304b
UTF-8	1	4	PIA HTML pages; inbound and outbound XML; Oracle databases.	LATIN SMALL LETTER A (a) 0x61 LATIN SMALL LETTER N WITH TILDE (ñ) 0xc3b1 HIRAGANA LETTER KA (か) 0xe3818b
UCS-2	2	2	In-memory Windows Client; Application Server	LATIN SMALL LETTER A (a) 0x0061 LATIN SMALL LETTER N WITH TILDE (ñ) 0x00f1 HIRAGANA LETTER KA (か) 0x304b

Non-Unicode Character Sets

Although much of the PeopleSoft system runs using Unicode, there are several components that you can choose to configure with a non-Unicode character set. When making these choices, you should understand the types of character sets other than Unicode that exist.



For the sake of terminology, some systems, such as Microsoft Windows refer to two types of character sets: Unicode and ANSI. ANSI, in this context, refers to the American National Standards Institute, which maintains equivalent standards for many national and international standard character sets. In the context of this book, ANSI character sets refer to non-Unicode character sets, which can be any international, national, or vendor standard character set, such as those discussed at the beginning of this chapter.

Single-Byte Character Sets

Most character sets use 1 byte to represent each character and are therefore known as single-byte character sets, or SBCS. These character sets are relatively simple and can represent up to 255 unique characters. Examples of single-byte character sets are ISO 8859-1, ISO 8859-2, Microsoft CP1252, and IBM CCSID 37.

Double-Byte Character Sets

Double-byte character sets (DBCS) use 1 or 2 bytes to represent each character and are typically used for writing complex scripts such as Japanese, Chinese, and Korean. Most double-byte character sets allow a mix of 1-byte and 2-byte characters, so you cannot assume an even-string byte length.

PeopleSoft supports the use of two distinct types of DBCS: shifting and non-shifting. The difference between these types of DBCS are in the way that the system determines whether a particular byte represents 1 character or is part of a 2-byte character.

Non-Shifting DBCS

Non-shifting double-byte character sets use ranges of code points, specified by the character set definition, to determine whether a particular byte represents 1 character or is part of a 2-byte character.

In non-shifting DBCS, the 2 bytes used to form a character are called *lead bytes* and *trail bytes*. The lead byte is the first in a 2-byte character, and the trail byte is the last in the 2-byte sequence. Non-shifting DBCS differentiate single-byte characters from double-byte characters by the numerical value of the lead byte. For example, in the Japanese Shift-JIS encoding, if a byte is in the range 0x81-0x9F or 0xE0-0xFC, then it is a lead byte and must be paired with the following byte to form a complete character.

The most popular client-side Japanese codepage, Shift-JIS, uses this lead byte/trail byte encoding scheme, as do most Microsoft Windows and UNIX ASCII-based, double-byte character sets representing Chinese, Japanese, and Korean characters.

Shifting DBCS

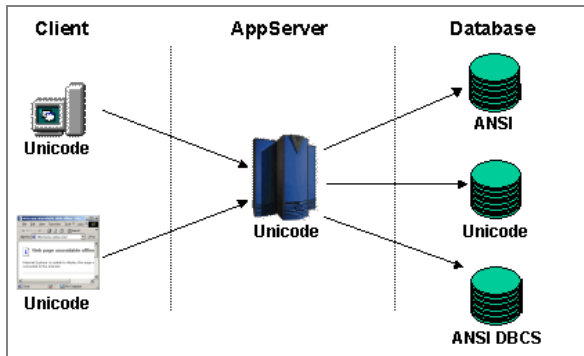
There is another double-byte encoding scheme in use that doesn't make use of the lead byte/trail byte concept. This system is known as a *shifting* character set. The IBM DB2/390 EBCDIC-based Japanese, Chinese, and Korean character sets use this shifting encoding scheme.

Instead of reserving a range of bytes as lead bytes, shifting double-byte character sets always keep track of whether they are in double-byte or single-byte mode. In double-byte mode, every 2 bytes form a character. In single-byte mode, every byte is a character in itself. To track what mode the character set is in, the system uses *shifting* characters. By default, the character set is expecting single-byte data. As soon as a double-byte character needs to be represented, a shift-in byte is added to the string. From this point on, all characters are expected to be 2 bytes per character. This continues until a shift-out byte is detected, which indicates that the character set should go back to single-byte per character mode.

This scheme, while more complex than the lead byte/trail byte scheme, provides greater performance, as the system always knows how many bytes should be in any particular character. Unfortunately, it also increases the length of the string. For example, a character string that is comprised of a mixture of single-byte and double-byte characters will require more space to store in a shifting character set because you need to include the shift-in and shift-out bytes, as well as the data itself.

Character Sets in the PeopleSoft Architecture

PeopleSoft installations include multiple components, each of which must handle differing character sets.



PeopleSoft Unicode Architecture

PeopleSoft clients and application servers use Unicode exclusively and do not rely on other character sets for the representation and processing of data. However, depending on your environment, not all components of your system may support Unicode encoded data, so you may not be able to run all parts of your system in Unicode. Therefore, PeopleTools enables you to configure several components of the system to use other character sets. The following PeopleTools system components or functions enable you to specify a non-Unicode character set:

- PeopleSoft database.

Not all database platforms support Unicode data storage. Even when Unicode storage is available, not all PeopleSoft implementations require the flexibility in language choices that are available with Unicode, so you may choose to use a non-Unicode encoding for your database.

- COBOL.

The character set used for PeopleSoft COBOL processing must match the character set of your database. If you created a Unicode database for PeopleSoft, you must run your COBOL in Unicode also.

- File input/output.

All file operations in PeopleTools, including file layout objects, trace and log files, and file operations from SQR can be performed in Unicode or any supported non-Unicode character set. This is useful in situations where you need to interface with an external system that does not support Unicode.

- Non-Unicode-compliant, third-party products.

Some third-party products that are supported by PeopleTools do not yet support the Unicode Standard. In this case, PeopleTools converts application data to a specific non-Unicode character set before communicating with these tools. Some of the products that do not yet support Unicode data are Crystal Reports, Hyperion Essbase, and Cognos PowerPlay.

When Unicode is not used for any of these types of operations or data storage, PeopleSoft transparently handles the conversion from Unicode to a non-Unicode character set. Which non-Unicode character set is used depends on several settings, which are discussed in detail later in this chapter.

The table below lists the character sets that are supported by PeopleSoft and the names by which they can be referred to in PeopleSoft applications. There are several places where you can use these character set names:

- In PeopleCode programs, for manipulating file layout objects.
- In SQR code, for controlling the character set of input, output, and report files.
- In PSSQR.INI to determine the default character set for files manipulated by SQR.
- In the UNIX application server configuration, to determine the default non-Unicode character set for log files, trace files, and UNIX operating system interfaces.
- In your database. A limited number of the character sets in this table are supported as database character sets. See the *PeopleSoft Hardware and Software Requirements* guide for details about the character sets that are supported for your database platform.

Character Set Name	Description and Comments	Character Set Type
ANSI	Current ANSI Code Page Not really a character set, but causes the system to use the default non-Unicode character set of the host operating system.	SBCS
ASCII	US 7-bit ASCII	SBCS
Big5	Big5 (Traditional Chinese)	Non-Shifting DBCS
CCSID1027	IBM EBCDIC 1027 (Japanese-Latin)	SBCS
CCSID1047	IBM EBCDIC 1047 (Latin1)	SBCS
CCSID290	IBM EBCDIC 290 (Katakana)	SBCS
CCSID300	IBM EBCDIC 300 (Kanji)	Non-Shifting DBCS
CCSID930	IBM EBCDIC 930 (Kana-Kanji)	Shifting DBCS
CCSID935	IBM EBCDIC 935 (Simplified Chinese)	Shifting DBCS
CCSID937	IBM EBCDIC 937 (Traditional Chinese)	Shifting DBCS
CCSID939	IBM EBCDIC 939 (Latin-Kanji)	Shifting DBCS
CCSID942	IBM EBCDIC 942 (Japanese PC)	Non-Shifting DBCS
CP1026	Windows 1026 (EBCDIC)	SBCS
CP1250	Windows 1250 (Eastern Europe)	SBCS
CP1251	Windows 1251 (Cyrillic)	SBCS

Character Set Name	Description and Comments	Character Set Type
CP1252	Windows 1252 (Latin1)	SBCS
CP1253	Windows 1253 (Greek)	SBCS
CP1254	Windows 1254 (Turkish)	SBCS
CP1255	Windows 1255 (Hebrew)	SBCS
CP1256	Windows 1256 (Arabic)	SBCS
CP1257	Windows 1257 (Baltic)	SBCS
CP1258	Windows 1258 (Vietnamese)	SBCS
CP1361	Windows 1361 (Korean Johab)	SBCS
CP437	MS-DOS 437 (US)	SBCS
CP500	Windows 500 (EBCDIC 500V1)	SBCS
CP708	Windows 708 (Arabic - ASMO708)	SBCS
CP720	Windows 720 (Arabic - ASMO)	SBCS
CP737	Windows 737 (Greek - 437G)	SBCS
CP775	Windows 775 (Baltic)	SBCS
CP850	MS-DOS 850 (Latin1)	SBCS
CP852	MS-DOS 852 (Latin2)	SBCS
CP855	MS-DOS 855 (IBM Cyrillic)	SBCS
CP857	MS-DOS 857 (IBM Turkish)	SBCS
CP860	MS-DOS 860 (IBM Portuguese)	SBCS
CP861	MS-DOS 861 (Icelandic)	SBCS
CP862	MS-DOS 862 (Hebrew)	SBCS
CP863	MS-DOS 863 (Canadian French)	SBCS
CP864	MS-DOS 864 (Arabic)	SBCS
CP865	MS-DOS 864 (Nordic)	SBCS
CP866	MS-DOS 866 (Russian)	SBCS
CP869	MS-DOS 869 (Modern Greek)	SBCS
CP870	Windows 870	SBCS
CP874	Windows 864 (Thai)	SBCS
CP875	Windows 875 (EBCDIC)	SBCS
CP932	Windows 932 (Japanese)	Non-Shifting DBCS
CP936	Windows 936 (Simplified Chinese)	Non-Shifting DBCS

Character Set Name	Description and Comments	Character Set Type
CP949	Windows 949 (Korean)	Non-Shifting DBCS
CP950	Windows 950 (Traditional Chinese)	Non-Shifting DBCS
EBCDIC	IBM EBCDIC CCSID37 (USA)	SBCS
EUC-JP	Extended UNIX Code (Japanese)	Non-Shifting DBCS
EUC-KR	Extended UNIX Code (Korean)	Non-Shifting DBCS
EUC-TW	Extended UNIX Code (Taiwan)	Non-Shifting DBCS
EUC-TW-1986	Extended UNIX Code (TW-1986)	Non-Shifting DBCS
GB12345	GB 2312 (Simplified Chinese)	Non-Shifting DBCS
GB2312	GB 2312 (Simplified Chinese)	Non-Shifting DBCS
ISO-2022-JP	ISO-2022-JP Japanese	Shifting DBCS
ISO-2022-KR	ISO-2022-JP Korean	Shifting DBCS
ISO_8859-1	ISO 8859-1 (Latin1)	SBCS
ISO_8859-10	ISO 8859-10 (Latin6)	SBCS
ISO_8859-11	ISO 8859-11 (Thai)	SBCS
ISO_8859-14	ISO 8859-14 (Latin8)	SBCS
ISO_8859-15	ISO 8859-15 (Latin9 / Latin0)	SBCS
ISO_8859-2	ISO 8859-2 (Latin2)	SBCS
ISO_8859-3	ISO 8859-3 (Latin3)	SBCS
ISO_8859-4	ISO 8859-4 (Latin4)	SBCS
ISO_8859-5	ISO 8859-5 (Cyrillic)	SBCS
ISO_8859-6	ISO 8859-6 (Arabic)	SBCS
ISO_8859-7	ISO 8859-7 (Greek)	SBCS
ISO_8859-8	ISO 8859-8 (Hebrew)	SBCS
ISO_8859-9	ISO 8859-9 (Latin5)	SBCS
JIS_X0201	Japanese Half-width Katakana	Non-Shifting DBCS
JIS_X_0208	Japanese Kanji	Non-Shifting DBCS
Java	Java (Unicode encoding)	Unicode
Johab	Johab (Korean)	Non-Shifting DBCS
Shift_JIS	Shift-JIS (Japanese)	Non-Shifting DBCS
UCS2	Unicode UCS-2	Unicode
UTF7	Unicode UTF-7	Unicode

Character Set Name	Description and Comments	Character Set Type
	An outdated Unicode 7-bit clean transformation sometimes used for email that must pass through gateways that do not support 8-bit characters.	
UTF8	Unicode UTF-8	Unicode



Although the PeopleSoft use of Unicode incorporates all languages, PeopleSoft does not currently support Arabic or Hebrew; those languages require bi-directional rendering because they are written with a right-to-left general orientation.

Database Character Sets

The primary character set decision that needs to be made when installing PeopleSoft is the character set used for the database system. Ideally, all databases will be encoded in Unicode; however, as can be seen from the Unicode encoding chart earlier in this chapter, it is obvious that in many cases Unicode requires several bytes to represent each character when only 1 byte may be required in a non-Unicode character set. Therefore, PeopleSoft still allows you to use a non-Unicode character set for your database character set, if you don't require the mix of languages enabled by a Unicode database.

Using a Unicode encoded database, you can maintain a single database with data in any combination of languages you choose. A single PeopleSoft application server can serve multiple users connecting to the mixed-language database, regardless of the language or character set of those users' client machines. The only restriction on a user's ability to access mixed-language data is the capability of the user's client workstation to interpret, display, and accept keyboard entry of the characters from the various languages.

Most language- or region-specific, non-Unicode character sets provide sufficient characters for only one or a small handful of languages. If you create a non-Unicode database, you must ensure that all the characters for all the languages that you plan on using can be represented in the character set you choose. See your operating system and database guides or the links provided earlier in this chapter to view the contents of each character set and to determine whether the languages you require are represented.

Depending on the data you store and how your database stores Unicode characters, a Unicode database can be significantly larger than a non-Unicode database. Keep in mind, however, that only the storage of character data is impacted; the space required for non-character data, such as numbers and dates (which are stored by your database system as numbers), is not affected.

Depending on your database platform, you can use one of the four character set types (SBCS, Non-Shifting DBCS, Shifting DBCS, and Unicode) when creating your database. However, the number of characters you can store in each column will be affected greatly by the type of character set you choose for your database encoding.



For more information about choosing a database character set, see the *PeopleSoft Hardware and Software Requirements* guide and the *PeopleSoft Installation and Administration* book for your database platform.

Application Server Character Sets

The PeopleSoft application server allows the creation of files on the server through the use of PeopleCode file layout objects and for log and trace files. Although the PeopleSoft application server uses Unicode internally for all data processing, it can create these files in Unicode or in a non-Unicode character set.

Each PeopleSoft application server is configured with a default non-Unicode character set. If a file operation needs to create a non-Unicode file, this is the character set used, unless another character set is explicitly specified in the file operation. For example, if you create a file layout object to write a non-Unicode file, but you don't specify which character set the file should be created in, the default non-Unicode character set of the application server is used.

On Windows NT application servers, the operating system's default ANSI character set is used for the PeopleSoft application server's default non-Unicode character set. In Windows NT 4.0, this default ANSI character set is hard-wired into the operating system, based on the language of the Windows version being used, and cannot be changed. For example, in English Windows NT 4.0, the default non-Unicode character set is Microsoft CP1252. In Japanese Windows NT 4.0, the default non-Unicode character set is Microsoft CP932.

Windows 2000 allows you to change the default character set of the system, although, as installed, the default character set matches the default locale of the Windows 2000 machine. To change the system default locale (and therefore the character set), on Windows 2000 servers, use the Control Panel's Regional Options applet and click the Set Default button under *Language settings for the system*.

When running on UNIX, the PeopleSoft application server enables you to specify the default non-Unicode character set in the application server's configuration file, which is selected using the PSADMIN tool. Any valid PeopleSoft character set (as listed in the table earlier in this chapter) with a character set type of SBCS or Non-Shifting DBCS is a valid default non-Unicode character set for PeopleSoft application servers that run on UNIX.



For more information and a list of valid PeopleSoft character sets, see Character Sets in the PeopleSoft Architecture.

Client Workstation Character Sets and Language Management

The client components of PeopleTools need to be considered when you are planning your language strategy. The requirements for language support on your client workstations are

markedly different, depending on whether you are using PeopleSoft Internet Architecture or the PeopleTools Windows client and development tools.

Character Sets and Fonts in PeopleSoft Internet Architecture

PeopleSoft Internet Architecture serves all HTML pages using the UTF-8 encoding of Unicode. This encoding is recognized automatically by the Web browser, because the encoding of the page is announced in the HTTP header when the browser communicates with the Web server. Most modern browsers can support the use of UTF-8 encoded HTML pages.

However, other components are needed by your browser in order to correctly display and enter the vast array of characters that are available in Unicode. Specifically, you need appropriate fonts to display the various scripts in which you expect data to be maintained, and you may need alternative keyboard layouts or, in the case of complex scripts, such as Chinese, Japanese, and Korean, input managers, to enter data in some languages. The requirement for alternate keyboard and input method editors is the same for both PeopleSoft Internet Architecture and the Windows client, and this is discussed in a separate section, below.

Not all fonts contain a full repertoire of Unicode characters, because many fonts are tailored to address a specific list of languages and contain only the glyphs that are required by those languages. If you try to view Unicode data with a font that does not contain the appropriate characters for the language being displayed, you will most likely see square boxes in place of the appropriate characters. Your data has not been corrupted; there is just no glyph available in the current font for the character that the system is trying to display. For this reason, you may need to license or configure several fonts for a global PeopleSoft system.

PeopleSoft Internet Architecture includes a set of style sheets, defined with PeopleSoft Application Designer, that determine the font used for displaying HTML pages. In some cases, your application data may contain characters that are not present in this font and that require a different font. For example, none of the default fonts in the English version of Microsoft Windows NT 4.0 include sufficient Japanese characters to display most Japanese translated data.

You may need to obtain and configure fonts that contain the characters for the languages that you are planning to use, if your workstations are not already configured with these fonts. Fonts can be obtained from many sources, as follows:

- Many Windows and other operating system applications are packaged with Unicode fonts containing glyphs covering a large range of languages. Microsoft Office 2000 is packaged with several fonts containing a large portion of the characters in the Unicode standard, including the Microsoft Sans Serif font. These fonts can be used in PeopleSoft Internet Architecture by specifying them in the PeopleSoft Application Designer style sheet definitions or by following the browser-specific instructions below.
- Several public domain fonts exist that contain a large character repertoire and can be used in Web browsers. A good index for such fonts can be found on the Web at:
<http://www.hclrss.demon.co.uk/unicode/fonts.html>.
- Several font foundries license fonts for individual or corporate use. Some of these foundries include Agfa Monotype, Bitstream, and Tiro Typeworks.

