PeopleSoft PeopleTools 8.63

Global Technology

G47437-01
July 2026

PeopleSoft PeopleTools 8.63 Global Technology,

G47437-01

Copyright © 1988, 2026, Oracle and/or its affiliates.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle®, Java, MySQL, and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Oracle customer access to and use of Oracle support services will be pursuant to the terms and conditions specified in their Oracle order for the applicable services.

Contents

Preface

Understanding the PeopleSoft Online Help and PeopleBooks i
Using and Managing the PeopleSoft Online Help iv
PeopleTools Related Links iv
Contact Us iv
Follow Us v

1 Getting Started with Global Technology

PeopleTools Global Technology Overview 1
PeopleTools Global Technology Implementation 2

2 Controlling International Preferences

Understanding International Preferences 1
Setting Up Language Preferences in the PeopleSoft Pure Internet Architecture 1
- Understanding the Session Language 2
- Choosing a Session Language 2
- Changing the Session Language While Signed In 3
- Changing the Session Language Programmatically 3
- Determining the Session Language When the Sign-in Page is Bypassed 4
- Controlling the User Profile Language Preference 4
- Determining the Language of Help Documentation 5

Applying Browser Language Preferences 5
- Understanding Browser Language Preferences 5
- Setting Browser Language Preferences 5
- Using Browser Language Settings to Determine the Sign-in Language 5

Setting Up Language Preferences for Microsoft Windows-Based PeopleTools 7
- Understanding Language Preferences for Microsoft Windows-Based PeopleTools 7
- Controlling PeopleSoft Configuration Manager Language Settings 7

Setting Up Locale-Based Formatting for the PeopleSoft Pure Internet Architecture 7
- Understanding Locale-Based Formatting 8
- Defining User-Specific, Locale-Based Formatting 9
- Defining Locales for Locale-Specific Formatting 9

Copyright © 1988, 2026, Oracle and/or its affiliates.

Defining Locale-Specific Formatting 11
Applying System-Wide Default Formatting 11
Setting Up Locale-Based Formatting for Microsoft Windows-based PeopleTools 12

3 Setting and Maintaining Time Zones

Understanding Time Zones 1
Setting the Base Time Zone 2
Maintaining Time Zones 4
Understanding Time Zone Maintenance 4
Defining Time Zones 4
Generating Time Zone Offsets 7
Defining DST IDs 8
Mapping Time Zone 9
Updating Time Zone Definitions to Comply with Legislative and Other Changes 13
Controlling Time Zone Display When Developing PeopleSoft Application Objects 13
Understanding Time Zone Display Settings 13
Understanding Time Zone Precedence 14
Setting Time Zone Options and Related Date Fields in Record Field Properties 15
Setting Time Zone Options in Page Field Properties 16
Using PeopleCode Time Functions 16

4 Selecting and Configuring Character Sets

Understanding Character Sets 1
Character Sets 1
Common Character Sets 1
The Unicode Standard 3
Non-Unicode Character Sets 8
Character Sets Across the Tiers of the PeopleSoft Architecture 10
Selecting Character Sets 15
Understanding Character Set Selection 15
Selecting Database Character Sets 16
Selecting Application Server Character Sets 18
Selecting and Managing Client Workstation Character Sets 18
Selecting Email Character Sets 21
Converting Between Character Sets 22
Setting Data Field Length Checking 24
Understanding Application Designer Field Length Semantics 24
Understanding Field Length Checking for Non-Unicode Databases 25
Enabling or Disabling Data Field Length Checking 26
Using CJK Ideographic Characters in Name Character Fields 27

Copyright © 1988, 2026, Oracle and/or its affiliates.

Detecting and Converting Between Character Types 30

5 Controlling Currency Display Format

Understanding Currency-Specific Settings 1
Setting Up Currency Amount Fields When Developing Applications 1
Setting Currency Field Display Properties When Developing Applications 2
Using System-Wide Multicurrency Settings 3
- Understanding Multicurrency Settings 3
- Connecting a Currency Control Field to the Multi-Currency Check Box 3
- Enabling or Disabling System-Wide Currency Settings 4

Resizing Currency Fields by Using the International Field Size Utility 4

6 Running COBOL in a Unicode Environment

Understanding COBOL in a Unicode Environment 1
- Unicode Encodings in PeopleSoft COBOL 1
- Expanded Storage Space Requirements 2
- Special Logic for Parsing Unicode Strings 3
- COBOL Sorting 3
- Unicode-Specific Error Messages 4

Running the COBOL Conversion Utility 4
- Understanding the COBOL Conversion Utility 4
- Running the Conversion Utility 5
- Identifying Converted COBOL Programs 6
- Understanding What Is Expanded 6
- Using Utility Directives 11
- Viewing Error Logs 13

Fine-Tuning COBOL Programs 14
- Identifying Unicode and Non-Unicode Data 14
- Specifying Column Lengths in Dynamic SQL 15
- Defining Single Character Arrays 16

7 Running COBOL in a z/OS Unicode Environment

Understanding Running COBOL in a z/OS Unicode Environment 1
- Unicode Encodings in PeopleSoft COBOL on z/OS 1
- Data Usage for Unicode Storage 2
- Input/Output in the z/OS Unicode Environment 2

Understanding the COBOL Unicode Conversion Utility for z/OS 4
- COBOL Unicode Conversion Utility for z/OS 4
- Applying Patches 4

Copyright © 1988, 2026, Oracle and/or its affiliates.

Running the COBOL Unicode Conversion Utility for z/OS 4
Automatically Running the COBOL Unicode Conversion Utility for z/OS 5
Manually Running the COBOL Unicode Conversion Utility for z/OS 5
Identifying Converted COBOL Programs 6
Understanding Converted Data 6
Copybook Conversion 6
Data Division Conversion Rules 7
Exceptions to Data Division Conversion Rules 8
Procedure Division Conversion Rules 10
Columns 12
Using Error, Exception and Summary Logging 12
Understanding Error, Exception and Summary Logs 12
Setting Logging Parameters 12
Viewing Exception Log Messages 13
Viewing Summary Log Messages 13
Fine-Tuning COBOL Programs for the z/OS Unicode Environment 14
Understanding Fine-Tuning Converted COBOL Programs for the z/OS Unicode Environment 14
Identifying Unicode Data for z/OS, Unicode Data for Microsoft Windows/Unix and Non-Unicode Data 14
Understanding Character Fields and Byte Size 15
Understanding Character Set Conversion 16

8 Sorting in PeopleTools

Understanding Sort Orders 1
Sorting Overview 1
PeopleTools Sorts 2
Database-Level SQL ORDER BY Sorts 3
PeopleTools In-Memory Sorts 3
Binary Sorts 4
Setting the Sort Order 5
Forcing a Binary Sort in SQL 5
Sorting in COBOL 6
Sorting in SQR Programs 7
Performing Linguistic Sorting as a Customization 7

9 Using Global Reporting and Data Analysis Tools

Using Language-Sensitive Queries 1
Using Scheduled Queries 1
Using the Strings Table for Language-Sensitive Text in Reports 2
Using SQR for PeopleSoft in Global Implementations 5

Copyright © 1988, 2026, Oracle and/or its affiliates.

Printing for A4 Paper 5
Currency Precision 5
Date and Time Formatting 5
Report Translation 6
The PSSQR.INI and PSSQR.UNX Files 9
SQR Configuration for Processing International Text 10
SQR Configuration for Printing International Text 11
International Text in SQR for PeopleSoft Programs 20
SQR for PeopleSoft-Supported Character Set Encodings 23
Using PS/nVision Reporting for Global Implementations 31
PS/nVision Design-Time Globalization Features 32
PS/nVision Run-Time Language Features 35
International Versions of Microsoft Excel 37
BI Publisher 37

10 Developing Global Applications

Developing Easily Translatable Applications 1
Designing Global-Ready Pages 2

11 Using Related Language Tables

Understanding Related Language Tables 1
Related Language Tables Overview 1
Related Language Table Structure 3
How Related Language Tables Store Translations 5
How Related Language Tables Are Used 6
Installing Oracle-Provided Translations 7
Understanding Translation Installation 7
Adding a New Language 8
Swapping the Base Language 9
Understanding Base Language Swapping 9
Prerequisites 9
Running the Swap Audit Report 10
Checking Space and Resources 15
Running the SWAP_BASE_LANGUAGE Data Mover Command 15
Managing and Patching Swapped Language Database 16
Creating Related Language Tables 17
Creating Related Language Views 19
Understanding Related Language Views 19
Creating Views for One Base Table and One Related Language Table 19
Creating Views for Two Base Tables and One Related Language Table 21

Copyright © 1988, 2026, Oracle and/or its affiliates.