Depending on your browser, you may also be able to download fonts from your browser's manufacturer.

Microsoft Internet Explorer 5.0 or later allows you to download fonts for most popular languages from the Microsoft Web site. You can trigger these downloads by visiting the Internet Explorer Web site at <http://www.microsoft.com/ie> and downloading the Internet Explorer Language pack in the language of your choice. Alternatively, select **View, Encoding** to select a language for which you want to download fonts. This will cause Internet Explorer 5.0 (or later) to automatically download the font pack from the Microsoft Web site. After this download is completed, you will need to select **View, Encoding** again in order to reset your encoding to UTF-8 so that PeopleSoft Internet Architecture pages display correctly.

Microsoft Internet Explorer automatically selects the appropriate font for displaying data in the language that is presented in the HTML page. The appropriate font is the one that most closely matches the font specified by PeopleSoft in the page's style sheet definition.

Netscape Navigator 6.0 also performs this automatic font matching, based on the language of the page, but does not provide international fonts.

Netscape Navigator 4.0 versions do not perform automatic font matching, and you will need to configure the browser manually if the font specified on the style sheet does not exist on particular workstations. You can do this by selecting **Edit, Preferences** and then selecting Fonts from the tree on the left side of the dialog box that appears. All PeopleSoft Internet Architecture pages are served using the UTF-8 Unicode encoding, so you will need to change the font setting for the Unicode character set in the Netscape Font dialog box.

Fonts and the Windows Client

The Windows client allows you to specify the font used for all graphical components in PeopleSoft Application Designer, the Windows client panel processor, and all other PeopleTools Windows modules. There are three separate areas that allow you to specify a font:

Configuration Manager Font Setting (Display Tab)

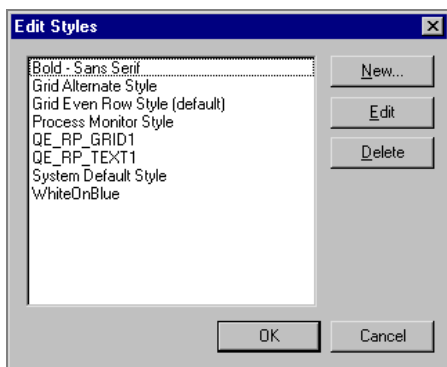
This setting affects the font that is used by all the designer components of PeopleTools, including all the text contained in the Windows resource files, as described in the chapter Translating Windows Resources.

Changing this font setting may be necessary if your workstation's default locale does not contain the characters that are used for the language you are attempting to display or maintain. For example, if you are attempting to view Japanese characters on an English Windows workstation, you can change the Configuration Manager Font setting to select a font that contains the characters for the language that you are trying to display.

Several fonts shipped with Microsoft Windows 2000 and Microsoft Office 2000, including Lucida Sans Unicode and Microsoft Sans Serif, contain a large number of glyphs, covering most of the languages supported by the Unicode Standard.

Styles

PeopleSoft Application Designer enables you to modify style settings by selecting **Tools, Miscellaneous Objects, Styles**. Styles affect the font that is used for displaying pages in the Windows client. They do not affect the formatting of pages served by PeopleSoft Internet Architecture.



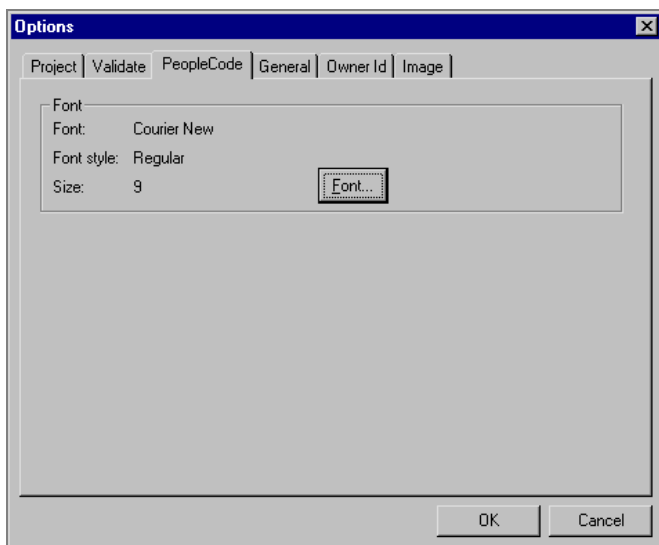
Edit Styles dialog box



For more information about style settings, see the Creating Style Sheet Definitions.

PeopleCode Font

The PeopleCode editor in PeopleSoft Application Designer also enables you to select a font for character display in the editor's window itself. This is useful if the PeopleCode programs you are working on contain Unicode characters. You select this font in Application Designer. Select **Tools, Options**, and then select the PeopleCode tab to set the font.



Font Options dialog box



Microsoft Windows 95, Windows 98, and Windows ME (Millennium Edition) do not support the native rendering of Unicode characters. Therefore, when using the Windows client on these platforms, you will be unable to view or enter characters that are outside the default non-Unicode character set of the workstation. For example, on a English Windows 98 workstation, you cannot view or enter Japanese data using the Windows client even if the appropriate fonts are installed. Of course, you can view or enter Japanese data using the Japanese edition of Windows 98.

PeopleSoft strongly recommends that when using the Windows client to connect to a multilingual Unicode database, you use a Windows NT or Windows 2000 workstation. Both fully support Unicode input and output. Microsoft Windows 95, Windows 98, and Windows ME workstations fully support the display and input Unicode characters when you use PeopleSoft Internet Architecture with a Web browser that supports Unicode, such as Microsoft Internet Explorer.

Input Methods

If users will be entering foreign language data using either PeopleSoft Internet Architecture or the Windows client, you need to ensure that an appropriate keyboard or input method editor is installed on the workstation.

Most alphabetic languages can be typed using a relatively simple keyboard layout. Several specialized keyboard layouts exist for most languages, and these keyboard layouts can be configured through your operating system. For example, a Spanish keyboard layout contains keys for the enyeh character (ñ) and several other accented characters.

There are several ways of entering these characters using a non-localized keyboard. Your operating system manual can help you use specialized keyboard layouts, such as the *English - International* layout, which allows the entry of accented characters using two keystrokes. For information about keyboards that are supported by Microsoft Windows, and instructions for installing and configuring Windows keyboard layouts, visit this Web site:
<http://www.microsoft.com/globaldev/keyboards/keyboards.asp>.

Other more complex languages, such as Chinese, Japanese, and Korean, require the use of a front-end processor to intercept multiple keyboard strokes and transform them into an ideographic character. These are known as *input method editors* (IMEs), and they must be installed on each workstation where you plan to enter the ideographic languages.

Most localized versions of operating systems for these languages come preconfigured with IMEs that are appropriate for the language supported by the operating system. But on systems where the default locale is not Chinese, Japanese, or Korean, you may need to configure or license an IME from a third-party vendor. PeopleSoft Internet Architecture supports any IME that is supported by your browser. The Windows client supports all standard Microsoft IMEs.

Data Sorting and Collation

An important part of any PeopleSoft system is the algorithm used to sort data that is displayed in reports, list boxes, and other outputs where you would expect to see a sorted list of values. This sorting is also known as *collation*. PeopleTools relies on several different components to provide the sorting of these lists, but it relies primarily on the database system's sort order as returned by the SQL ORDER BY clause.

As of PeopleTools 8.1, PeopleSoft requires that the database server use a binary sort, based on the order of the characters in the character set used by the database server. This limitation exists because some PeopleTools and application processes rely on an ordered list sorted in memory via SQL match. This problem may be addressed in a future release of PeopleSoft.

Therefore, an important consideration when you are choosing your database character set is the order in which you expect the values to be displayed to the end user. Most character sets are ordered in such a way that when they are sorted in binary, the resultant data is ordered appropriately for the languages concerned. However, when you are dealing with European accented characters, there are several different culturally accepted sorts, depending on the country and/or language of each user, so a single database sort may not meet all users' needs. Until PeopleSoft supports non-binary database sorting, you will need to choose a character set whose binary order matches the needs of your users. Alternatively, you may want to re-sort data output using a third-party tool, such as Microsoft Excel, that supports culturally sensitive sorting.

This is particularly important when you are dealing with Unicode databases, because the Unicode standard itself places very little relevance on the binary ordering of codepoints within the standard. Therefore, some single-language implementations, such as a Japanese-only implementation, may use a national standard character set, such as Shift-JIS, in place of Unicode in order to have report and list output sorted by the binary order of this standard instead of by the binary order of the codepoints in Unicode.

Field Length Validation

Application Designer Field Length Semantics

One of the more significant impacts that your database character set choice has on your PeopleSoft system is the way that PeopleTools interprets the length of the columns that you define in PeopleSoft Application Designer.

If you create a Unicode database, the field length, as shown in Application Designer, indicates the maximum number of Basic Multilingual Plane (BMP) characters that are permitted in the field, regardless of the Unicode encoding used by your database. To accomplish this, PeopleSoft sizes your database columns based on the worst-case ratio between bytes and characters in the Unicode encoding used by your database. For example, the UTF-8 character set is used by Oracle; therefore, the worst-case character-to-byte ratio, when running against an Oracle Unicode database, is 1:3.

Some database platforms, such as Oracle, use byte lengths to measure column sizes when operating in a Unicode database, while others use character lengths. The table below shows the

data types and length semantics for each of the valid PeopleSoft database/character set combinations.

If you create a non-Unicode database, the field length in PeopleSoft Application Designer represents the number of bytes permitted in the field, based on the character set you used to create your database. Therefore, a PeopleSoft Unicode database will allow you significantly more space for character data within the database when dealing with ideographic languages such as Japanese, Chinese, and Korean.

The table below shows some of the possible database encodings for database platforms that are supported in Unicode and/or DBCS by PeopleSoft, and their effects on database column sizes.

Database Platform	Database Character Set	Database Representation of a Character Field With Length 10 in Application Designer	Worst-case Number of Characters Allowed in a Character field with Length 10 in Application Designer
Oracle	Unicode (UTF-8)	VARCHAR2(30)	10
	Any SBCS	VARCHAR2(10)	10
	Non-Shifting DBCS	VARCHAR2(10)	5
Microsoft SQL Server	Unicode (UCS-2)	NCHAR(10)	10
	Any SBCS	CHAR(10)	10
	Non-Shifting DBCS	CHAR(10)	5
Sybase	Any SBCS	CHAR(10)	10
	Non-Shifting DBCS	CHAR(10)	5
IBM DB2 for OS/390	Any SBCS	CHAR(10)	10
	Shifting DBCS	CHAR(10)	4 (4 x 2 byte characters, plus shift-in & shift-out bytes)
All Others	Any SBCS	CHAR(10)	10

Field Length Checking for Non-Unicode Databases

As you can see from the table above, the number of characters allowed in a PeopleSoft field varies, depending on the character set of your database. Because all the components of PeopleTools run in Unicode, by default, field length checking occurs in terms of Unicode character counts. Therefore, special length checking has to occur each time you move off a field to ensure that the string you entered will fit in the database column when the string is converted to the database's character set.

For sizing page fields, PeopleTools uses the Unicode length of the field as defined in PeopleSoft Application Designer. For example, if a field is defined in Application Designer as a 10-character field, page fields in both PeopleSoft Internet Architecture and the PeopleTools Windows clients allow 10 characters to be entered.

This simple validation is sufficient if your database is created as a Unicode database or as a non-Unicode SBCS database.

If you are running a Unicode database, this is true because PeopleSoft Unicode databases measure database column lengths in terms of BMP characters, not bytes; so 10 Unicode characters entered through PeopleSoft Internet Architecture or through the Windows client will fit in a 10-character Unicode field. If you are running a non-Unicode SBCS database, the ratio of characters to bytes in the database's character set is 1:1, as a SBCS character set can use only 1 byte per character, and the simple character-based length check also succeeds.

However, if your database is encoded in a non-Unicode DBCS character set, special length validation will need to take place because your database column size is created relative to a byte count, not a character count as is used by the simple field length validation.

For example, if a user enters 10 Japanese characters into a field that is defined as CHAR(10) in Application Designer, this string needs 20 bytes of storage in a non-shifting DBCS character set and 22 bytes of storage in a shifting character set. This 10-character input would fail insertion in both of these databases.

To address this issue, the page processor checks the Data Field Length Checking option on the PeopleTools Options page and performs character-set specific length validation against the contents of each field when the field is validated. Typically length validation occurs when the field's FieldChange PeopleCode event fires, so the actual time of validation may differ, depending on whether your page uses deferred mode processing.

To enable or disable data field length checking:

1. Select PeopleTools, Utilities, Use, PeopleTools Options.

The PeopleTools Options page appears:

The screenshot shows the 'PeopleTools Options' window. It has three main sections: 'Language Settings', 'General Options', and 'Help Options'.
 - 'Language Settings': 'Base Language Code' is set to 'English'. There is a checkbox for 'Translations Change Last Updated Information' which is unchecked.
 - 'General Options':
 - 'Disconnect Cursors After:' is set to '30' seconds (0 = Never).
 - 'Temp Table Instances (Total):' and 'Temp Table Instances (Online):' are both set to 0.
 - 'Multi-Currency' is checked.
 - 'Use Business Unit in nVision' is unchecked.
 - 'Multiple Jobs Allowed' is unchecked.
 - 'Allow DB Optimizer Trace' is checked.
 - 'Grant Access' is checked.
 - 'Maximum App Message Size:' is set to '10,000,000'.
 - 'Base Time Zone:' is set to 'PST'.
 - 'Last Help Context # Used:' is set to '100222'.
 - '*Data Field Length Checking:' is set to 'Others' (indicated by a dropdown arrow).
 - 'System Style Sheet:' is set to 'PSSTYLEDEF'.
 - 'Help Options':
 - 'F1 Help URL:' and 'Ctrl-F1 Help URL:' are both empty text fields.
 At the bottom left of the window is a 'Save' button.

PeopleTools Options page - Data Field Length Checking field

2. Choose a value for the **Data Field Length Checking** option.

The valid values for the Data Field Length Checking option are **DB2 MBCS**, **MBCS**, and **Others**.

Choose a value based on the character set you are using for your database.

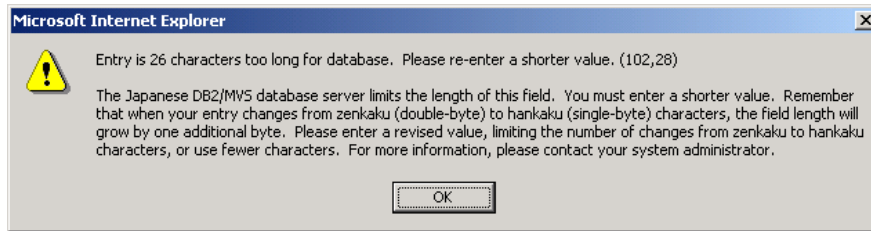
- Choose **Others** if you are using a Unicode encoded database or a non-Unicode SBCS database. This prevents special field length checking. As discussed above, these types of databases do not require such checking.
- Choose **DB2 MBCS** if you are running a Japanese database on the DB2/MVS platform. This enables field length checking based on a shifting DBCS character set.
- Choose **MBCS** if you are running a non-Unicode Japanese database on any other platform. This enables field length checking based on a non-shifting DBCS character set.



The non-Unicode DBCS settings are specifically oriented to Japanese language installations, as Japanese is the only language supported by PeopleSoft in a non-Unicode DBCS encoding. All other languages requiring double-byte character sets are supported by PeopleSoft only when using Unicode encoded databases.

Once this option is set, the PeopleTools page processor and other components will automatically validate the length of the data that is entered into page fields and Windows client dialog boxes, based on the character set used by your database.

If more characters are entered into a field than are allowed by the database, you receive an error message indicating how many characters need to be trimmed from the value entered and the reason for the requirement. The error message also contains some helpful hints specifically for Japanese users in determining how many characters may fit in a field:



Error message - too many characters in the field

3. Save the page.

International Character Handling in PeopleTools

PeopleTools contains several features that allow the manipulation of character data based on the language, script or type of character being processed. Some field formats, such as the Name format are dependent on the language of the data being maintained, and several PeopleCode functions are provided to allow manipulation of language-dependent characters.

Standard Name Formats for Chinese, Japanese, and Korean Characters

The PeopleSoft standard name format is:

```
[lastname] [suffix], [prefix] [firstname] [middle name/initial]
```

However, if the name contains any of the types of characters listed below, typically used in Chinese, Japanese, and Korean names, the first and last names must be separated by a space, instead of a comma:

```
[lastname] [suffix] [prefix] [firstname] [middle name/initial]
```

- Chinese, Japanese and Korean Unified Ideographs (Japanese “Kanji”, Korean “Hancha”)
- Japanese half-width or full-width Katakana
- Japanese Hiragana
- Korean Hangul

This format allows for Chinese, Japanese, and Korean names, which are traditionally separated by spaces and would display incorrectly if separated by a comma.

Detecting and Converting Characters

PeopleTools also provides PeopleCode string functions that recognize and convert between different characters within the Japanese character set. This lets you detect, convert, and enforce the types of characters that can be entered in any PeopleSoft field. PeopleSoft used these functions in the development of Alternate Character Architecture in some PeopleSoft

applications. This architecture is used in several PeopleSoft products to provide a feature that allows the entry of, and enforces the characters contained in, Japanese phonetic spellings (Furigana) using the Hiragana or Katakana scripts.



For more information about these functions, see the corresponding entries in PeopleCode Reference.

Common Char_Code Values

The character sets are represented by a set of *char_code* values shared by these functions, which are listed in the following table:

Character Code	Character Set
0	Alphanumeric (7-bit ASCII codes: A-Z, a-z, 1-9, punctuation)
1	Extended Latin-1 characters (ISO8859-1 accents for Spanish, French, and so forth)
2	Hankaku Katakana (half-width Japanese Katakana)
3	Zenkaku Katakana (full-width Japanese Katakana)
4	Hiragana (Japanese)
5	Unified Han Ideographs (Japanese Kanji, Chinese, Korean Hancha)
6	Double-byte alphanumeric (Japanese)
7,8,9	Reserved for future use
10	Japanese punctuation

CharType

The syntax of CharType is:

```
CharType(source_str, char_code)
```

Where

`source_str` is the first character to be test

`char_code` is a number representing the character type to be test for



For information about values to pass in `char_code` see Common Char_Code Values.

CharType returns one of the following number values:

Return Value	Description
1	Character is of type char_code.
0	Character is not of type char_code.
-1	UNKNOWN: unable to determine whether character is of set char_code. This return code was given by previous versions of PeopleTools when the character set could not be detected, but this should not occur with PeopleTools 8.

Example

This example tests to see if a character is Kanji:

```
&ISKANJI = CharType(&STRTOTEST, 5);

If (&ISKANJI = 1) Then

    WinMessage("Character type is Kanji");

Else

    If (&ISKANJI = 0) Then

        WinMessage("Character type is not Kanji");

    Else

        WinMessage("Character type is UNKNOWN");

    End-If

End-If
```

ContainsCharType

ContainsCharType determines whether any of the characters in a source string are of a specific character type. The char_code is a number value representing a character set. Its syntax is:

```
ContainsCharType(source_str, char_code)
```

Where

source_st is the string to be examined.

char_code is a number value representing the character set to be tested for.



For information about values to pass in char_code, see Common Char_Code Values.

CharType returns one of the following number values:

Return Value	Description
1	String contains at least one character of set char_code.
0	String contains no characters of set char_code.
-1	UNKNOWN: unable to determine whether one or more characters in the string is of set char_code. This return code was given by previous versions of PeopleTools when the character set could not be detected, but this should not occur with PeopleTools 8.

Example

This example tests to see if the string contains any Kanji:

```
&ANYKANJI = ContainsCharType(&STRTOTEST, 5);

If (&ANYKANJI = 1) Then

    WinMessage("There are Kanji characters");

Else

    If (&ANYKANJI = 0) Then

        WinMessage("There are no Kanji characters");

    Else

        WinMessage("UNKNOWN");

    End-If

End-If
```

ContainsOnlyCharType

ContainsOnlyCharType determines whether every character in a source string belongs to one or more of the character sets in a list of character codes. Its syntax is:

```
ContainsOnlyCharType(source_str, char_code_list)
```

Where

- source_str is the string to search
- char_code_list is a list of character set codes in the form char_code_1 [, char_code_2]...

ContainsOnlyCharType returns one of the following number values:

<i>Return Value</i>	<i>Description</i>
1	String contains only characters that belong to the sets listed in char_code_list.
0	String contains one or more characters that do not belong to sets listed in char_code_list.
-1	UNKNOWN: unable to determine whether one or more characters in the string is of set char_code. This return code was given by previous versions of PeopleTools when the character set could not be detected, but this should not occur with PeopleTools 8.



If any character is determined to be unknown, the return value is *UNKNOWN*.

Example

This example tests to see if the string is only Hiragana or punctuation:

```
&ONLYHIRAGANA = ContainsOnlyCharType(&STRTOTEST, 4, 10);

If (&ONLYHIRAGANA = 1) Then

    WinMessage("There are only Hiragana and Punctuation characters");

Else

    If (&ONLYHIRAGANA = 0) Then

        WinMessage("Mixed characters");

    Else

        WinMessage("UNKNOWN");

    End-If

End-If
```

ConvertChar

ConvertChar converts every character in a source string to a target character code type, if possible, and places the converted string in and output string. Its syntax is:

```
ConvertChar(source_str, source_char_code, &output_str, target_char_code)
```

ConvertChar supports the following conversions:

- Conversion among the Japanese Hankaku Katakana, Zenkaku Katakana, and Hiragana

characters.

- Conversion of Japanese Hankaku (half-width) Katakana, Zenkaku (full-width) Katakana, and Hiragana characters to ASCII alphabetic characters.
- Conversion of Japanese full-width alphanumeric characters to their alphanumeric equivalents.
- Conversion of Japanese full-width punctuation characters to their regular equivalents.

Other *source_str* and *target_char_code* combinations are either passed through without conversion or not supported. Character types 0 and 1 (alphanumeric and extended Latin-1) are always passed through to *output_str* without conversion.



For more information, see Supported Conversions.

UNKNOWN characters are echoed as-is to the output string. If *source_str* can be partially converted, ConvertChar will partially convert the string, echo the remaining characters to the output string as-is, and return a value of -2 (Completed with Issues).

Returns

ConvertChar returns a number with one of the following values:

<i>Value</i>	<i>Description</i>
1	String successfully converted.
0	String not converted.
-1	UNKNOWN condition: this return code was given by previous versions of PeopleTools when the character set could not be detected, but this should not occur with PeopleTools 8.
-2	Completed with issues. Conversion was executed, but one or more characters were encountered that were either not recognized or whose conversion is not supported.



If any character cannot be translated, it is echoed as-is to *output_str*. *Output_str* could therefore be a mixture of converted and non-converted characters.

Parameters

source_str String to be converted.

source_str_category Language category of input string.

Character Code	Description
0	Alphanumeric (7-bit ASCII codes: A-Z, a-z, 1-9, punctuation)
1	Extended Latin-1 Characters (ISO8859-1 accents, Spanish, French, and so forth)
2	Japanese (any)

Output_str A string variable to receive the converted string.

Target_char_code A number representing the conversion target character type.

Character Code	Description
0	Alphanumeric (7-bit ASCII codes: A-Z, a-z, 1-9, punctuation)
2	Hankaku Katakana (half-width Japanese Katakana)
3	Zenkaku Katakana (full-width Japanese Katakana)
4	Hiragana (Japanese)
6	Double-byte alphanumeric (Japanese)

The following target values are not supported; if the source string is of the same type as any of these values, then the string is passed through without conversion.

Character Code	Description
1	Extended Latin-1 Characters (ISO8859-1 accents, Spanish, French, and so forth)
5	Kanji (Japanese)
10	Japanese punctuation

Supported Conversions

The following table shows which conversions are supported, which are passed through without conversion, and which are not supported:

Source	Target	Conversion
0 (Alphanumeric ASCII)	0-6 (All supported character types)	Pass through without conversion

Source	Target	Conversion
1 (Extended Latin-1 characters)	0-6 (All supported character sets)	Pass through without conversion
2 (Hankaku Katakana)	0 (Alphanumeric)	Conversion supported
	1 (Extended Latin)	Not supported
	2 (Hankaku Katakana)	Pass through without conversion
	3 (Zenkaku Katakana)	Conversion supported
	4 (Hiragana)	Conversion supported
	5 (Kanji)	Not supported
	6 (Full-width alphanumeric)	Not supported
3 (Zenkaku Katakana)	0 (Alphanumeric)	Conversion supported
	1 (Extended Latin)	Not supported
	2 (Hankaku Katakana)	Conversion supported
	3 (Zenkaku Katakana)	Pass through without conversion
	4 (Hiragana)	Conversion supported
	5 (Kanji)	Not supported
	6 (Full-width alphanumeric)	Not supported
4 (Hiragana)	0 (Alphanumeric)	Conversion supported
	1 (Extended Latin)	Not supported
	2 (Hankaku Katakana)	Conversion supported
	3 (Zenkaku Katakana)	Conversion supported
	4 (Hiragana)	Pass through without conversion
	5 (Kanji)	Not supported
	6 (Full-width alphanumeric)	Not supported
5 (Kanji)	0-4, 6	Not supported
	5 (Kanji)	Pass through without conversion
6 (Double-byte alphanumeric)	0 (Alphanumeric)	Conversion supported
	1-5	Not supported
	6 (Full-width alphanumeric)	Pass through without conversion

Source	Target	Conversion
10 (Japanese punctuation)	0 (Alphanumeric)	Conversion supported
	1 (Extended Latin)	Not supported
	3-6, 10	Pass through without conversion

Example

This example attempts to convert a string to Hiragana:

```

&RETVALUE = ConvertChar(&INSTR, 2, &OUTSTR, 4);

If (&RETVALUE = 1) Then

    WinMessage("Conversion to Hiragana successful");

Else

    If (&RETVALUE = 0) Then

        WinMessage("Conversion to Hiragana failed");

    Else

        If (&RETVALUE = - 1) Then

            WinMessage("Input string is UNKNOWN character type.");

        Else

            WinMessage("Some characters could not be converted.");

        End-If

    End-If

End-If

```


CHAPTER 14

COBOL Globalization

Unicode support is necessary to support truly global databases. Non-Unicode databases can support a *single* language, such as Japanese, French, or English, or specific combinations of languages that use similar characters (English and German). But only Unicode databases enable you to mix dissimilar languages in the same database.

The character set used for PeopleSoft COBOL processing must match the character set for your database. If you created a Unicode database for PeopleSoft, you must also run your COBOL in Unicode.

This chapter discusses what it means to run COBOL in Unicode.



For more information about character sets and Unicode, see Understanding Character Sets.

PeopleSoft and Unicode

Unlike other character sets, Unicode has no fixed ratio of bytes per character. Due to the diverse nature of different languages, a character can be represented by 1, 2, or 3 bytes.



In this document, the word *character* refers to a single character in any language, regardless of how many bytes are required to store the character.

Four encodings of the Unicode standard are widely used. All of them are fully compatible with each other and share the same repertoire of characters. They differ, however, in how each character is represented at the byte level.

PeopleSoft uses two different encodings in different parts of the system:

- UCS-2 is the encoding that PeopleTools uses internally for the online portion of the system. UCS-2 encodes all characters of any language in to a fixed storage space of 2 bytes.
- UTF-8 is the encoding that PeopleSoft uses in COBOL. UTF-8 is a variable format of 1 to 4 bytes per character (currently no characters use 4 bytes). In UTF-8, the actual number of bytes used to store a character can be determined by the first 3 bits of the first (or only) byte of a character:

0xxxxxxx - character length is 1 byte

110xxxxx - character length is 2 bytes

111xxxxx - character length is 3 bytes

For example, the following graphic shows the different UCS-2 and UTF-8 encodings for two sample characters:

Character	UCS-2	UTF-8
a	= 0x00,0x61	= 0x61
学	= 0x5B,0x66	= 0x5B,0x66,0x22

UCS-2 and UTF-8 encodings

PeopleSoft transparently handles the conversion between UCS-2 and UTF-8 in a Unicode implementation.

- In the online system, PeopleSoft performs translations between the database platform's native character set (UCS-2 or UTF-8) and the internal PeopleTools character set (UCS-2).
- For the batch portion of the system, PeopleSoft performs translations between the database platform's native database character set and the internal COBOL character set (UTF-8).



For more information about Unicode encodings, see Unicode Encodings

COBOL and Unicode

This topic discusses aspects of COBOL that are specific to a Unicode environment.

Expanded Storage Space Requirements

Moving to a COBOL Unicode environment means that character data can potentially require three times the storage space that it would require in a single-character environment. To allow for this, all internal data definitions for character-type data in COBOL programs must be expanded to allow for three times as many bytes.

PeopleSoft provides a COBOL conversion utility to expand the character-data fields in the working storage to accommodate the number of bytes in the UTF-8 encoding scheme.



For more information about the COBOL conversion utility, see The COBOL Conversion Utility.

Special Logic for Parsing Unicode Strings

In ANSI, it's simple to parse through a string because you can assume that all characters coming in are 1 byte in length. But in a UTF-8 Unicode character set, a character can vary between 1 byte and 3 bytes in length, so special logic needs to be incorporated to handle string parsing when you are dealing with characters in UTF-8 format.



For more information about determining whether a string uses ANSI or Unicode encoding, see *Identifying Unicode and ANSI Data*. For more information about parsing strings, see *Single Character Arrays*.

Unicode-Specific Error Messages

The following error messages can occur when you are running a COBOL program against a Unicode database:

- Fetch failed: Unsuccessful UCS-2 to UTF-8 conversion on column <column number>
- Bind of parameter failed: Unsuccessful UTF-8 to UCS-2 conversion on column <column number>
- Attempting to use an ANSI API to access a Unicode database
- Attempting to use an ANSI COBOL with a Unicode database
- Attempting to use a Unicode API to access an ANSI database
- Fetch failed: The converted Unicode length of <length> exceeds the allocated buffer length <length> for column <column number>

These messages are displayed in the COBOL output log file.

The COBOL Conversion Utility

Moving to a COBOL Unicode environment means that character data can potentially require three times the storage space that it would require in a single-character environment. To allow for this, all internal data definitions for character-type data in COBOL programs must be expanded to allow for three times as many bytes.

In an ANSI implementation, one character occupies 1 byte of storage. So, for a 10-character field, you can define a PICTURE clause of PIC X(10). In a Unicode implementation, however, you must allow for the maximum number of storage bytes that could be required for any character field. Therefore, in the Unicode environment, this same 10-character field must be defined with a PICTURE clause of PIC X(30).

PeopleSoft provides a COBOL conversion utility to expand the character fields in the working storage, to accommodate the number of bytes in the UTF-8 encoding scheme.

Running the Conversion Utility

Use the following command syntax to run the COBOL conversion utility:

```
<PS_HOME>\bin\client\winx86\pscblucvrt.exe -s:<Source Directory>
-t:<Destination Directory> [-r:<TEMP Directory>]
```

Command	Description
-s:<Source Directory>	Specify the source directory where the ANSI version of COBOL resides. For the directory, you must specify where the COBOL subdirectories reside (\BASE, \WIN32, \UNIX, and so on). Example: -s:d:\PT8\SRC\CBL
-t:<Destination Directory>	Specify where you want to place the expanded version of COBOL. The utility puts the modified source file in the same COBOL subdirectory it was found in. Example: -t:d:\PT8\SRC\CBLUNICODE
-r or -rd:<Temp directory>	(Optional) If you want the utility to generate the exception log files, specify the directory where you want the utility to place these output files. -r generates only the summary log file; -rd generates all the log files.

The utility produces a new source file for each .CBL file that is found. These new files are placed in the PS_HOME\src\.... directory.

As delivered by PeopleSoft, the batch utilities that compile your COBOL programs include logic to convert all programs and copybooks prior to compiling. This logic is only triggered only when the Unicode version of PeopleTools is installed.

Compiling COBOL in Windows NT or Windows 2000

The <PS_HOME>\setup\cbl2uni.bat command converts all the COBOL programs and copybooks found in the <PS_HOME>\src\cbl directory. After the conversion, you'll find your COBOL files in the following directories:

- <PS_HOME>\src\cblansi contains the ANSI version of the COBOL source codes.
- <PS_HOME>\src\cbl contains the expanded COBOL source codes

Compiling COBOL in UNIX

The <PS_HOME>/install/pscbl.mak command triggers the conversion before any COBOL programs are compiled. This utility stores all converted programs in the <PS_HOME>/src/cblunicode directory.

Identifying Converted COBOL Programs

When the COBOL conversion utility runs, it places a comment at the beginning of each COBOL program that it converts:

```
***** Converted for Unicode: <date converted>
```

```
***** Converted for Unicode: April 01, 1999
IDENTIFICATION DIVISION.

PROGRAM-ID. PSPARRY.

ENVIRONMENT DIVISION.

*****
*                                     *
*           Confidentiality Information:           *
*                                     *
* This module is the confidential and proprietary information of *
```

Sample code with conversion stamp

This comment line identifies converted programs in two ways:

- A person looking at the program can tell whether it has been converted.
- If you attempt to convert the COBOL source file again, this comment line prevents the conversion utility program from expanding the working storage of this COBOL source file again.

Understanding What Is Expanded

In order for the utility to recognize when it is appropriate to expand data, strict adherence to the PeopleSoft COBOL coding standards is required. The utility looks for certain code-style patterns in order to make these decisions.

The conversion utility expands all PIC X[(N)] data fields to triple their original size, with the following exceptions:

- SQL Buffer setup data that refers to numeric or date data types SELECT-SETUP or BIND-SETUP is not expanded.
- Character fields that are redefined to a numeric field and group-level fields that contain such character fields are not expanded. In instances where the redefined field is also redefined as a character field, the original character field and the redefinition that is a character field will be expanded.
- Fields and group-level fields that appear to be dates are not expanded, unless the EXPAND directive is specified for this field or group-level field.
- For arrays that are composed of a single character, the PIC clause is expanded for character data, but the OCCURS clause is not expanded. However, if the data name ends with *-POS*, *-CHAR* or *-BYTE*, the OCCURS clause is expanded, instead of the element size.

The utility also converts copybooks on the fly: The first time that a copybook is referenced inside a code module, it is expanded immediately.

The utility processes an entire set of COBOL modules in a single run. It maintains a record of what it has converted in order to avoid converting copybooks twice.



The COBOL conversion utility ensures that edited lines do not past the 72nd column. If the conversion would normally cause a line to exceed that limitation, the utility removes some of the blank spaces between the field name and the PIC X string so that the line fits in the allowed area.

Exception 1: Certain SQL Buffer Setup Data

SQL Buffer setup data that refers to numeric or date data types SELECT-SETUP or BIND-SETUP is not expanded.

For the interface to PTPSQLRT, a COBOL program passes a SELECT list (SELECT-DATA) and a descriptor area (SELECT-SETUP). The program also passes similar data and setup areas for bind variables. The descriptors that are passed are always character-type data with embedded values that signal the actual data type and length of the data fields. Because these descriptors represent the lengths of the associated data fields in the corresponding SELECT-DATA/BIND-DATA structures, the utility adjusts only the length of the descriptors that are representing character-type data.

Example 1

In the following code, the select list contains two character fields (EMPLID and NAME), a small integer (EMPL_RCD), and a date (EFFDT):

```

SELECT-SETUP

02  FILLER PIC X(20)      VALUE ALL 'C' .

02  FILLER PIC X(2)       VALUE ALL 'S' .

02  FILLER PIC X(10)      VALUE ALL 'D' .

02  FILLER PIC X(30)      VALUE ALL 'C' .

SELECT-DATA.

02  EMPLID PIC X(20) .

02  EMPL_RCD      PIC 99 COMP .

02  EFFDT  PIC X(10) .

02  NAME      PIC X(30) .

```

In the Unicode world, the only fields that should be expanded are the two character fields (EMPLID and NAME). Numeric data is never impacted by Unicode, and (according to the

PeopleTools definition), dates are not impacted either: they are treated as numeric strings and cannot have special characters.