Creating Views for Two Base Tables and Two Related Language Tables 23

12 Working With Language-Sensitive Application Data

Understanding Data Editing in Related Language Tables and Base Tables 1
Editing Data in Multiple Languages 2
Understanding Multi-Language Entry 2
Enabling Multi-Language Entry 3
Entering Data in Multiple Languages 4

13 Implementing Bidirectional Language Support

Understanding Bidirectional Language Support 1
Features 1
Translations Provided 1
Navigating in Bidirectional PeopleSoft Pure Internet Architecture Pages 1
Customizing PeopleSoft Pure Internet Architecture Pages for Bidirectionality 2
Working with HTML Areas and Style Sheets 2
Working with Images 3
Working with Bidirectional Languages and Fluid 4

14 Modifying Terminology

Understanding Terminology Management 1
Terminology Management Overview 1
Terminology Searches 2
Terminology Replacement 2
Terminology Replacement Undo Process 3
Terminology Search Statuses 3
Granting Terminology Search Privileges 4
Searching Terminology 5
Defining Search Criteria 5
Cloning Existing Search Criteria 11
Running the Search Process 12
Viewing and Replacing Terminology Search Results 12
Replacing Non-Message Text 13
Replacing Message Text 16
Running the Replace Process 17
Viewing and Undoing Terminology Replacement Results 17
Undoing Data Replacements 18
Undoing Text Replacements 19
Running the Undo Process 19

Copyright © 1988, 2026, Oracle and/or its affiliates.

Reviewing Terminology Search Information 20
Reviewing Search Criteria 20
Viewing Results for Search Only Searches 20
Viewing Search Results for All Searches 21
Selecting Terminology Search Objects 21
Viewing Searchable Objects 21
Modifying Searchable Objects 22
Reviewing Upgrade Considerations 22

15 Adding New Languages

Understanding the Addition of New Language Definitions 1
Adding New Language Codes to the System 1
Determining PeopleSoft and ISO Codes for Your Language 2
Determining Appropriate Non-Unicode Character Sets 3
Adding New Translate Values to the LANGUAGE_CD Field 4
Managing Languages in the PSLANGUAGES Table 5
Modifying Configuration Manager Windows Resources 8
Modifying the PeopleSoft Pure Internet Architecture Sign-in Page 9
Modifying Text and Error Properties Files for New Languages 10
Adjusting Truncated Toolbar Buttons 12

16 Translating PeopleTools

Understanding PeopleTools Translation 1
Using Alternate Language DLLs 1
Locating Resource Directories 3
Translating Resource Files 4
Translating Resource Files 5
Compiling Translated Resource Files Using MAKEALTL.BAT 6
Compile Translated Resource Files Using MAKEUNIX.BAT 7
Translating Objects 7
Translating Properties Files 7
Translating Images 8

17 Translating Application Definitions

Understanding Application Definition Translation 1
Using Translation Designer 1
Opening Translation Designer 1
Using Translation Designer Display Options 3
Working in Translation Designer 5

Copyright © 1988, 2026, Oracle and/or its affiliates.