Thus the utility converts this code as follows:

```
SELECT-SETUP.

02  FILLER PIC X(60)      VALUE ALL 'C' .

02  FILLER PIC X(2)       VALUE ALL 'S' .

02  FILLER PIC X(10)      VALUE ALL 'D' .

02  FILLER PIC X(90)      VALUE ALL 'C' .

SELECT-DATA.

02  EMPLID PIC X(60) .

02  EMPL_RCD      PIC 99 COMP.

02  EFFDT  PIC X(10) .

02  NAME        PIC X(90) .
```

Example 2

The following code represents ANSI COBOL (COBOL that has not yet been expanded):

```
01  I-ERRL.

      05  SQL-STMT          PIC X(18)  VALUE
                                'INPXPROC_I_ERRL'.

      05  BIND-SETUP.

          10  FILLER          PIC X(10)  VALUE ALL 'C'.

          10  FILLER          PIC X(6)   VALUE '0PPPPP'.

          10  FILLER          PIC X(4)   VALUE ALL 'I'.

          10  FILLER          PIC X      VALUE 'H'.

          10  FILLER          PIC X(18)  VALUE ALL 'C'.

          10  FILLER          PIC X(4)   VALUE ALL 'I'.

          10  FILLER          PIC X(4)   VALUE ALL 'N'.

          10  FILLER          PIC X(30)  VALUE ALL 'H'.

          10  FILLER          PIC X(30)  VALUE ALL 'C'.

          10  FILLER          PIC X(30)  VALUE ALL 'H'.
```

```

10  FILLER          PIC X(10)  VALUE ALL 'C'.

10  FILLER          PIC X(6)   VALUE '0PPPPP'.

10  FILLER          PIC X(8)   VALUE '0PPPPPPP'.

10  FILLER          PIC X      VALUE 'Z'.

05  BIND-DATA.

10  TSE-JOBID       PIC X(10)  VALUE SPACES.

10  TSE-PROC-INSTANCE PIC 9(10) VALUE ZERO  COMP-3.

10  TSE-SET-NBR     PIC 9(6)   VALUE ZERO  COMP.

10  TSE-EDIT-TYPE   PIC X      VALUE SPACE.

10  TSE-FIELDNAME   PIC X(18)  VALUE SPACES.

10  MESSAGE-SET-NBR PIC 9(5)   VALUE ZERO  COMP.

10  MESSAGE-NBR     PIC 9(5)   VALUE ZERO  COMP.

10  MESSAGE-PARM    PIC X(30)  VALUE SPACES.

10  MESSAGE-PARM2   PIC X(30)  VALUE SPACES.

10  MESSAGE-PARM3   PIC X(30)  VALUE SPACES.

10  BUSINESS-UNIT   PIC X(10)  VALUE SPACES.

10  TRANSACTION-NBR PIC 9(10)  VALUE ZERO  COMP-3.

10  SEQ-NBR         PIC 9(15)  VALUE ZERO  COMP-3.

10  FILLER          PIC X      VALUE 'Z'.

```

The utility converts this code as follows:

```

01  I-ERRL.

05  SQL-STMT        PIC X(54)  VALUE

                                'INPXPROC_I_ERRL'.

05  BIND-SETUP.

10  FILLER          PIC X(30)  VALUE ALL 'C'.

10  FILLER          PIC X(6)   VALUE '0PPPPP'.

10  FILLER          PIC X(4)   VALUE ALL 'I'.

10  FILLER          PIC X(3)   VALUE ALL 'H'.

10  FILLER          PIC X(54)  VALUE ALL 'C'.

```



```

10  FILLER          PIC X(4)    VALUE ALL 'I'.
10  FILLER          PIC X(4)    VALUE ALL 'N'.
10  FILLER          PIC X(90)   VALUE ALL 'H'.
10  FILLER          PIC X(90)   VALUE ALL 'C'.
10  FILLER          PIC X(90)   VALUE ALL 'H'.
10  FILLER          PIC X(30)   VALUE ALL 'C'.
10  FILLER          PIC X(6)    VALUE '0PPPPP'.
10  FILLER          PIC X(8)    VALUE '0PPPPPPP'.
10  FILLER          PIC X      VALUE 'Z'.

05  BIND-DATA.

10  TSE-JOBID       PIC X(30)   VALUE SPACES.
10  TSE-PROC-INSTANCE PIC 9(10) VALUE ZERO  COMP-3.
10  TSE-SET-NBR     PIC 9(6)    VALUE ZERO  COMP.
10  TSE-EDIT-TYPE   PIC X(3)    VALUE SPACES.
10  TSE-FIELDNAME   PIC X(54)   VALUE SPACES.
10  MESSAGE-SET-NBR PIC 9(5)    VALUE ZERO  COMP.
10  MESSAGE-NBR     PIC 9(5)    VALUE ZERO  COMP.
10  MESSAGE-PARM     PIC X(90)   VALUE SPACES.
10  MESSAGE-PARM2    PIC X(90)   VALUE SPACES.
10  MESSAGE-PARM3    PIC X(90)   VALUE SPACES.
10  BUSINESS-UNIT   PIC X(30)   VALUE SPACES.
10  TRANSACTION-NBR PIC 9(10)   VALUE ZERO  COMP-3.
10  SEQ-NBR         PIC 9(15)   VALUE ZERO  COMP-3.
10  FILLER          PIC X      VALUE 'Z'.

```

Exception 2: Certain Redefined Character Fields

Character fields that are redefined to a numeric field, and group-level fields that contain such character fields, will not be expanded. In instances where the redefined field is also redefined as a character field, the original character field and the redefinition that is a character field *will* be expanded.

Example 1

In this example, the DB-PIC-PRECIS-CHAR will not be expanded:

```
07 DB-PIC-PRECIS-CHAR PIC X(2) .

07 DB-PIC-PRECIS-NUM REDEFINES
    DB-PIC-PRECIS-CHAR PIC 9(2) .
```

Example 2

In this example, the I-REMIT-ADDR-SEQ will not be expanded:

```
02 I-REMIT-ADDR-SEQ PIC 9(04) .

02 I-REMIT-ADDR-SEQ-C REDEFINES
    I-REMIT-ADDR-SEQ PIC X(04) .
```

Example 3

In this example, the original definition is a character-type field. Although some of the redefined fields are numeric fields, all the character fields, including the original definition, will be expanded.

```
02 MSGDATA1 PIC X(30) VALUE SPACES .

02 FILLER REDEFINES MSGDATA1 .
    03 MSGDATA1-INT PIC Z(9)9- .
    03 INT-FILL1 PIC X(19) .

02 FILLER REDEFINES MSGDATA1 .
    03 MSGDATA1-DOL PIC Z(9)9.99- .
    03 DOL-FILL1 PIC X(16) .

02 FILLER REDEFINES MSGDATA1 .
    03 MSGDATA1-DEC PIC Z(9)9.9(5)- .
    03 DEC-FILL1 PIC X(13) .
```

Exception 3: Certain Fields That Appear to Be Dates

Fields and group-level fields that appear to be dates will not be expanded, unless the EXPAND directive is specified for this field or group-level field.



For more information about overriding this exception with the EXPAND directive, see Using Utility Directives.

The following table describes the criteria used to determine fields or group-level fields as dates:

DATE Data Type	Field or Group-Level Field Name	Field Length or Total Length of a Group-Level Field*
Date	Contains "-DT' or "DATE"	10
Time	Contains "-TM' or "TIME"	15
Date-Time	Contains "-DTTM", "DATE" or "TIME"	26

* When calculating the total length, the utility takes into consideration that a group-level field may contain REDEFINE fields. The length of the REDEFINE field will not be included in determining the total length of the group field.

Example 1

The field in this example will not be expanded:

```
10  START-DATE          PIC X(10)  VALUE SPACES.
```

Example 2:

The fields in this example will not be expanded:

```
02  W-EMP-BIRTHDATE.

03  YEAR                PIC 9(4) .

03  FILLER              PIC X(1) .

03  MONTH              PIC 99 .

03  FILLER              PIC X(1) .

03  DAYS               PIC 99 .
```

Example 3

The field in this example will not be expanded:

```
03  PAY-DATE-TIME.

04  PAY-DTTM-DATE      PIC X(10) .

04  PAY-DTTM-DELIM1 PIC X          VALUE '- ' .

04  PAY-DTTM-TIME      PIC X(15) .
```

Example 4

The field in this example will not be expanded:

```
05  BEGIN-DTTM-TIME.
```

```

07  SYS-HOUR          PIC 99      VALUE ZERO.

07  FILLER            PIC X       VALUE SPACE.

07  SYS-MINUTE        PIC 99      VALUE ZERO.

07  FILLER            PIC X       VALUE SPACE.

07  SYS-SECOND        PIC 99      VALUE ZERO.

07  FILLER            PIC X       VALUE SPACE.

07  SYS-MICRO-SECOND PIC 9(6)    VALUE ZERO.

```

Example 5

In this example, the group field contains REDEFINE fields. The conversion utility will expand the fields because the group field meets the criteria for expansion: it has a total length of 10 and the field name includes the *-DT* string.

```

02  END-DT.

03  END-DT-YY          PIC X(4) .

03  END-DT-YY-NUM      REDEFINES END-DT-YY
PIC 9999.

03  FILLER             PIC X.

03  END-DT-MM          PIC XX.

03  END-DT-MM-NUM      REDEFINES END-DT-MM
PIC 99.

03  FILLER             PIC X.

03  END-DT-DD          PIC XX.

03  END-DT-DD-NUM      REDEFINES END-DT-DD
PIC 99.

```

Exception 4: Arrays Composed of a Single Character

For arrays that are composed of a single character, the PIC clause is expanded for character data, but the OCCURS clause is not expanded. However, if the data name ends with *-POS*, *-CHAR*, or *-BYTE*, the OCCURS clause will be expanded, instead of the element size.

Example 1

In this example, the field is expanded:

```

01  CHAR-ARRAY        PIC X          OCCURS 80 TIMES.

```

Is expanded to:

```
01 CHAR-ARRAY      PIC X(3)      OCCURS 80 TIMES.
```

Example 2

In this example, the data name ends with *-POS*; therefore, the OCCURS clause is expanded, instead of the element size:

```
01 CHAR-POS  PIC X      OCCURS 80 TIMES.
```

Is expanded to:

```
01 CHAR-POS  PIC X      OCCURS 240 TIMES.
```

Using Utility Directives

The COBOL conversion utility accepts various directives in the first six columns of COBOL code. You can use these directives to override the utility's normal mode of processing for a single source code line or for a block of lines:

Directive	Description	Purpose
NOCBGN	No Conversion: Begin	Starting with this line, do not perform expansions.
NOCEND	No Conversion: End	End the NOCBGN directive after this line (the directive line will not be expanded).
NOCLN	No Conversion: Line	Do not perform expansions in this single line.
COCCUR	Convert Occurrence	Expand the OCCURS clause instead of the PIC clause in this line.
EXPEOF	Expand end of field	Expand a group item by increasing the length of the last field in the group.
EXPAND	Instruct utility to expand field	Force expansion of fields that would normally not be expanded because they appear to be date, time, or datetime fields.

Sample Uses for Utility Directives

The following examples use existing PeopleSoft COBOL programs to illustrate possible uses for the utility directives.

NOCBGN, NOCEND, and NOCLN Directives

One of the PeopleSoft Human Resources COBOL programs has a unique way of setting the PAY-PERIODS group field. The program defines an 88 level definition based on the

concatenated value of the five one-column, character-type fields. If the conversion utility were to convert the program without the special directives, none of the cases defined in the 88-level field would ever be true:

```

NOCBGN      03  PAY-PERIODS.

                88  PAY-PERIODS-ALL              VALUE 'YYYYY'.

                88  PAY-PERIODS-ALL-BIWEEKLY      VALUE 'YYNN'.

                88  PAY-PERIODS-ALL-SEMIMONTHLY   VALUE 'YNNN'.

                88  PAY-PERIODS-NONE              VALUE 'NNNNN'.

                04  PAY-PERIOD1          PIC X.

                04  PAY-PERIOD2          PIC X.

                04  PAY-PERIOD3          PIC X.

                04  PAY-PERIOD4          PIC X.

                04  PAY-PERIOD5          PIC X.

                03  FILLER REDEFINES PAY-PERIODS.

                04  PAY-PERIOD          PIC X      OCCURS 5.

                        88  PAY-PERIOD-YES      VALUE 'Y'.

NOCEND      88  PAY-PERIOD-NO              VALUE 'N'.

01  S-DEDPDS.

02  SQL-STMT          PIC X(18)  VALUE

                        'PSPDCFSA_S_DEDPDS'.

02  BIND-SETUP.

03  FILLER          PIC X(10)  VALUE ALL 'C'.

03  FILLER          PIC X(10)  VALUE ALL 'H'.

03  FILLER          PIC X(10)  VALUE ALL 'D'.

03  FILLER          PIC X(10)  VALUE ALL 'A'.

NOCBGN      03  FILLER          PIC X      VALUE ALL 'C'.

03  FILLER          PIC X      VALUE ALL 'H'.

03  FILLER          PIC X      VALUE ALL 'C'.

03  FILLER          PIC X      VALUE ALL 'H'.

NOCEND      03  FILLER          PIC X      VALUE ALL 'C'.

```

```

03  FILLER                      PIC X          VALUE 'Z' .

02  BIND-DATA .

03  COMPANY                     PIC X(10) .

03  PAYGROUP                     PIC X(10) .

03  PAY-END-DT                   PIC X(10) .

03  YEAR-END-DT                  PIC X(10) .

NOCLN 03  PAY-PERIODS             PIC X(5) .

03  FILLER                      PIC X          VALUE 'Z' .

```

COCCUR Directive

The conversion utility wouldn't normally expand the size of the array in this line from one of the PeopleTools COBOL programs. Use of the COCCUR directive ensures that the OCCURS clause is expanded:

```

02  PARM .

COCCUR 05  PARM-CH                OCCURS 30 TIMES

PIC X .

```

EXPEOF Directive

In the example below, the FIELDNAME group-level field is broken down to check the first 4 characters of the string. In this instance, it makes more sense to adjust the length of the FILLER field. By using the EXPEOF directive, you direct the utility to expand the FILLER field to a length of 50:

```

EXPEOF 02  FIELDNAME .

03  FIELDNAME4                   PIC X(4)        VALUE SPACE .

      88  FIELDNAME-TSE                      VALUE 'TSE_' .

03  FILLER                       PIC X(14)       VALUE SPACE .

```

Understanding Error Logs

The COBOL conversion utility produces a set of error and warning logs with messages that identify non-standard code styles and inconsistencies. The utility also logs expansion actions that may require manual review.

The utility produces the following logs:

- Exception Log

This log contains warnings that occurred because of ambiguous working storage definitions. You may need to modify code or add utility directives to resolve the issues logged.

- **Exception BIND/SELECT Log**

This log contains warnings and errors that occurred because of ambiguous BIND and SETUP definitions.

- **Exception Date Log**

This log lists all group-level date fields detected by the utility.

- **Summary**

This log provides general statistics regarding the number of programs processed.

When you specify the -4 flag, you only see only the summary log. Set the -rd flag on the conversion utility command line if you want the utility to produce all the detail logs: the Exception Log, the BIND/SELECT log, and the Exception Date Log.

The following tables summarize all the messages that can be displayed in the three Exception log files. Errors indicate problems that are encountered by the conversion utility; warnings indicate that you need to verify an ambiguous field definition.

Messages From the Exception Log

Message	Type	Note
Non-matching conversion block found in line <line number>	Error	Detected the NOCBGN directive, but couldn't find the corresponding NOCEND.
Error in determining numeric length in line <line number>	Error	Program couldn't decipher the numeric PICTURE clause.
The size of the one character array will be expanded in line <line number>	Warning	Detected a one-character array where field contains the string -BYTE, -POS, or -CHAR.
A one character array is found in <line number>. The conversion routine will expand this to PIC X(3)	Warning	
Unable to find the copy library <copy library name>	Error	Program references a copy library, but the file couldn't be located.

Messages From the Exception BIND/SELECT Log

Message	Type	Note
Didn't find the corresponding DATA section for <DATA field name> in line <line number>	Error	Detects either a BIND-DATA or a SELECT-DATA, but cannot find the SETUP group field.

number>		
No delimiter found on <group field name> section in line <line number>	Warning	Didn't find a FILLER field with a value Z in a DATA or SETUP group-field.
Unable to convert the <group field name> section due to problems reading the Copylib	Error	DATA or SETUP refers to a copy library that cannot be located.
The <group field name> found in line "<line number>" has a mismatch count	Warning/Error	The number of columns in DATA doesn't match the count for the corresponding SETUP.
Incompatible date type match for field in line <line number>	Error	The data type definition in SETUP doesn't correspond to the data type in DATA.

Messages From the Exception Date Log

Message	Type	Note
Date/time/datetime detected and will not be expanded in line	Warning	
Verify if a date/time/datetime field in line number: <line number>	Warning	Found a character-type field or group field with a total length of 10, 15, or 26 that could be a date, time, or datetime.

Manual Fine-Tuning of COBOL Programs

Although the COBOL conversion makes most of the changes that are needed to run COBOL in a Unicode environment, some manual fine-tuning may still be necessary.

Identifying Unicode and ANSI Data

Your COBOL program may need to determine whether it's dealing with ANSI or COBOL data. For example, if your program parses a string character, it needs to apply different logic depending on whether the string is an ANSI string or a Unicode string. The program can get this information from the ENCODING-MODE-SW in the PTCSQLRT copy library:

```

03  ENCODING-MODE-SW      PIC X(3)      VALUE SPACE.

                                88  ANSI-MODE                      VALUE 'A' .

                                88  UNICODE-MODE                   VALUE 'U' .

```

The ENCODING-MODE-SW value is set by the COBOL API, which determines which type of data it's dealing with by checking the value of the UNICODE_ENABLED field in the

PSSTATUS table. When the value of the UNICODE_ENABLED flag is set to 1 (TRUE), this signals the COBOL API that it is accessing a Unicode database.

The PeopleSoft COBOL trace file provides information about which encoding scheme COBOL API used:

PeopleSoft Batch Timings Report: cobsql_0416165133.trc								
Encoding Scheme Used: Unicode								
PeopleSoft Batch Statistics (All timings in seconds)								
Statement	Retrieve Count	Retrieve Time	Compile Count	Compile Time	Execute Count	Execute Time	Fetch Count	Fetch Time
APIBNN	0	0.00	0	0.00	266043	4.71	0	0.00
APISSB	0	0.00	0	0.00	1315	0.00	0	0.00
COMMIT	0	0.00	0	0.00	10	0.22	0	0.00
CONNECT	0	0.00	0	0.00	75	0.92	0	0.00
DISCONNECT	0	0.00	0	0.00	75	1.91	0	0.00

Sample trace file

Note that the COBOL API also performs the necessary translations between the UTF-8 encodings that are required by COBOL and the UCS-2 encodings that are used elsewhere in the PeopleSoft system.

Specifying Column Lengths in Dynamic SQL

Perhaps the biggest effort in getting COBOL fully functional in a Unicode environment is setting up the bind parameters and select buffers of any dynamic SQL statements.

Programs that use dynamic SQL need to specify the column lengths for bind or select fields prior to calling the PTPDYSQL program. Within a COBOL program, there are two ways bind parameters and select buffers of dynamic SQL statements can be assigned:

- Using a predefined working storage area in conjunction with the dynamic SQL statement. This is very similar to the method used for stored SQL statements.

In this case, PTPDYSQL adjusts the length of character-data fields that are passed to PTPSQLRT. This is necessary because the COBOL Unicode conversion utility expands only the working storage fields; it does not modify the length of fields that are hard-coded in the PROGRAM-DIVISION section of your COBOL programs.

Because PTPDYSQL sends the correct length to PTPSQLRT, no changes to the COBOL program are necessary.

- Use a buffer array. At runtime, this array is partitioned, based on the properties of all the fields that are referenced by the dynamic SQL statement. The properties of those fields are retrieved from the PSDBFIELD table; they include both the field's data type and the field's length.

In this case, you must modify the COBOL program to adjust the length specified for a character field. Adjust the length by a factor of 3.

To adjust the length of the character field appropriately, the program needs to recognize the encoding scheme that is used by the COBOL API. The program can take advantage of the ENCODING-MODE-SW field in PTCSQLRT to determine when the length of the field needs to be adjusted.



For more information about the ENCODING-MODE-SW field, see Identifying Unicode and ANSI Data.

The following code section illustrates the use of a buffer array to calculate the length of a character field in the Unicode environment:

```

02  SELECT-DATA.

      03  FIELDNAME          PIC X(54)    VALUE SPACE.

      03  FIELDNUM           PIC 9(3)     VALUE ZERO  COMP.

      03  DEFFIELDNAME       PIC X(90)    VALUE SPACES.

      03  FIELDLEN           PIC 9(3)     VALUE ZERO  COMP.

      03  FIELDTYPE          PIC 9(2)     VALUE ZERO  COMP.

      88  RDM-CHAR                      VALUE ZERO.

      88  RDM-LONG-CHAR                  VALUE 1.

      88  RDM-NUMBER                     VALUE 2.

      88  RDM-SIGNED-NUMBER               VALUE 3.

      88  RDM-DATE                       VALUE 4.

      88  RDM-TIME                       VALUE 5.

      88  RDM-DATETIME                    VALUE 6.

      03  DECIMALPOS           PIC 9(2)    VALUE ZERO  COMP.

      03  FILLER               PIC X      VALUE 'Z'.

      88  RDM-NUMBER                     VALUE 2.

      88  RDM-SIGNED-NUMBER               VALUE 3.

      88  RDM-DATE                       VALUE 4.

      88  RDM-TIME                       VALUE 5.

      88  RDM-DATETIME                    VALUE 6.

      03  DECIMALPOS           PIC 9(2)    VALUE ZERO  COMP.

      03  FILLER               PIC X      VALUE 'Z'.

```

```

      . . .

      . . .

      . . .

      . . .

MOVE CORR SELECT-DATA OF S-RECFLD

      TO FLD-ARRAY OF RECFLD (FLD-IDX)

      * CONVERT FIELD TYPE FROM PSDBFIELD TYPE TO SQLRT CODE.

MOVE FIELDLEN OF S-RECFLD TO SETUP-LENGTH OF RECFLD (FLD-IDX)

IF RDM-CHAR OF S-RECFLD

      SET SETUP-TYPE-CHAR OF RECFLD (FLD-IDX) TO TRUE

      IF UNICODE-MODE OF SQLRT

          COMPUTE SETUP-LENGTH OF RECFLD (FLD-IDX) =

              SETUP-LENGTH OF RECFLD (FLD-IDX) * 3

      END-IF

      END-IF

```

Single Character Arrays

Some COBOL programs define single-character arrays in order to parse or examine a string of characters, one character at a time. In a Unicode environment, you need to be sure that you're examining the string one *character* at a time, not one *byte* at a time.

Following is an example of a code fragment in which the program is examining a string, one character at a time:

```

01  CHAR-ARRAY.

      02  CHAR-POS          PIC X  OCCURS 256 TIMES

          INDEXED BY CHAR-IDX.

      88  FIELD-DELIM          VALUE '*' .

      . . .

      . . .

      SET CHAR-IDX  TO  1

      SEARCH CHAR-ARRAY

```

```

      WHEN FIELD-DELIM (CHAR-IDX)

          SET W-OFFSET TO CHAR-IDX

          DISPLAY 'FIELD DELIMITER FOUND AT POSITION ' W-OFFSET

      END-SEARCH

```

The intent of the code above is to examine each character of the array, looking for the first delimiter character. When that character is found, display the position of the delimiter.

In an ANSI environment that uses only the Latin1 character set, this works because we know that there is 1 byte (one array element) per character. In a Unicode environment (or in an ANSI environment that allows double-byte character sets), this fails because we could potentially be examining the second or third byte of a two- or three-byte character. It is possible for the second or third byte to match the bit pattern of the delimiter character, thus falsely passing the test and ending the search loop.

How do you correct this situation? The solution lies in knowing the length (in bytes) of each character that we're processing. A new COBOL function, PTPSTRFN, is available that returns the length of a character so that the code can take this into account when performing a character search. The PTPSTRFN subroutine works for both Unicode character set sets and ANSI double-byte characters sets.

The PTPSTRFN subroutine offers two ways of retrieving the character length of a character database type field:

- Requesting the length of a single character
- Requesting a map of an entire character string. Choose this option if your application program needs to get the length information of all characters within a string.



For more information about mapping entire character strings, see Example.

Requesting the Length of a Single Character

Input Parameters

<i>Parameter</i>	<i>Value</i>	<i>Notes</i>
ACTION-TYPE	ACTION-CHARLEN	
CHAR-CD	The character whose length you want to verify the length.	This variable is included in the PTCSTRFN.CBL copy library.

Values Returned

<i>Parameter</i>	<i>Value</i>	<i>Notes</i>
CHAR-LENGTH	The subroutine returns one of the following values, representing the length of the character referenced by CHAR-CD: <ul style="list-style-type: none"> • ONE-BYTE • TWO-BYTES • THREE-BYTES 	This variable is included in the PTCSTRFN.CBL copy library.
STRFN-RC	Returns one of the following values: <ul style="list-style-type: none"> • STRFN-RC-OK • STRFN-INVALID-ACTION 	This variable is included in the PTCSTRFN.CBL copy library.

Example

At the beginning of this topic, there was a code fragment in which the program was examining a string, one character at a time, looking for the first delimiter character:

```

01  CHAR-ARRAY.

02  CHAR-POS          PIC X  OCCURS 256 TIMES
                        INDEXED BY CHAR-IDX.

      88  FIELD-DELIM          VALUE '*' .

. . .

. . .

SET CHAR-IDX  TO  1

SEARCH CHAR-ARRAY

    WHEN FIELD-DELIM(CHAR-IDX)

        SET W-OFFSET  TO  CHAR-IDX

        DISPLAY 'FIELD DELIMITER FOUND AT POSITION ' W-OFFSET

END-SEARCH

```

After being modified for Unicode, the code looks like this:

```

01  CHAR-ARRAY.

02  CHAR-POS          PIC X  OCCURS 256 TIMES
                        INDEXED BY CHAR-IDX.

```

```

      88 FIELD-DELIM                VALUE '*' .

01 STR-FUNC                        COPY 'PTCSTRFN' .

. . .

. . .

SET CHAR-IDX TO 1

SEARCH CHAR-ARRAY

    WHEN FIELD-DELIM(CHAR-IDX)

        SET W-OFFSET TO CHAR-IDX

        DISPLAY 'FIELD DELIMITER FOUND AT POSITION ' W-OFFSET

    WHEN OTHER

        MOVE CHAR-POS(CHAR-IDX) TO CHAR-CD OF STR-FUNC

        CALL 'PTPSTRFN' USING ACTION-CHARLEN

                                STR-FUNC

        IF TWO-BYTES OF STR-FUNC

            SET CHAR-IDX UP BY 1

        ELSE IF THREE-BYTES OF STR-FUNC

            SET CHAR-IDX UP BY 2

        END-IF

    END-SEARCH

```

The above modification ensures that the code will continue to function properly in a Unicode environment. However, we can only be sure that modification works when the character the program is searching for is 1 byte in length.

Consider the following code fragment, where the character that the program is searching for may be longer than 1 byte:

```

01 W-DELIMITER                    PIC X(3)        VALUE '<some extended character>' .

01 CHAR-ARRAY .

    02 CHAR-POS                    PIC X OCCURS 256 TIMES

                                INDEXED BY CHAR-IDX

                                CHAR-IDX2

                                CHAR-IDX3 .

```

```

      88 FIELD-DELIM                VALUE '*' .

      . . .

      . . .

SET CHAR-IDX  TO  1

SEARCH CHAR-ARRAY

      WHEN CHAR-POS (CHAR-IDX)  =  W-DELIMITER

      SET W-OFFSET  TO  CHAR-IDX

      DISPLAY 'FIELD DELIMITER FOUND AT POSITION ' W-OFFSET

END-SEARCH

```

In order for this code to work in a Unicode environment, a Unicode-specific search algorithm must be used. You need to ensure that the program always compares the correct bytes from the array (up to 3 bytes, based on the current character length) to the fixed 3-byte field containing the search value.

The proper search method would look like this:

```

01 CHAR-ARRAY.

02 CHAR-POS          PIC X OCCURS 256 TIMES

                        INDEXED BY CHAR-IDX.

      88 FIELD-DELIM                VALUE '*' .

01 STR-FUNC          COPY 'PTCSTRFN' .

      . . .

      . . .

SET CHAR-IDX  TO  1

PERFORM UNTIL CHAR-IDX > 256

      MOVE CHAR-POS (CHAR-IDX)  TO CHAR-CD OF STR-FUNC

      CALL 'PTPSTRFN'  USING  ACTION-CHARLEN

                        STR-FUNC

      INITIALIZE W-WORK

      EVALUATE TRUE

      WHEN ONE-BYTE OF STR-FUNC

      MOVE CHAR-POS (CHAR-IDX)  TO  W-WORK

```



```
        SET CHAR-IDX UP BY 1

    WHEN TWO-BYTES OF STR-FUNC

        SET CHAR-IDX2 TO CHAR-IDX

        SET CHAR-IDX2 UP BY 1

        STRING CHAR-POS (CHAR-IDX)

            CHAR-POS (CHAR-IDX2)

            DELIMITED BY SIZE

            INTO W-WORK

        END-STRING

        SET CHAR-IDX UP BY 2

    WHEN THREE-BYTES OF STR-FUNC

        SET CHAR-IDX2 TO CHAR-IDX

        SET CHAR-IDX2 UP BY 1

        SET CHAR-IDX3 TO CHAR-IDX

        SET CHAR-IDX3 UP BY 2

        STRING CHAR-POS (CHAR-IDX)

            CHAR-POS (CHAR-IDX2)

            CHAR-POS (CHAR-IDX3)

            DELIMITED BY SIZE

            INTO W-WORK

        END-STRING

        SET CHAR-IDX UP BY 3

    WHEN OTHER

        DISPLAY '**ERROR** INVALID CHARACTER LENGTH'

        <ABEND>

    END-EVALUATE

    IF W-WORK = W-DELIMITER

        SET W-OFFSET TO CHAR-IDX

        DISPLAY 'FIELD DELIMITER FOUND AT POSITION ` W-OFFSET
```

END-IF

END-PERFORM

As you can see from the above example, searching a string array for a particular value that may be an extended character can be very difficult; if possible, you should avoid such a search.

Requesting a Map of an Entire Character String

<i>Parameter</i>	<i>Value</i>	<i>Notes</i>
ACTION-TYPE	ACTION-STRMAP.	
STRING-LENGTH	Length of the character string <String Parameter 1>.	This variable is included in the PTCSTRFN.CBL copy library.
<String Parameter 1>	The character string .	
<String Parameter 2>	The buffer area to be updated by the subroutine.	

Values Returned

<i>Parameter</i>	<i>Value</i>	<i>Notes</i>
<String Parameter 2>	This buffer will be updated with the appropriate values. This field will contain at least one of these values: <ul style="list-style-type: none"> • 1 - The next character position is part of a 1-byte character. • 2X. The next 2 character positions are part of a 2-byte character. • 3XX - The next 3 character positions are part of a 3-byte character. 	Refer to the example below this table to see how the function works.
STRFN-RC	Returns one of the following values: <ul style="list-style-type: none"> • STRFN-RC-OK • STRFN-INVALID-ACTION 	This variable is included in the PTCSTRFN.CBL copy library

Example

The PTPSTRFN COBOL subroutine provides an option to map the byte length of an entire character string. The sample COBOL code below provides an example of how this feature is used. The code will read the all fields from the following Unicode table:

LANGUAGE	UNITEXT
Catalan	Quan el món vol conversar, parla Unicode
Chinese (Simplified)	当世界需要沟通时, 请用Unicode !
Chinese (Traditional)	當世界需要溝通時, 請用統一碼(Unicode)
Danish	Når verden vil tale, taler den Unicode
Dutch	Als de wereld wil praten, spreekt hij Unicode
English	When the world wants to talk, it speaks Unicode
Esperanto	Kiam la mondo volas paroli, ĝi parolas Unicode
Finnish	Kun maailma haluaa puhua, se puhuu Unicodea
French	Quand le monde veut communiquer, il parle en Unicode

Sample Unicode table

For each row shown in the table above, the code performs the following functions:

- Calls the PTPSTRFN to get the string mapping of the UTF-8 character string of the text retrieved for the UNITEXT field.
- Displays the UTF-8 string equivalent of the text.
- Tallies the number of 1-byte, 2-byte and 3-byte characters of the text.

The COBOL code analyzes the text as follows:

```

01  W-WORK.

      02  LANGUAGE                      PIC X(20) .

      02  UNICODE-TEXT                  PIC X(300) .

      02  UNICODE-TEXT-MAP              PIC X(300) .

      02  DATA-LEN                      PIC 9(3)    COMP .

      02  BYTE-POS-MAX                   PIC 9(4)    COMP .

      02  COUNTERS.

          05  COUNT-1BYTE-CHAR          PIC 9(02)   VALUE ZEROS .

          05  COUNT-2BYTE-CHAR          PIC 9(02)   VALUE ZEROS .

          05  COUNT-3BYTE-CHAR          PIC 9(02)   VALUE ZEROS .

01  BYTE-ARRAY.

      02  BYTE-POS                      PIC X      OCCURS 300   TIMES

                                     INDEXED BY BYTE-IDX.

          88  ONE-BYTE-CHAR              VALUE '1' .

          88  TWO-BYTES-CHAR             VALUE '2' .

          88  THREE-BYTES-CHAR           VALUE '3' .

          88  BYTE-STRING-END            VALUE SPACE.

01  STR-FUNC                            COPY 'PTPSTRFN' .

```

```

. . .

. . .

    <Code to retrieve the text from the database and assign to the
appropriate fields>

. . .

. . .

* Initialize the string map before calling the function

    MOVE SPACES      TO UNICODE-TEXT-MAP

    CALL 'PTPSTRFN'   USING  ACTION-STRMAP

                                STR-FUNC

                                UNICODE-TEXT

                                UNICODE-TEXT-MAP

    IF NOT STRFN-RC-OF OF STR-FUNC

        <ABEND PROGRAM>

    END-IF

    SET BYTE-POS-MAX TO 300

    MOVE 300 TO DATA-LEN OF W-WORK

    MOVE UNICODE-TEXT-MAP TO BYTE-ARRAY

    PERFORM VARYING BYTE-IDX FROM 300 BY -1

        UNTIL BYTE-IDX <= 1

        OR NOT BYTE-STRING-END(BYTE-IDX)

        SUBTRACT 1 FROM DATA-LEN OF W-WORK

    END-PERFORM

* Initialize counters

    MOVE ZEROS      TO COUNT-1BYTE-CHAR

    MOVE ZEROS      TO COUNT-2BYTE-CHAR

    MOVE ZEROS      TO COUNT-3BYTE-CHAR

    PERFORM UNTIL BYTE-IDX > DATA-LEN OF W-WORK

        EVALUATE TRUE

```

```

        WHEN ONE-BYTE-CHAR

            ADD 1 TO COUNT-1BYTE-CHAR

        WHEN TWO-BYTES-CHAR

            ADD 1 TO COUNT-2BYTE-CHAR

        WHEN THREE-BYTES-CHAR

            ADD 1 TO COUNT-3BYTE-CHAR

    END-EVALUATE

    SET BYTE-IDX UP BY 1

END-PERFORM

DISPLAY ' LANGUAGE = ' LANGUAGE

DISPLAY '   UTF8 TEXT: (LENGTH = ' DATA-LEN ') '

DISPLAY '           ' UNICODE-TEXT

DISPLAY ' '

DISPLAY '   UTF8 BYTE MAPPING: '

DISPLAY '           ' UNICODE-TEXT-MAP

DISPLAY ' '

DISPLAY '   TALLY: '

DISPLAY '   NUMBER OF ONE-BYTE CHAR:   '

           COUNT-1BYTE-CHAR

DISPLAY '   NUMBER OF TWO-BYTES CHAR:   '

           COUNT-2BYTE-CHAR

DISPLAY '   NUMBER OF THREE-BYTES CHAR: '

           COUNT-3BYTE-CHAR

DISPLAY ' '

DISPLAY ' '

```

The output of this program appears below. Note that certain strings appear to be garbled. This is because there is no font to display the characters properly.

```

LANGUAGE = Catalan

UTF8 TEXT: (LENGTH = 0041)

```


CHAPTER 15

Global Reporting and Analysis

This chapter covers the globalization features of PeopleSoft reporting and analysis tools, as well as third-party reporting and analysis tools that are used in conjunction with PeopleTools.

Using the Strings Table

The PeopleTools Strings table (the record name for which is STRINGS_TBL) stores textual strings, which are used for enabling language-sensitive labels and other text in PS/nVision and SQR reports.

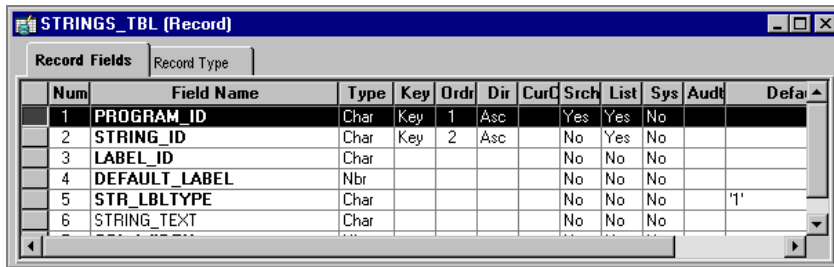
This functionality helps you avoid hard coding labels in the report file itself. The use of strings rather than hard-coded text in reports enables translators to translate the report layout in the database without having to edit the report's code itself. This enables you to run a single copy of a report in multiple languages, while avoiding the duplication of code.

Each row in the Strings table keys each string to a STRING_ID, which is associated with one of two different types of string:

- The short (RFT Short) or long (RFT Long) field description of the field. In this case the STRING_ID must equal the field name.
- A free-form text string.

In either case, when a report requests a string in a particular language, the system returns the translated string if a translation is available. If no translation exists, the base language string is returned.

The Strings table is also keyed by a PROGRAM_ID field, which enables you to classify strings into groups that are used in similar reports. PROGRAM_ID can refer to a specific SQR or PS/nVision report name, or it can be a mnemonic for a group of common strings that are shared between reports.