Translating Definitions 8
Understanding the Translation Process 8
Searching the Object Definitions Alphabetically 8
ACE Analytic Model ID Definitions 9
Application Data 9
Application Engine Descriptions 12
BI Publisher Definitions 12
Business Processes 14
Composite Query 16
Connected Query Definitions 17
Data Set Definitions 18
Feed Definitions 18
Integration Broker Definitions 21
Integration Groups 23
Menu and Component Objects 23
Messages 27
Page Objects 28
PeopleSoft Process Scheduler Objects 29
Portals 31
PeopleSoft Search Definitions 31
Records 32
Related Content Definitions 33
Review Page Text 35
Queries 36
Strings 37
Time Zone Labels 38
Trees 39
Translating HTML Definitions 40
Providing Context Information 40

18 Converting PeopleSoft Systems to Unicode Databases

Understanding Converting PeopleSoft Systems to Unicode Databases 1
Understanding Planning Unicode Conversions 1
Platform Support 1
Database Sizing 1
Coexistence of Unicode and Non-Unicode Databases 2
Converting to Unicode on Oracle Databases 2
Understanding Converting PeopleSoft Systems on Oracle Databases to Unicode 3
Exporting PeopleSoft Table Structures Using PeopleSoft Data Mover 3
Exporting Database Contents 4
Creating a New Oracle Database Instance 5

Copyright © 1988, 2026, Oracle and/or its affiliates.

Pre-Creating the PeopleSoft Table Structures Using PeopleSoft Data Mover 8
Importing Database Contents 9
Specifying the Unicode Database and Data Types in Your PeopleSoft System 10
Running GRANT.SQL 11
Rerunning an Oracle Database Import 11
Converting to Unicode on Oracle Databases Using Database Migration Assistant for Unicode 11
Verifying a Database After Migrating to Unicode: 13
Rerunning Microsoft SQL Server and DB2 Database Imports 14
Final Database Cleanup 14
Converting to Unicode on Microsoft SQL Server and DB2 Databases 15
Understanding Converting to Unicode on Microsoft SQL Server and DB2 Databases 15
Exporting PeopleSoft Databases Using PeopleSoft Data Mover 15
Creating New SQL Server or DB2 Unicode Databases 15
Importing Database Contents Using PeopleSoft Data Mover 16
Specifying Unicode Databases in PeopleSoft Systems 16
Rerunning GRANT.SQL 16

19 Updating Time Zone Definitions

Understanding the U.S. Energy Policy Act of 2005 1
Adding New DST IDs 1
Update Affected Time Zones 2
Generate New Query Offsets 2
Define New Time Zones 3

20 Troubleshooting

Understanding Troubleshooting 1
PeopleSoft Hot Keys Do Not Function As Expected on a non-U.S. Keyboard 1
Daylight Saving Time Issues on Oracle WebLogic Server 4
Special International Characters Appear Corrupted in the View Log/Trace Files 4

Copyright © 1988, 2026, Oracle and/or its affiliates.

Preface

Understanding the PeopleSoft Online Help and PeopleBooks

The PeopleSoft Online Help is a website that enables you to view all help content for PeopleSoft applications and PeopleTools. The help provides standard navigation and full-text searching, as well as context-sensitive online help for PeopleSoft users.

Hosted PeopleSoft Online Help

You can access the hosted PeopleSoft Online Help on the Oracle Help Center. The hosted PeopleSoft Online Help is updated on a regular schedule, ensuring that you have access to the most current documentation. This reduces the need to view separate documentation posts for application maintenance on My Oracle Support. The hosted PeopleSoft Online Help is available in English only.

To configure the context-sensitive help for your PeopleSoft applications to use the Oracle Help Center, see Configuring Context-Sensitive Help Using the Hosted Online Help.

Locally Installed PeopleSoft Online Help

If you're setting up an on-premises PeopleSoft environment, and your organization has firewall restrictions that prevent you from using the hosted PeopleSoft Online Help, you can install the online help locally. Installable PeopleSoft Online Help is made available with selected PeopleSoft Update Images and with PeopleTools releases for on-premises installations, through the Oracle Software Delivery Cloud.

Your installation documentation includes a chapter with instructions for how to install the online help for your business environment. The documentation zip file may contain a README.txt file with additional installation instructions. See *PeopleSoft 9.2 Application Installation* for your database platform, "Installing PeopleSoft Online Help."

Installable PeopleSoft Online Help now includes PDF versions of each PeopleBook. For more information, see Understanding Locally Installed PeopleSoft Online Help.

Downloadable PeopleBook PDF Files

You can access downloadable PDF versions of the help content in the traditional PeopleBook format on the Oracle Help Center. The content in the PeopleBook PDFs is the same as the content in the PeopleSoft Online Help, but it has a different structure and it does not include the interactive navigation features that are available in the online help.