Num	Field Name	Type	Key	Ordr	Dir	CurC	Srch	List	Sys	Audt	Defa
1	PROGRAM_ID	Char	Key	1	Asc		Yes	Yes	No		
2	STRING_ID	Char	Key	2	Asc		No	Yes	No		
3	LABEL_ID	Char					No	No	No		
4	DEFAULT_LABEL	Nbr					No	No	No		
5	STR_LBLTYPE	Char					No	No	No		'1'
6	STRING_TEXT	Char					No	No	No		

STRINGS_TBL record definition

The following topics describe how the system uses the Strings table to perform language-sensitive evaluations of PS/nVision string variables and how to maintain the Strings table, using the PeopleTools Strings Table utility.



For more information about using the Strings table in SQR reports, see SQR Report Translation.

Evaluation of PS/nVision String Variables

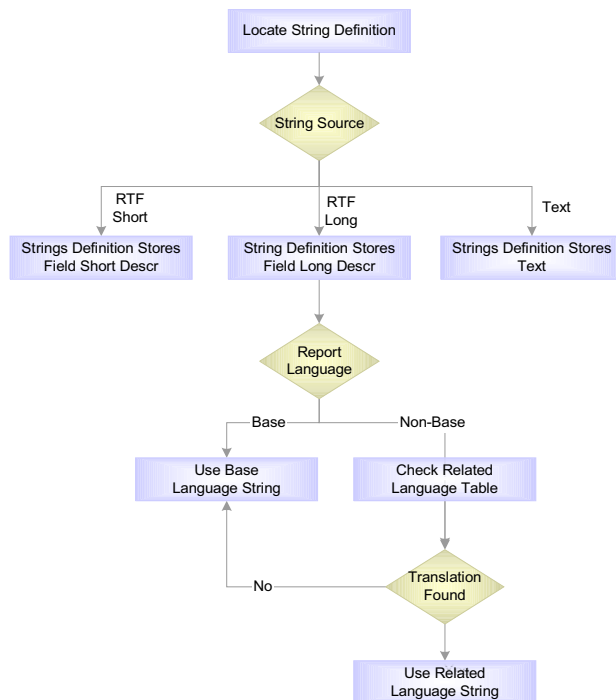
The system uses the string definitions in the Strings table to evaluate PS/nVision variables, based on three factors:

- A Program ID, which identifies a set of strings used in similar reports.
- A String ID, which uniquely identifies the string definition within a Program ID group.
- The language of the report request. If no language is specified, the system assumes the preferred language of the current user.

The syntax of a PS/nVision string variable includes the String ID and the Program ID:

```
%.StringID,ProgramID%
```

The following flowchart shows how the system evaluates language-sensitive string variables:



Evaluation of SQR and PS/nVision string variables

The evaluation follows this sequence:

1. The system locates a string definition in the Strings table, using values for the Program ID and String ID.

The text string stored in the Strings table (or its related language table) can be a short or long field description (validated against the PSDBFIELD or PSDBFIELDLANG table) or a free-form text string that is not associated with a field.
2. The system determines the language of the report request. If the value is unspecified, the system uses the user's current language preference setting.
3. If the language is the system's base language, the system retrieves the string values from the base-language Strings table (STRINGS_TBL).
4. If the system is a non-base language, the system looks for a string with the correct language code in the related-language table (STRINGS_LNG_TBL).
5. If no translation exists in the related-language table, the system uses the string in the base-language table (this is the default).

Using the Strings Table

The Strings Table page enables you to:

- Create groups, or categories, of language-sensitive strings that are used in similar reports.

- Add or modify string definitions in an existing Program ID group.
- Translate the text strings that are stored in the Strings table.

To define a string group:

1. Select PeopleTools, Utilities, Use, String Table.
2. Search for an existing Program ID or add a new one.

Use the standard search or add method to enter your Program ID and access the Strings Table page.

Program IDs enable you to create sets of strings that are used in similar reports. For example, you might want to create one Program ID group for human resources reports, another for quarterly financial reports, and so forth.



To create a generic Program ID group for PS/nVision reports, create a Program ID called *NVISION*. PS/nVision string variables will use this default Program ID if the nVision Only String check box is selected in the nVision String Criteria dialog box.



For more information, see Evaluation of PS/nVision String Variables.

The Strings Table page appears:

Strings Table

Program ID: STDHGDGTR Report Language: English

String List Find | View All First 1-6 of 6 Last

*String Source	*String ID	Default Label	String Text	Width
Text	STDHGDG_CO_NM		PeopleSoft	10
Text	STDHGDG_END_REP		End of Report	13
Text	STDHGDG_PAGE_NO		Page No.	8
Text	STDHGDG_REP_ID		Report ID:	10
Text	STDHGDG_RUN_DT		Run Date	8
Text	STDHGDG_RUN_TM		Run Time	8

Save Return to Search Add Update/Display

Strings Table page

At the top of the page, you see the **Program ID** and the **Report Language** (the user's preferred language set in the User Profile).



The **Report Language** setting shows the user's preferred language, as set in the user's profile in PeopleSoft Security. The value in this field does not change to reflect the user's signon language.

3. For each string definition in the page, select a string source.

In the **String Source** field, select one of the following values:

Value	Description
Text	Use Text string defined in the String table.
RFT Short	Use the Short description of a field that is associated with the string definition.
RFT Long	Use the Long description of a field that is associated with the string definition.

4. Assign a String ID.

The **String ID** field can contain:

- A field name (if **String Source** is set to *RFT Long* or *RFT Short*).

If the **String Source** is *RFT Long* or *RFT Short*, the **String ID** field must contain a valid field name. T



The system will not force you to enter a valid field name in the **String ID** field (because the field might contain a text string ID); so you should use the **Search** button to prompt for a valid field name.

Remember that fields can have multiple labels. Select the **Default Label** check box to ensure that the default label is used.



If you do not use the field's default label, you must select which of the field's labels to use, using the **Label Properties** button located to the right of the **Default Label** check box.

- An ID string that is unique to the Program ID group (if **String Source** is set to *Text*).

To assign a *Text* string ID, type the ID in the **String ID** field.

5. Specify string text and width


If **String Source** is *RFT Short* or *RFT Long*, and if **String ID** contains a valid field name, the RFT Short or RFT Long string description will be copied into the Text field. This display is for reference only, as you can only modify field labels using the Field Designer.

If **String Source** is *Text*, you must type the string text into the **Text String** field.

In either case, the number of characters in the string is automatically placed in the **Width** field. This width is used to tell the translator the maximum number of column positions on the report is available for their translation. For example, if you are using a string as the heading of a column in a tabular report, and the column print width is 30 characters, you should specify a string width of 30 to inform the translator that translations longer than 30 characters will exceed the available space on the report.

Because the Strings Table page cannot open your report definition, the **Width** defaults to the current width of the string you entered or selected. Be sure to update the width based on the actual space available on your report layout to avoid limiting a translator to an artificially short length, which is likely to degrade the quality of their translation.

6. Repeat steps 3 through 5 for each string definition.

Use the  Add button to insert additional string definitions.

7. Save the page.

To translate string definitions into a non-base language:

1. Log into the database as a user of the target language (the language into which you want to translate the string definitions).

To view or translate string definitions on the Strings Table page, you must log in to the database with the preferred language in your User Profile set to the target language. You can modify your User Profile preferences on the Maintain Security pages.



Unlike other PeopleSoft pages, the Strings Table page does not use the current signon language of the operator. The User Profile language preference, not the signon language, determines the report language.

2. Select PeopleTools, Utilities, Use, String Table.
3. Search for an existing Program ID.

Use the standard search method to enter your Program ID and access the Strings Table page.

The Strings Table page appears, displaying a **Report Language** that is the same as the Preferred Language defined on the Maintain Security pages for the current user.

Strings Table

Program ID: STDHDGTR Report Language: Spanish

String List Find | View All First 1-6 of 6 Last

*String Source	*String ID	Default Label	String Text	Width
Text	STDHDG_CO_NM		PeopleSoft	10
Text	STDHDG_END_REP		End of Report	13
Text	STDHDG_PAGE_NO		Page No.	8
Text	STDHDG_REP_ID		Report ID:	10
Text	STDHDG_RUN_DT		Run Date	8
Text	STDHDG_RUN_TM		Run Time	8

Strings Table Page set for the Spanish report language

Notice that the **String Text** values for strings with **String Source** set to **RFT Short** or **RFT Long** are read-only. These values are copied from field description values stored in the field definition. If these values have not been translated into the target language, you will need to exit the String Table page, translate the description field in the field definition, and then return to the Strings Table utility to view the translated string.

4. Translate editable String Text values into the target language.

Overwrite the base-language text strings with target language strings.

5. Save the page.

PS/nVision Layout Translation

PS/nVision has several features that take into account the language of the user who is designing or requesting a report, the language of the user who makes a report request, and the language of the ultimate audience of a report (the person or group of people for whom the report was generated). The primary goal of these features is to allow a single layout to produce instances of any requested language or group of languages. To that end, the features are designed to present reports to end users in their preferred languages. Production reports can be produced by users other than those who read the reports. Thus, there is a distinction between the *designer's* language and the *user's* language.

The globalization features of PS/nVision can be broken into two distinct parts:

- **Design-time features.** Features that are related to designing PS/nVision layouts.
- **Runtime features.** Features that are activated when a user makes a PS/nVision report request.



For more information, see PeopleSoft nVision.

Design-Time Features

The functions of the design-time features of PS/nVision are to adapt to the current user's language and to build a layout that can produce reports in multiple languages.

Column Headings

When the user selects a column from a list, the list is presented in the user's language (unless database field names are requested). Wherever the heading is stored in the layout, it's stored in the base language.

Tree Names

When prompting for tree names or presenting a tree description—for example, when adding tree criteria for a field—the tree description is displayed in the user's language. This may require adding related language records to one or more prompting views, based on the structure of your tree.

String Variables

Layouts typically contain a considerable amount of constant text, such as the column heading "Last Year to Date" or "Current Budget." PS/nVision provides the option to build multilingual layouts where these text strings are replaced by specially formatted strings whose user-language equivalent can be retrieved from the PeopleTools Strings table. This enables a single PS/nVision layout definition to be used in multiple languages without duplicating the layout itself. These string names resemble user-defined PS/nVision variables.

PS/nVision translates only the strings that occupy a layout cell and are in the following format:

`% .name , program%`

where *name* is the string name and *program* is the optional group that is used to collect a common set of strings that are used on similar reports. You can look up the *name* in the Strings table, using the Strings Table utility.



This is similar to the approach used by PeopleSoft application SQRs. Names are case sensitive.



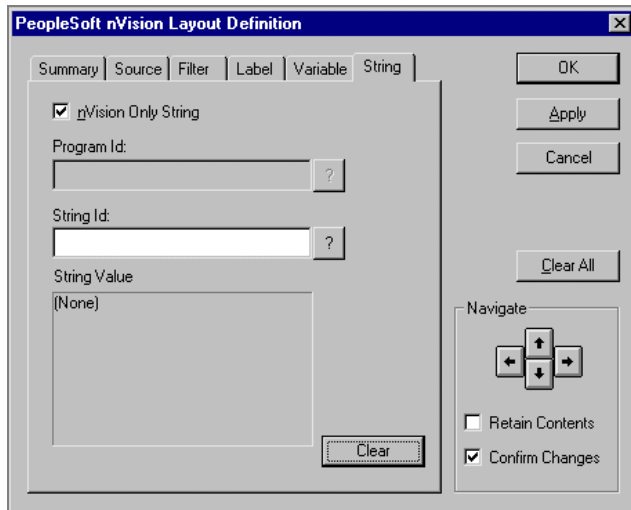
For more information about the Strings table, see Using the Strings Table.

To set up string criteria for the current cell selection:

1. Open the PeopleSoft nVision Layout Definition dialog box.

2. Select the String tab.

The String tab in the PeopleSoft nVision Layout Definition dialog box appears.



PeopleSoft nVision Layout Definition dialog box - String tab

You use this tab to insert a string from the Strings table into a cell in your layout.

3. Clear the **nVision Only String** check box, if appropriate.

By default, this tab displays only the strings that were created for use with PS/nVision—those with a program ID of NVISION. If you want to be able to select from all available strings, clear the **nVision Only String** check box.

4. If nVision Only String is cleared, select a Program ID.

5. In the **String ID** field, select the ID of the string you want to insert.

You can select any of the strings assigned to the Program ID that you specified.



If **nVision Only String** is selected, the Program ID is NVISION.

6. Click **Apply** to save your changes and define string criteria for a different cell, or click **OK** to save your changes and close the dialog box.

If you clicked **Apply**, and you want to reuse all or part of the criteria you just applied, select the **Retain Contents** check box. This will preserve all the dialog box information when you navigate to a new cell selection. Then repeat steps 3 through 5 to define additional string criteria.

PS/nVision Runtime Globalization Features

The following features are activated when a user makes a PS/nVision report request.

Path Search

As PS/nVision looks for a file (such as an Excel spreadsheet, layout, or template), it steps through the directories listed in the appropriate path (such as the Layout path, defined in the Configuration Manager). Before looking in each directory, PS/nVision looks in a subdirectory named by the user's language code; if the file is not there, or if the appropriate directory does not exist, PS/nVision looks in the directory named in the path; if the file is not there, PS/nVision moves to the next directory in the path and repeats the process.

This feature supports users who need or prefer different layouts and for users of different languages. For example, it may be necessary to create an earnings report differently in Germany and France because differences in accounting rules or management requirements dictate different criteria and/or formatting. So, if a French user runs a PS/nVision report, PeopleTools first looks in the FRA directory under each PS/nVision directory that is defined for an installation. If the required report isn't found in the FRA directory, PeopleTools will fall back and use the generic report in the base PS/nVision directory.

Request Language

The PS/nVision Report Request component includes a Language Template text box on the Advanced Options page. You can use the Foreign Language Translation field to provide the language code for the language in which the instances produced by the request are to be generated. If you are applying a scope to the report request, you can enter a string containing one or more PS/nVision variables.

The screenshot shows the 'nVision Report Request' window with the 'Advanced Options' tab selected. At the top, 'Business Unit' is 'US001' and 'Report ID' is 'BALANCE'. Below this is a section titled 'Advanced Report Instance Options' with three checkboxes: 'Enable nPlosion If Specified In Layout' (unchecked), 'Translate Summary Ledgers to Detail' (unchecked), and 'Data From Requesting B.U. Only' (checked). Below that is a section titled 'Foreign Language Translation' containing a text input field and a description: 'Enter an alternate language code for automatic translation. Available if alternate language features are in the layout.' At the bottom of the form is a link: 'Enter Delivery Template Options - View All'. The footer of the window contains several buttons: 'Save', 'Return to Search', 'Previous tab', 'Next tab', 'Add', and 'Update/Display'. Below the window, the text 'nVision Report Request | Advanced Options' is visible.

Specifying a language template on the Advanced Options page

If you enter one or more variables in the **Foreign Language Translation** field, then the value of each variable is interpreted at runtime to get the language code for each report instance. This enables retrieval of the language code from a tree node or value table that is associated with the values in the scope.

The syntax of this variable is as follows:

```
%DES.[scopefield].[detailfield|.nodefield|detailfield.nodefield]%
```

Note that the field names required vary, depending on the type of scope you're using, but the periods between the values must always exist—except in the case of a trailing period. The *scopefield* parameter is optional. If you don't specify one, PS/nVision will use the first scope field you defined. If you want to use a field other than the first one defined in the scope, then you must specify which one.

You can provide a *detailfield* or a *nodefield* or both. PS/nVision uses the appropriate field type, based on your scope. Consequently, specifying one of each type enables you to change your scope definition without necessarily changing this variable. The detail and node table fields you specify should store PeopleSoft language codes and be found on the same node or detail value table associated with the *scopefield*. Typically, the field name used is LANGUAGE_CD.

For example, a common implementation of this variable is as follows. (Note the inclusion of the extra period):

```
%DES..LANGUAGE_CD.LANGUAGE_CD%
```

If your scope consists of multiple scope fields (PRODUCT and BUSINESS_UNIT), your variable might look like this:

```
%DES.BUSINESS_UNIT.LANGUAGE_CD.LANGUAGE_CD%
```

If you don't include a *scopefield* or *detailfield* value, it's important that you still include the extra periods that follow those values. For example:

```
%DES...nodefield%
```

```
%DES.scopefield..nodefield%
```

However, if you don't include a *nodefield*, there's no need to include the trailing period after the *detailfield*. For example:

```
%DES..detailfield%
```

Using a *scopefield* to drive the language of the PS/nVision report enables you to run a single report in multiple languages. Each time the contents of the *scopefield* changes, PS/nVision resets the language of the report and reloads the strings and other language-sensitive objects in the new language. This enables you to create a PS/nVision report in multiple languages so that the report can be separated and delivered to multiple recipients, each of whom may have a different language preference.



For more information, see *Creating a Report Request and Defining Report Scopes*.

Labels

When retrieving node or detail row and column labels, PS/nVision uses the appropriate alternate language record, if one has been defined through PeopleSoft Application Designer, to get labels in the user's language.

PS/nVision Variables

When retrieving a descriptive value for a PS/nVision variable (such as Business Unit Description), PS/nVision determines whether the table being queried has a related language record and if the field being retrieved is on that record. If so, and if the user does not use the base language, PS/nVision retrieves the value from the related language record or, if the row doesn't exist on the related language record, from the base record.

International Versions of Microsoft Excel

Microsoft Excel is available in numerous languages. PeopleTools is designed to work with all international editions of Excel.



For more information, see the PeopleSoft Hardware and Software Requirements guide.

SQR Globalization

PeopleTools enables SQR labels to be maintained in multiple languages. The following topics cover some of the internationalization features of SQR.

SQR Printing for A4 Paper

PeopleTools supports printing to paper sizes such as A4. Changing the PAPER_SIZE setting in SETENV.SQC changes the paper size from Letter to A4 for all reports that share the same file server directory.

Currency Precision

NUMBER.SQC enables you to make use of currency precision, both in terms of character string values—in other words, with an edit mask—and numeric values, rounded to a specified precision. Use these functions to achieve your currency precision requirements:

- Format_Currency_Amt
- Format_Currency_Amt_Numeric

Documentation of these functions can be found in the comments block at the top of the NUMBER.SQC SQR include file.

Date and Time Formatting

The DATETIME.SQC program provides several procedures to aid in formatting of date and time values for display in report output. While DATETIME.SQC provides generic procedures for formatting dates and times, the actual format used for date and time values can vary in each report. By default, when printing dates and times, reports use the system-wide default date and time formats that are specified in SETENV.SQC.

During your PeopleSoft installation, you should edit SETENV.SQC to specify the system-wide default format you prefer for date and time values. Edit the following lines in your SETENV.SQC to tell SQR which format you prefer, if it is not otherwise specified in the SQR report source.

```
#define Year4      '1'           !0 = 2 digit year

#define Prompt-Date mm/dd/yyyy

#define Prompt-Mask 'MDY'

#define DateType   '0'           !iDate 0 = mdy, 1 = dmy 2 = ymd

#define TimeDisplay '1'           !iTime 0 = 12hr, 1 = 24hr
```

Based on these settings or any overrides, procedures in DATETIME.SQC provide support for various date formats. The table below identifies these formats and provides examples of the date December 14, 2000 in each.

SQR Date Format	Description	Example
{DEFDMY}	DD/MM/YYYY	14/12/2000
{DEFMDY}	MM/DD/YYYY	12/14/2000
{DEFKAN}	Japanese Kanji-format dates (using the Japanese Imperial calendar)	平成12年12月14日
{DEFROM}	Japanese Romaji-format dates (using the Japanese Imperial calendar)	H.12.12.14

For details on how to include date/time formatting in your own SQR reports, refer to the documentation inside the comment block of the DATETIME.SQC SQR include file on your file server. If you plan to use Japanese date formatting in your SQR report, remember to include *#define JapaneseDates* at the top of your SQR report.

SQR Report Translation

PeopleTools enables you to print or format SQR report output for multiple languages, using string definitions that are stored in the Strings table. The procedures defined in SQRTRANS.SQC enable your SQR program to access these strings dynamically.



For more information about updating and maintaining the Strings table, see Using the Strings Table.

To enable the use of the Strings table in your SQR, you need to include the PeopleTools SQC file SQRTRANS.SQC, which includes the routines that are necessary to initialize and load translated strings from the Strings table.

SQRTRANS.SQC has four main functions that can be called from your report:

- Init_Report_Translation
- Get_Field_Information
- Append_Report_Translation
- Add_Report_Translation

Init_Report_Translation

Call the Init_Report_Translation procedure from your SQR before using any of the string table information. Typically you should call Init_Report_Translation in the Init-Report section of your SQR. Init_Report_Translation takes two parameters:

Parameter Name	Description
\$Report_ID	\$Report_ID is normally the name of your SQR. This parameter is used as the Program ID when looking up strings in the Strings table.
\$Report_Language	\$Report_Language indicates the preferred language for the strings that are being retrieved. Init_Report_Translation will attempt to load all strings in the language specified; however, if a translation for any string does not exist, it will load the base language description for that string.

If you want to change languages during the processing of an SQR report (for example, if you want each page of the report to be in a different language), you can call Init_Report_Translation multiple times within a single SQR, each time passing a new \$Report_Language value.

Get_Field_Information

Call the Get_Field_Information procedure for each string that you want to retrieve from the Strings table. It retrieves the label or string table entry for the field specified, and places it in a

report variable. You can then print the contents of this variable on your report as a label, column heading, or free text. `Get_Field_Information` takes four parameters:

Parameter Name	Description
<code>\$Report_ID</code>	<code>\$Report_ID</code> is normally the name of your SQR. This parameter is used as the Program ID when looking up strings in the Strings table. You must have already called <code>Init_Report_Translation</code> specifying this <code>\$Report_ID</code> before passing it to <code>Get_Field_Information</code> .
<code>\$Field_ID</code>	This is the String ID of the string whose text you want to retrieve. It must exist as an entry in the Strings table under the <code>\$Report_ID</code> you specified.
<code>\$Field_Text</code>	<code>\$Field_Text</code> is the output variable. <code>Get_Field_Information</code> populates this variable with the text that corresponds to the <code>\$Report_ID</code> and <code>\$Field_ID</code> that are specified in the preferred language or in the database's base language (if a translation doesn't exist in the preferred language).
<code>\$Field_Width</code>	<code>\$Field_Width</code> is an output variable that <code>Get_Field_Information</code> populates with the width of the text string that is returned.

Append_Report_Translation

If your SQR program uses strings from more than one Strings table Program ID, call `Append_Report_Translation` to add the strings from another Program ID to the initialization array created by `Init_Report_Translation`. This function is particularly useful if you have a set of strings that are used across many of your SQR programs. You can group these strings under a generic Program ID and use them in multiple SQRs.

`Append_Report_Translation` takes a single argument: `$Report_ID`. It assumes the same language that was specified in `Init_Report_Translation`. It must be called after `Init_Report_Translation`.

Add_Report_Translation

The `Add_Report_Translation` procedure calls `Init_Report_Translation` or `Append_Report_Translation`, depending on whether the Strings table has been initialized. It takes the same arguments as `Init_Report_Translation`. If `Init_Report_Translation` has not yet been called during the processing of this SQR, this function will call it, passing both parameters. If `Init_Report_Translation` has already been called, `Add_Report_Translation` calls `Append_Report_Translation`, passing only the `$Report_ID` parameter.

This function is very useful in your own SQC files if you cannot be certain that the calling SQR program has already initialized the strings table. The function ensures that the table will be initialized or appended to correctly.

Sample Strings Table Enabled SQR

The following sample code demonstrates how to use the Strings table to retrieve string values in SQR, using the procedures described in the preceding sections.

```

!*****

!  SAMP001:  Report on database's base language *

!*****

#include 'setenv.sqc'      !Set environment

begin-report

    do Init-Report

    do Process-Main

    do Reset

end-report

begin-heading 6

do Get_Field_Information ('SAMP001',      'REPORT_TITLE', $REPORTTITLE,      #DWRT)

do Get_Field_Information ('SAMP001',      'EXPLAIN_TEXT', $EXPTTEXT,          #DWET)

PRINT $REPORTTITLE          (1)      CENTER

PRINT $EXPTTEXT              (+2,1)

end-heading

begin-procedure Init-Report

    move 'SAMP001' to $ReportID

    move 'ENG' to $Language_cd

    do Init_Report_Translation ($ReportID, $Language_cd)

    do Append_Report_Translation ('GEN')

end-procedure

begin-procedure Process-Main

do Get_Field_Information ('GEN',          'BASELANGUAGE', $BASELANGUAGE,      #DWBL)

do Get_Field_Information ('GEN',          'ENDOFREPORT',  $ENDOFREPORT,      #DWER)

begin-SELECT

LANGUAGE_CD &Base_Language

    let $langlabel = $BASELANGUAGE || ':'

    print $langlabel      (+1,1)

    let #fieldpos = #DWBL + 3

```



```

        print &Base_Language (0,#fieldpos)

FROM PSOPTIONS

end-SELECT

print $ENDOFREPORT      (+4,1)

end-procedure

#include 'reset.sqc'      !Reset printer procedure

#include 'sqrtrans.sqc'  !SQR Strings Table procedures

```

Required Settings for SQR and Unicode

SQR 6.1 uses Unicode for internal storage of characters. This enables a single instance of SQR to process data in virtually any language simultaneously. It does, however, introduce some special considerations for SQR reports that are designed to operate on Unicode data.

The PSSQR.INI file in the %PS_HOME%\SQR directory provides settings that control the behavior of SQR, including the determination of the character set that is used for report output, communication with the database, and file I/O. The settings in your PSSQR.INI take effect across all the PeopleSoft installations that share your batch server environment.

The following settings under the [Default-Settings] section in PSSQR.INI must be set as follows, as they enable SQR features required by PeopleSoft:

```

UseUnicodeInternal=TRUE

AutoDetectUnicodeFiles=TRUE

```

UseUnicodeInternal

The UseUnicodeInternal parameter tells SQR to use Unicode for all internal processing, regardless of the character set of the database or operating system it is running against. Setting this parameter enables SQR to support a large list of character sets and provides support for language-sensitive functions, such as substrp(), which are documented later in this section.

AutoDetectUnicodeFiles

The AutoDetectUnicodeFiles parameter tells SQR to check for a Unicode Byte-Order-Mark (BOM) at the top of each opened file and, if the Unicode BOM is found, to treat the file as a Unicode UCS-2 encoded file, regardless of the encoding specified in the SQR.INI or OPEN command. The BOM, (represented in hexadecimal as 0xFFEF or 0xFEFF, depending on your hardware) indicates that a file is encoded in Unicode and tells SQR which hardware byte ordering was used to write the file.

The following settings under the [Environment:Common] section in PSSQR.INI are the default as shipped by PeopleSoft. You should ensure that they are set correctly for your environment.

The valid values for these parameters are any valid PeopleSoft encoding.



For more information and a full list of the PeopleSoft-supported character sets, see Character Sets and Language Input/Output.

```
ENCODING=LATIN1
```

```
ENCODING-SQR-SOURCE=LATIN1
```

```
ENCODING-DATABASE=LATIN1
```

```
ENCODING-FILE-OUTPUT=LATIN1
```

```
ENCODING-FILE-INPUT=LATIN1
```

```
ENCODING-REPORT-OUTPUT=LATIN1
```

```
ENCODING-CONSOLE=CP437
```

ENCODING

The ENCODING parameter sets the default encoding for all the SQR encoding variables, if they are not set.

If an individual encoding parameter is explicitly included in PSSQR.INI, this specific setting overrides the default encoding specified by the ENCODING parameter.

ENCODING-SQR-SOURCE

The ENCODING-SQR-SOURCE parameter specifies the encoding in which the SQR source (*.SQR, *.SQC) files are encoded.

As delivered by PeopleSoft, all SQR source is encoded in Latin-1 on the Windows NT and UNIX platforms and in EBCDIC (CCSID37) on OS/390.

If you plan to include non-Latin1 characters directly in your SQR source, as string constants, then you need to change this setting to match the encoding used for your SQR source. However, if the SQR.INI parameter AutoDetectUnicodeFiles is TRUE, and SQR detects a Unicode Byte-Order-Mark (BOM) at the top of the SQR source file, it will always be read as a Unicode file, regardless of this setting. This is useful if only a few of your SQR source files need to include non-Latin characters.

The BOM is typically written to the top of any file that is encoded in Unicode UCS-2 by the editor, including Windows Notepad.

ENCODING-DATABASE

The ENCODING-DATABASE parameter determines the encoding that SQR uses to communicate with your database.

This should match the encoding in which you want SQR to communicate with your database's application programming interface (API) on the machine on which SQR is running. It should be set to the following values, based on your database platform and database encoding:

Database Platform	Database Character Set	ENCODING-DATABASE Setting
Microsoft SQL Server	Unicode	UCS-2
	ISO-1 (CP1252)	CP1252 or LATIN1
	CP932 (Japanese)	CP932 or Shift_JIS
Oracle	Unicode	UTF-8
	WE8ISO8859P1	LATIN1
	WE8ISO8859P15	LATIN9
	WE8MSWIN1252	CP1252
	JA16SJIS	SHIFT_JIS
All other NT and UNIX Platforms	All	LATIN1
DB2 for OS/390	CCSID 37 and 500	EBCDIC
	CCSID 930 (Japanese/Kana)	CCSID930
	CCSID 939 (Japanese/Latin)	CCSID939

ENCODING-FILE-OUTPUT and ENCODING-FILE-INPUT

The ENCODING-FILE-OUTPUT and ENCODING-FILE-INPUT parameters control the character set that is used to read and write files using the SQR OPEN command.

If no character set is specified explicitly in the OPEN command, the character set specified in these parameters is used to read or write the file.

ENCODING-REPORT-OUTPUT

The ENCODING-REPORT-OUTPUT parameter determines the encoding for the report output file (.LIS, .PDF, .HTM, and so on).

The parameter should be set to a character set that contains all the characters you expect to print using SQR. If you are using a Unicode database, and your printer or report browser supports Unicode, we recommend that you set the character set to either UTF8 or UCS2, to ensure that all characters can be represented correctly in the output. However, you must ensure that your output devices support this encoding. Both PDF and HTML output fully support Unicode data. HP PCL format does not support Unicode.

ENCODING-CONSOLE

The ENCODING-CONSOLE parameter determines the character set that is used to write progress and other messages to the console during a SQR run.

In Windows, the encoding used by DOS consoles, such as the one that is used by SQR, is known as an *OEM* encoding; this is often different from the character set that is used by Windows itself. In Western European windows editions, CP437 should be sufficient to display all Latin-1 characters. In Japanese Windows, this should be set to CP932. See <http://www.microsoft.com/globaldev> for a list of Microsoft Windows OEM codepages. For UNIX systems, this setting should match the character set that is supported by your terminal device, such as LATIN1.

Additional Options for SQR and Unicode

String Length

When you work with strings in SQR, you have to consider three different ways of measuring string length:

- The number of characters in the string.
- The number of print columns occupied by these characters. For example, characters from the Latin alphabet normally require one print column; characters in Japanese often require two.
- The number of bytes used to store the character. SQR uses the Unicode UCS-2 method of encoding characters, which means that every character occupies 2 bytes; however, your database may use a different encoding that requires a different number of bytes for each character.

SQR provides several functions to help you manage string length according to each of these criteria.



Although SQR uses Unicode internally, it can still communicate with ANSI databases and files.



For more information about character sets and Unicode, see Character Sets and Language Input/Output.

Length() and Substr()

Length() and Substr() deal with the number of characters in a string. As the PeopleSoft field lengths as defined in PeopleSoft Application Designer are character-based, the Length() and Substr() functions are useful for calculating the length of the string as it is stored in the database.

For example, the following code determines whether string &abc will fit into a database field that is 10 characters long. If the string will not fit into the field, the code truncates the string to use only the first 10 characters.

```
If length(&abc) > 10 then
```

```
&abc = substr(&abc,1,10)

End-if
```

Lengthp() and Substrp()

Lengthp() and Substrp() deal with the number of print columns required to print the character using a monospace (nonproportional) font.

For example, the following code determines whether string &abc will fit into a print area that is 10 columns wide. If the string will not fit into the print area, the code truncates the string to use only the first 10 columns' worth of characters.

```
If lengthp(&abc) > 10 then

&abc = substrp(&abc,1,10)

End-if
```

Lengtht() and Substrt()

Lengtht() and Substrt() deal with the number of bytes that the string occupies in a specified character set. Typically, you would use lengtht() and substrt() if you are writing to a file in a specific character set, and you need to check or limit the byte length of the string in the output file, as would be required by most interface files.

For example, the following code determines whether string &abc will require more than 10 bytes in an output file. If the string is more than 10 bytes, the code truncates the string, to use only the first 10 bytes' worth of characters. The SQR system variable \$sqr-encoding-file-output is being used to reference the SQR.INI ENCODING-FILE-OUTPUT variable, which determines the default character set of any file that is written to by the SQR OPEN command. You can substitute any valid PeopleSoft encoding for the \$sqr-encoding-file-output variable.



For more information and a list of valid PeopleSoft encodings, see Character Sets and Language Input/Output.

```
If lengtht(&abc, $sqr-encoding-file-output) >10 then

&abc = substrt(&abc, $sqr-encoding-file-output,1,10)

End-if
```

OPEN command

The SQR OPEN command, used to read and write files from within SQR reports, enables the report designer to specify the character set of the file being opened. You can specify a character set explicitly in the OPEN command. If you do not specify a character set, SQR will use the character set specified in the SQR.INI parameter ENCODING-FILE-OUTPUT or ENCODING-FILE-INPUT, depending on whether you are opening the file for reading or writing.

For SQR reports where a file is being opened and written to in order to interface to a third-party system, it is often useful to be able to specify the character set in which the target system will expect the data. For example, a mainframe-based payroll system may expect an EBCDIC format file. Specifying the character set directly in the OPEN command enables the SQR program to be independent of the SQR.INI settings and enables SQR to create the file directly in the character set expected by the target system, without the need to convert the output in a separate step.



For more information and a list of the character sets supported by PeopleSoft and SQR, see Character Sets and Language Input/Output. For details about the syntax of the SQR OPEN command, see the SQR 6.1 Reference Guide.

Language-Sensitive Queries

PeopleSoft Query supports language-sensitive query output. If you create a query on a table that has a related language record, or if your query includes such a table, PeopleTools automatically performs the language lookup on the related language record. This means that the output of the query appears in the current, preferred language, if translations exist in the related language tables.

For example, if you built a simple query on the Country table (COUNTRY_TBL), the country descriptions in your Query would appear in your preferred language, even though you queried the base language record (COUNTRY_TBL) and not the related language record, COUNTRY_TBL_LANG. Query performs the necessary join to the related language table to retrieve the translations; if they don't exist, it retrieves the descriptions in the database's base language.

You should keep in mind that the related language join is performed only when you run the query. Therefore, if you create a query as an English user, and then execute the same query as a French user, you will see the output in French, even though the query was created in English. The language preference of the user who is running the query drives the language joins.



The language lookup is applied only to the data that is *output* by the query. WHERE criteria, HAVING criteria, and ORDER BY clauses are applied only to the base-language data. For example, assuming that the base language of your database is English, if you set a WHERE criterion that restricts country *descriptions* to those that begin with *Ger*, PeopleSoft Query retrieves the *Allemagne* country description when the query is run by a French language user, because the English description of *Allemagne* is *Germany*, which begins with *Ger*.

The automatic language join features in PeopleSoft Query are also used when you access your PeopleSoft database via the PeopleSoft Open Query ODBC interface.



For more information about PeopleTools ODBC connectivity, see Open Query ODBC Driver and API.

PeopleSoft Cube Manager

In PeopleSoft Cube Manager, you can build both Cognos PowerPlay cubes and Hyperion Essbase cubes that display translated data. The cubes themselves do not have to be translated; they work by accessing the translations (stored in related language tables) of language-sensitive query output and tree nodes that correspond to the cube's members.



For information about translating application data and tree nodes, see Working With Language-Sensitive Application Data and Translating Tree Descriptions and Tree Node Labels.

Cognos PowerPlay Local-Language Cubes

Cognos PowerPlay cubes are not language-sensitive. You have to build a separate cube in each language that your implementation supports; users must then access the language version of the cube that matches their language preference.

When you build a PowerPlay cube instance, you can, optionally, override the default language in which the cube is built. If you do not override the language, PeopleSoft Cube Manager builds the PowerPlay cube using the database's base language.

Each PowerPlay cube instance is built using a single language. This language can be the database's base language or any non-base language that your implementation supports. If the cube is built in the base language, then all language-sensitive elements are displayed in the base language. If the cube is built in a non-base language, then the cube's language-sensitive elements are displayed in that language, using translations of language-sensitive query output and tree nodes. If the language-sensitive fields for the query output and tree nodes have not been translated into the cube's language, the cube displays the base-language descriptions.

To override the language of a Cognos PowerPlay cube:

1. Select PeopleTools, Cube Manager, Build, Cube Instances.

Use the standard search method to access the Cube Inst Def (cube instance definition) page.