Common Help Documentation

Common help documentation contains information that applies to multiple applications. The three main types of common help are:

Copyright © 1988, 2026, Oracle and/or its affiliates.

- Application Fundamentals
- Using PeopleSoft Applications
- Using and Managing the PeopleSoft Online Help

Most product families provide a set of application fundamentals help topics that discuss essential information about the setup and design of your system. This information applies to many or all applications in the PeopleSoft product family. Whether you are implementing a single application, some combination of applications within the product family, or the entire product family, you should be familiar with the contents of the appropriate application fundamentals help. They provide the starting points for fundamental implementation tasks.

In addition, the *PeopleTools: Applications User's Guide* introduces you to the various elements of the PeopleSoft Pure Internet Architecture. It also explains how to use the navigational hierarchy, components, and pages to perform basic functions as you navigate through the system. While your application or implementation may differ, the topics in this user's guide provide general information about using PeopleSoft applications.

Finally, *Using and Managing the PeopleSoft Online Help* (this guide) introduces you to the setup, features, and usage of the online help system.

Field and Control Definitions

PeopleSoft documentation includes definitions for most fields and controls that appear on application pages. These definitions describe how to use a field or control, where populated values come from, the effects of selecting certain values, and so on. If a field or control is not defined, then it either requires no additional explanation or is documented in a common elements section earlier in the documentation. For example, the Date field rarely requires additional explanation and may not be defined in the documentation for some pages.

Typographical Conventions

The following table describes the typographical conventions that are used in the online help.

Typographical Convention	Description
Key+Key	Indicates a key combination action. For example, a plus sign (+) between keys means that you must hold down the first key while you press the second key. For **Alt+W,** hold down the **Alt** key while you press the **W** key.
. . . (ellipses)	Indicate that the preceding item or series can be repeated any number of times in PeopleCode syntax.
{ } (curly braces)	Indicate a choice between two options in PeopleCode syntax. Options are separated by a pipe (|).
[ ] (square brackets)	Indicate optional items in PeopleCode syntax.
& (ampersand)	When placed before a parameter in PeopleCode syntax, an ampersand indicates that the parameter is an already instantiated object. Ampersands also precede all PeopleCode variables.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Typographical Convention	Description
⇒	This continuation character has been inserted at the end of a line of code that has been wrapped at the page margin. The code should be viewed or entered as a single, continuous line of code without the continuation character.

ISO Country and Currency Codes

PeopleSoft Online Help topics use International Organization for Standardization (ISO) country and currency codes to identify country-specific information and monetary amounts.

ISO country codes may appear as country identifiers, and ISO currency codes may appear as currency identifiers in your PeopleSoft documentation. Reference to an ISO country code in your documentation does not imply that your application includes every ISO country code. The following example is a country-specific heading: "(FRA) Hiring an Employee."

The PeopleSoft Currency Code table (CURRENCY_CD_TBL) contains sample currency code data. The Currency Code table is based on ISO Standard 4217, "Codes for the representation of currencies," and also relies on ISO country codes in the Country table (COUNTRY_TBL). The navigation to the pages where you maintain currency code and country information depends on which PeopleSoft applications you are using. To access the pages for maintaining the Currency Code and Country tables, consult the online help for your applications for more information.

Region, Country, and Industry Identifiers

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

An example of a region-specific heading is: "(Latin America) Setting Up Depreciation"

Region Identifiers

Regions are identified by the region name. The following are examples of region identifiers that may appear in PeopleSoft Online Help:

- Asia Pacific
- Europe
- Latin America
- North America

Country Identifiers

Countries are identified by a three-letter country code. The following are examples of country identifiers that may appear in the PeopleSoft Online Help:

- ARG (Argentina)
- AUS (Australia)
- CAN (Canada)
- NLD (The Netherlands)

Copyright © 1988, 2026, Oracle and/or its affiliates.

Industry Identifiers

Industries are identified by the industry name or by an abbreviation for that industry. The following are examples of industry identifiers that may appear in the PeopleSoft Online Help:

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

Translations and Embedded Help

PeopleSoft 9.2 software applications include translated embedded help. With the 9.2 release, PeopleSoft aligns with the other Oracle applications by focusing our translation efforts on embedded help. We are not planning to translate our traditional online help and PeopleBooks documentation. Instead we offer very direct translated help at crucial spots within our application through our embedded help widgets. Additionally, we have a one-to-one mapping of application and help translations, meaning that the software and embedded help translation footprint is identical—something we were never able to accomplish in the past.

Using and Managing the PeopleSoft Online Help

Select About This Help in the left navigation panel on any page in the PeopleSoft Online Help to see information on the following topics:

- Using the PeopleSoft Online Help.
- Managing hosted Online Help.
- Managing locally installed PeopleSoft Online Help.

PeopleTools Related Links

PeopleSoft 8.63 Home Page

PeopleSoft Search and Insights Home Page

PeopleSoft Online Help

PeopleSoft Information Portal

PeopleSoft Spotlight Series

PeopleSoft Training and Certification | Oracle University

My Oracle Support

Oracle Help Center

Contact Us

Send your suggestions to psoft-infodev_us@oracle.com.

Please include the applications update image or PeopleTools release that you're using.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Follow Us

Icon	Link
	Watch PeopleSoft on YouTube
	Follow @PeopleSoft_Info on X.
	Read PeopleSoft Blogs
	Connect with PeopleSoft on LinkedIn

Copyright © 1988, 2026, Oracle and/or its affiliates.

1

Getting Started with Global Technology

PeopleTools Global Technology Overview

Oracle's PeopleSoft applications are functionally equipped for implementation in global enterprises. However, country-specific application business logic is not enough to make an application global-ready. The technology behind the applications, Oracle's PeopleTools, provides the core global functionality that is common to all PeopleSoft application products, including support for multiple languages in a single database, data display formats that use the standards that users expect with their language and country, and multiple time zone support. PeopleTools is designed so that a single implementation can serve users in different regions: users can share the same data while operating in different languages and applying different date, time, and numeric formatting conventions.

This PeopleBook covers many aspects of global technology implementation of interest to a variety of users:

- Administrators responsible for installing and maintaining the PeopleSoft database and applications.
- Developers designing new PeopleTools and PeopleSoft application objects.
- Translators translating PeopleTools and PeopleSoft application objects into new languages.
- End users setting up their user preferences or working with multiple languages.

PeopleTools global technology implementation can be divided into four main task areas:

1. Installation-level globalization tasks.

   These tasks include selecting and setting up character sets and sort orders, installing PeopleSoft-delivered translations, and setting up currency and time zone options.

   Many installation-level globalization tasks are required in a multinational implementation, even if you use only English-language data.

2. Post-installation globalization setup tasks.

   These tasks include a variety of setup tasks, some of them optional, that can be performed by administrators, developers, and end users.

3. PeopleSoft application translations.

   The PeopleSoft application delivers non-English translations in a number of languages, which enables users to process data in multiple languages from a single database. The PeopleSoft application also provides tools that enable users to modify delivered translations to suit local needs.

   You may also need to translate applications or application objects into languages that are not supplied by the PeopleSoft application, or you may need to translate your customizations. PeopleTools provides translation tools and recommendations for successful translations.

4. Global application object development.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Development for global applications requires special tools and special considerations. This book describes the PeopleTools that developers need and the practices that developers should follow to develop successful global, translatable applications.

PeopleTools Global Technology Implementation

Before using PeopleSoft applications in a multinational, multilanguage environment, you may need to perform some globalizing implementation tasks in addition to your standard PeopleSoft installation and implementation tasks.

Installation-Level Globalization Setup and Maintenance Tasks

Most installation tasks for a global implementation are covered in your PeopleSoft installation guide, but some global-only tasks are covered in this PeopleBook and are listed here. These tasks are performed primarily by system and database administrators. You can also perform some of these tasks after installation, if needed.

Step	Reference
Review the character set requirements for your database, application servers, and client workstations, and select an appropriate character set for your database.	See Understanding Character Sets.
If required, convert your existing non-Unicode database to Unicode.	See Understanding Character Sets. See also the special Unicode database conversion documentation available on My Oracle Support.
Set the database sort order. Most database systems require that you select a sort order when you create the database. **Note:** You should also set the PeopleTools in-memory sort options during installation, although you can change these after installation.	See Understanding Sort Orders.
Install the PeopleSoft-delivered translations that you want, using the PeopleSoft Global Multi-Language CD-ROM. **Note:** If you are installing a new PeopleSoft database, we recommend that you install these translations at the same time that you install the English-language database. However, you may also install translations to existing PeopleSoft databases at any time.	See Installing Oracle-Provided Translations. **Note:** Multilanguage installation steps are covered in the PeopleSoft installation guide for your database platform. The PeopleTools Global Technology PeopleBook covers only installation steps for adding or upgrading additional languages to an existing PeopleSoft database.
If the PeopleSoft database base language (language in which data is stored in the PeopleSoft core tables) is not to be English, swap the base language.	See Understanding Related Language Tables.
Set up time zones and generate query offsets. **Note:** You can also maintain some time zone settings after installation.	See Understanding Time Zones.
If you plan to use currency conversion, set system-wide multicurrency options and resize currency fields, if necessary. **Note:** You can set currency settings at installation time or later.	See Using System-Wide Multicurrency Settings. See Resizing Currency Fields by Using the International Field Size Utility.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Step	Reference
If you are running a Unicode database, convert any COBOL programs to a Unicode environment by running the COBOL conversion utility.	See Understanding COBOL in a Unicode Environment.

Post-Installation Globalization Setup

After installing the PeopleSoft database, there are setup tasks that administrators, developers, or end users may need or want to perform to complete their global implementation. Most of the tasks in this list are optional, and depend on your organization's particular needs.

> **Note**
>
> Some tasks listed previously in the Installation-Level Globalization Setup and Maintenance Tasks section can also be performed after installation.

Step	Reference
Set up language preferences for PeopleSoft Pure Internet Architecture display, your browser, and Microsoft Windows-based PeopleTools. **Note:** This setup is for language preferences only. Language preferences do not affect locale-specific formatting (such as number separators or date format).	See Setting Up Language Preferences in the PeopleSoft Pure Internet Architecture. See Applying Browser Language Preferences. See Setting Up Language Preferences for Microsoft Windows-Based PeopleTools.
Set up PeopleSoft Pure Internet Architecture locale-based formatting, such as date format and number separators. You can set up locale-based formatting defaults at three levels: user, locale, and system-wide. The user-defined defaults override the locale-specific defaults, which override the system-wide defaults. User-defined defaults can be set up by end users. Locale-specific and system-wide defaults should be performed by a systems administrator or similar role. This task is necessary only if you need to override the delivered formatting.	See Setting Up Locale-Based Formatting for the PeopleSoft Pure Internet Architecture.
Enable multilanguage data entry for users who will enter data in multiple languages during a single session. This task can be performed by end users or administrators.	See Editing Data in Multiple Languages.

Translation Tasks

To modify PeopleSoft-delivered terminology in any delivered language, including English, perform the following task:

Copyright © 1988, 2026, Oracle and/or its affiliates.

Step	Reference
Modify PeopleSoft-delivered terminology and translations to meet local requirements.	See Understanding Terminology Management.

If you have developed new application objects, or if you need to translate PeopleSoft-delivered PeopleTools and application objects into languages that the PeopleSoft application does not deliver, you may need to perform the following tasks:

Step	Reference
Define any languages you need that are not delivered by the PeopleSoft application.	See Understanding the Addition of New Language Definitions.
Translate PeopleTools objects into non-delivered languages or modify PeopleSoft-delivered translations of PeopleTools objects.	See Understanding the Addition of New Language Definitions.
Translate PeopleSoft application definitions into non-delivered languages (or modify delivered PeopleSoft translations) using the Translation Designer feature, Translation pages, and Application Designer.	See Understanding Application Definition Translation.

Global Application Development Tasks

If you are developing new application objects for use in a global implementation, you may need to perform the following tasks.

Step	Reference
Review best practices for developing easily translatable applications.	See Developing Easily