Cube Inst Def **Aggregates**

Cube Instance ID: EMPLOYMENT

Description: Department Headcount

Long Description: Headcount of different job codes and employment status for different department.

Platform: Cognos PowerPlay [Platform Options](#)

Based On: ☒ Cube Definition ☐ Cube Instance
EMPLOYMENT

Included Components First 1-4 of 4 Last

Dimension	Filter	*Dimension ID	Role	Dim Type	Status	*Sparsity		
4		HEADCOUNT	Measure		Mapped	Default	+	-
1		DEPARTMENT	Dimension	Standard	Mapped	Sparse	+	-
2		JOBCODE	Dimension	Standard	Mapped	Sparse	+	-
3		EMPL_STATUS	Dimension	Standard	Mapped	Sparse	+	-

[Save](#) [Return to Search](#) [Add](#) [Update/Display](#)

Cube Inst Def page

2. Confirm that the Analysis Platform is *Cognos PowerPlay*.
3. Click Platform Options.

The PowerPlay Options page appears.

PowerPlay Options

Cube Instance ID: EMPLOYMENT

PowerCube Path/File (.MDC): C:\TEMP\EMPLOY.MDC

Model Path/File (.MDL): C:\TEMP\EMPLOY.MDL

Data Working Directory: C:\TEMP

Language Override: French ☐ Always Include

Enterprise Server Options

Enterprise Server ☐

Service Port:

Logon Password: *****

[OK](#) [Cancel](#)

Power Play Options page

4. Set the **Language Override** to the desired language.
5. Click **OK** to close the page.
6. Save the component.



For more information, see *Designing Cubes*.

Hyperion Essbase Multiple-Language Cubes

When building a Hyperion Essbase cube instance, you specify the languages that the cube supports. These can be the database base language and any other languages supported by the implementation. When users access the cube in Essbase Administrator (or in the Essbase Excel Client), they can choose to view aliases for the cube members (stored on Essbase alias tables) in any of the supported languages. The translations of the aliases are derived from language-sensitive query output and tree nodes, so the translated aliases appear only if the tree nodes have been translated and the query data is language-sensitive and has been translated. Specific elements that have not been translated appear in the database's base language.



For more information about using Essbase alias tables, see the Essbase documentation.

You specify the languages that supported by an Essbase cube when you define the Essbase cube instance. If no languages are specified, no local-language Essbase alias tables are built, and the cube member descriptions are displayed in the database's base language.

To specify the languages of an Essbase cube instance:

1. Select PeopleTools, Cube Manager, Build, Cube Instance.

Use the standard search method to access the Cube Inst Def (cube instance definition) page.

Cube Inst Def **Aggregates**

Cube Instance ID: EMPLOYMENT

Description: Department Headcount

Long Description: Headcount of different job codes and employment status for different department.

Platform: *Platform: Hyperion Essbase [Platform Options](#)

Based On: ☒ Cube Definition ☐ Cube Instance
EMPLOYMENT

Included Components First 1-4 of 4 Last

Dimension	Filter	*Dimension ID	Role	Dim Type	Status	*Sparsity
4		HEADCOUNT	Measure		Mapped	Default + -
1		DEPARTMENT	Dimension	Standard	Mapped	Default + -
2		JOBCODE	Dimension	Standard	Mapped	Default + -
3		EMPL_STATUS	Dimension	Standard	Mapped	Default + -

Save **Return to Search** **Add** **Update/Display**

Cube Inst Def page

2. Confirm that the Analysis Platform is *Hyperion Essbase*.
3. Click Platform Options.

The Essbase Advanced Options page appears.

Essbase Advanced Options

Cube Instance ID: EMPLOYMENT

Server: Rpt-Tls1 **User Name:** ADMIN

Application: OLAP_DMC **Password:** *****

Database: EMPLOY

***User Access:** Read-Only

Meta-Data Update Action: Replace All

Data Update Action: Delete Upper Level D

Rate Cube Action: No Action

Advanced Options

☒ Run Default Calc

☒ Auto Config Sparsity

***Language Code**

1	Spanish	+ -
---	---------	-----

Rate Cube Mapping

Rate Cube Name: EMPLO_MC

Query Name:

Time:

Account:

Country:

Type:

Rate:



Default Currency Name: US\$

Default Currency Category:

OK **Cancel**

Essbase Advanced Options page

4. Specify the cube's supported languages in the **Language Code** drop-down list.

To set the value of a prompt field, click in the field and then choose from the list of supported languages. To insert a new language, press the  button. To remove a language from the list of languages that are supported by the cube, press the  button.

5. Click **OK** to close the page.

6. Save the component.



For more information, see *Designing Cubes*.

Crystal Reports

PeopleTools supports multiple language editions of the Crystal Reports Designer program, as well as producing translated output from Crystal.



For more information about supported Crystal Reports languages, see the *PeopleSoft Hardware and Software Requirements* guide.

There are separate directories below the Crystal directory that correspond to the language codes that are available for your PeopleSoft implementation. Each directory contains the appropriate report in the specified language, if such a report has been defined. Based on the user's preferred language, PeopleTools tries to find the directory that corresponds to the user's language code and reads the report from there, if it exists in that location. If the system cannot find the report in the directory that corresponds to the user's language code, then it goes back to the base language directory and uses the report that it finds there.



Because Crystal Reports does not support Unicode, you are limited to a single character set in each report. For example, you can't mix French and Japanese data in the same report; instead, you must run the report once in each language on a Windows installation whose default ANSI codepage matches the character set that is required for the language. For example, Crystal reports containing Japanese characters must be run on a Japanese Windows workstation, but Crystal reports containing Spanish characters can be run on any Windows workstation using a Western European (CP 1252) codepage, including English, German, French and Spanish Windows.



For more information, see the *PeopleSoft Hardware and Software Requirements* guide.

Language Parameter for PSCRRUN.EXE

Process Scheduler uses the PSCRRUN.EXE process to execute Crystal reports. The process's Language Code parameter specifies the language directory that PSCRRUN.EXE uses to fully qualify the report path. Unlike SQR, PSCRRUN.EXE searches for a separate Crystal report file for each translation of the report. The language directory parameters are needed because each language-specific report is stored in a separate language directory. If the report is not found in the language that is specified by the parameter, PSCRRUN looks in the default, base-language directory.

Typically the language parameter is passed to PSCRRUN.EXE from the run control that is associated with your Crystal report.



For more information, see Process Scheduler Development.

Index

A

- Application Designer
 - translating with 6-2
- Application Servers
 - character sets for 13-13

B

- base language
 - choosing for an implementation 1-7, 3-1
 - defined 1-1
 - non base language pages 6-20
 - related language tables 4-1
 - swap base language audit report 3-1
 - swapping 3-3
- base time zone 10-1
 - setting 10-2
- browser language preferences 2-7
 - and date formatting 2-8
 - and number formatting 2-8

C

- CD-ROM
 - Global Multi-Language 3-12
 - ordering xv
- character sets 1-4. *See also* Unicode
 - and adding new languages 3-6
 - and file input/output 13-8
 - ASCII 13-1
 - COBOL 13-8
 - data sorting, and 13-18
 - defined 13-1
 - double-byte 13-2, 13-7, 13-18
 - for Application Servers 13-13
 - for PeopleSoft Internet Architecture 13-14
 - for Windows clients 13-15
 - functions for converting 13-22, 13-28
 - in databases 13-8, 13-12
 - in PeopleSoft 13-8, 14-1
 - in third party products 13-8
 - input methods 13-17
 - ISO standards 13-1
 - non-Unicode 3-6, 13-6, 13-8, 13-19
 - PeopleCode for 13-22
 - single-byte 13-2, 13-6
 - supported 13-9

- understanding field length 13-18
- Unicode 13-1
 - Unicode conversions 13-9
- CharType function 13-23
- Chinese characters, standard name format 13-22
- COBOL
 - and Unicode 14-1, 14-2
 - ANSI vs Unicode 14-17
 - character sets in 13-8
 - parsing Unicode strings 14-3, 14-20
 - Unicode conversion utility 14-3
- Configuration Manager
 - font setting 13-15
 - language setting 2-12
 - modifying list of languages 3-9
- ContainsCharType function 13-24
- ContainsOnlyCharType function 13-25
- ConvertChar function 13-26
- Crystal Reports globalization 1-6, 13-8, 15-27
- Cube Manager globalization 1-6, 13-8, 15-23
- currencies
 - amount fields 11-2
 - conversion *See* market rates
 - currency codes 11-1
 - PeopleCode for 12-26
 - resizing currency fields 11-7
- Currency Code page 12-11
- currency display
 - currency control fields 11-5
 - hiding and displaying currency fields 11-5
 - in SQR 15-12
 - Multi-Currency setting 11-5
- currency formatting
 - currency symbol 2-9, 11-4
 - decimal character 2-9
 - displayed on pages 11-4
 - setting with browser preferences 1-4
 - setting with PeopleTools 11-1
 - setting with Windows Regional Settings 2-13
 - thousands separator 2-9, 11-4
- Currency Quotation Method page 12-19

D

- Data Mover
 - swap base language command 3-1
 - using to import translations 3-14
- databases
 - character sets in 13-8, 13-12
 - field length checking, and 13-19
- date and time functions 10-7

- date formatting 2-8
 - in SQR 15-13
 - setting with Windows Regional Settings 2-13
- Daylight Savings page 10-10
- daylight savings time 10-1, 10-7, 10-10
- documentation, globalizing 1-8
- double-byte characters *See* character sets, double-byte

E

- effective dates
 - and time zones 10-1
- euro 12-1
 - triangulating via 12-14
- Exchange Rate Detail page 12-16
- exchange rates *See* market rates

F

- field length, understanding 13-18
- fields
 - currency *See* currency display
 - translating with design tools 6-3, 6-5
- file input/output
 - character sets in 13-8
- folder tabs, translating
 - with design tools 6-3, 6-9
 - with translate utilities 7-15
- fonts 13-15
- fonts, global 13-14, 13-15

G

- globalization
 - character sets *See* character sets
 - Crystal Reports 1-6, 15-27
 - Cube Manager 1-6, 15-23
 - documentation 1-8
 - globalization strategy 1-6
 - nVision 15-7
 - overview 1-1
 - page design considerations 6-19
 - Query 1-6
 - reporting and analysis tools 1-6, 15-1
 - SQR 15-12
 - translating *See* translating
 - upgrade considerations 1-8

H

- HTML objects, translating 6-16

I

- images, translating 6-15
- International Field Size page 11-8
- international settings 1-1, 2-1
 - and browser settings 2-7
 - changing during a PeopleSoft session 2-5
 - currency formatting 1-4
 - date and time formatting 1-4
 - for PeopleSoft Internet Architecture 2-2, 2-3
 - for Windows client 2-2, 2-12
 - PeopleCode for 2-6
- ISO locales 2-7, 3-5

J

- Japanese characters 13-22
 - standard name format 13-22

K

- keyboards 13-17
- Korean characters, standard name format 13-22

L

- language preferences *See also* international settings
 - %Language variable 2-6
 - %Language_Base variable 2-6
 - default signon language 2-7
 - defined 2-1
 - in Configuration Manager 2-12
 - in the user profile 2-10
 - in Windows Regional Settings 2-13
 - SetLanguage function 2-6
- languages 2-1
 - base and non-base 1-1
 - character sets 13-1
 - creating derivative translations 9-1
 - defining new languages 3-4
 - delivered 3-5
 - language codes 2-1, 3-5
 - multilingual support 5-4
- locales *See* ISO locales

M

- MAKEALTL.BAT 8-5
- MAKEUNIX.BAT 8-6
- Market Rate Index page 12-13
- market rates 12-1
 - application-specific requirements 12-36
 - calculated reciprocal rates 12-2
 - categories of 12-6

- cross rate 12-5
- Currency Code page 12-11
- Currency Quotation Method page 12-19
- current quote 12-17
- euro 12-1
- Exchange Rate Detail page 12-16
- Exchange Rate work record 12-25
- formulas 12-43
- functional requirements 12-1
- historic quote 12-17
- indices 12-6
- Market Rate Index page 12-13
- Market Rate table 12-7
- Market Rates page 12-14
- PeopleCode built-in functions 12-26
- PeopleCode library functions 12-27
- primary visual rate 12-5
- quotation methods 12-2, 12-10, 12-21
- quotation units 12-2
- Rate Definition page 12-22
- Rate Definition secondary page 12-16
- Rate Detail page 12-25
- Rate Type page 12-24
- RATE_MULT and RATE_DIV 12-4
- sample code 12-37
- tolerance checking 12-3, 12-10, 12-16, 12-22
- triangulation 12-3, 12-5
- types 12-9
- visual rate 12-4
- Market Rates page 12-14
- menu navigation
 - translating with design tools 6-3, 6-6
 - translating with translate utilities 7-11, 7-13
- messages
 - translating 7-16
- Microsoft Excel
 - and translation workbench 7-9
- MLS *See* multilingual support
- Multi-Currency
 - setting 11-5
- multilingual support 5-4

N

- name format
 - Chinese, Japanese, and Korean characters in 13-22
- navigation, translating
 - with design tools 6-3, 6-6
 - with translate utilities 7-11, 7-13
- number formatting
 - decimal character 2-9
 - thousands separator 2-9
- nVision
 - globalization 15-7
 - string variable evaluation 15-2

O

- override text (in pages) 7-21

P

- Page Field Properties
 - display time zone 10-6
- pages
 - checking for override text in 7-21
 - creating non-base language 6-20
 - designing for globalization 6-19
 - displaying time zones on 10-3
 - translating 6-2, 6-18
- PeopleBooks
 - CD-ROM, ordering xv
 - globalizing 1-8
 - printed, ordering xv
- PeopleCode 10-7
 - double-byte characters 13-22
 - font used in 13-16
 - for currencies 12-26
- PeopleSoft Internet Architecture
 - character sets for 13-14
- PeopleTools Options
 - base time zone 10-2
 - data field length checking 13-20
 - multi-currency setting 11-5, 11-7
- Process Scheduler definitions
 - translating 6-3, 6-12
- projects, translating objects in 7-2
- PSLANGUAGES table 3-8
- pstools.properties file 2-7, 3-11

Q

- queries
 - language-sensitive 1-6, 15-22
 - translating with translate utilities 7-18
- Query globalization 1-6

R

- Rate Definition page 12-22
- Rate Definition secondary page 12-16
- Rate Type page 12-24
- record definitions, translating
 - with design tools 6-3, 6-5
 - with translate utilities 7-9
- Record Field Properties
 - time zone 10-4
- Regional Settings, Windows 2-13
- related language tables 1-1, 4-1
 - creating 4-6
 - for views 4-8

- requirements for 4-3
- storage of translated values 4-4
- structure of 4-2
- system maintenance of 5-3
- translate table 6-2
- use of 4-4
- Replace Data page 9-18
- Replace Messages page 9-22
- Replace page 9-23
- Replace Results/Undo page 9-25
- resource translation 8-1
 - alternate-language DLLs 8-2
 - and UNIX application servers 8-5
 - application server files 8-1
 - MAKEALTL.BAT and MAKEUNIX.BAT 8-4
 - procedures 8-4
 - required files and tools 8-2
 - resource directories 8-3

S

- search and replace
 - and terminology management 9-1
 - in the translation workbench 7-8
 - security for 9-3
- Search page 9-15
- Search Profile page 9-5
- security
 - for terminology search and replace 9-3
- SetLanguage function 2-6
- signon language
 - and translating application data 5-1
 - choosing 2-3
 - default 2-7
 - defined 2-3
- signon page
 - adding new languages 3-10
 - choosing a language 2-3
- single signon 2-6
- single-byte characters *See* character sets, single-byte
- SQR globalization 15-12
- Strings table 15-1, 15-3
 - translating strings 15-6
- SWPAUDIT.SQR 3-1
- system variables
 - %Language 2-6
 - %Language_Base 2-6

T

- terminology management 1-4, 9-1
 - cloning search criteria 9-10
 - in messages 9-22
 - inquiry pages 9-29
 - overview 9-1
 - preventing replace processing 9-3, 9-6

- replacing terms 9-18
- search privileges 9-3
- search result, viewing 9-18
- search statuses 9-3
- searchable objects, configuring 9-34
- searchable objects, understanding 9-32
- text labels, checking pages for 7-21
- Time and date functions 10-7
- time formatting
 - in SQR 15-13
 - in Windows Regional Settings 2-13
- Time Zone Data page 10-8
- time zones 1-4, 10-1
 - base time zone 10-1, 10-2
 - daylight savings time 10-1, 10-7, 10-10
 - defining time zones 10-7
 - displayed on pages 10-3
 - PeopleCode functions 10-7
 - time zone control fields 10-4
 - types of time zones 10-1
 - user time zone 10-2
 - variable time zone 10-2
- TLSINST1.SQR 11-7
- translate table
 - adding new language codes 3-7
 - translating with design tools 6-16
 - views over 4-14
- translating 1-2. *See also* terminology management
 - application data 1-3, 5-1
 - choosing what to translate 1-7
 - fields 6-3, 6-5
 - folder tabs 6-3, 6-9, 7-15
 - HTML objects 6-16
 - images 6-15
 - menu navigation 6-3, 6-6, 7-11, 7-13
 - messages 7-16
 - pages 6-2, 6-18
 - PeopleTools objects 1-3, 6-2
 - Process Scheduler definitions 6-3, 6-12
 - queries 7-18
 - record definitions 6-3, 6-5, 7-9
 - system resources *See* resource translation
 - translate (XLAT) values 6-16, 7-2
 - translate utilities vs. translating with design tools 7-1
 - trees 6-4, 6-13
 - windows resources 1-4
 - with design tools 6-1, 6-3
 - with the multilingual support feature 5-4
 - with the translation workbench *See* translation workbench
 - with translate utilities 7-1
 - workflow maps 6-3, 6-10
- translation workbench 7-1
 - expanding and collapsing nodes 7-5
 - navigating in 7-7
 - opening 7-2
 - resizing columns and rows 7-6

- search and replace 7-8
- viewing and hiding rows 7-5
- translations
 - loading into the database 3-12
 - related language table architecture 4-1
- trees
 - translating 6-4, 6-13

U

- Undo page 9-28
- Unicode
 - and COBOL 14-1, 14-2
 - and SQR 15-17
 - character encoding standards 13-3, 14-1
 - converting character sets 13-9
 - defined 13-3
 - operations that don't support Unicode 3-6
 - Windows support for 13-16
- upgrade
 - and global considerations 1-8, 3-15, 9-35
- user profile
 - language preference 2-10

- User Search Options page 9-3

V

- variables
 - %Language 2-6
 - %Language_Base variable 2-6

W

- Windows
 - Unicode support in 13-16
- Windows client
 - character sets for 13-15
- Windows Regional Settings 2-13
- workflow maps, translating 6-3, 6-10

X

- XLATTABLE *See* translate table

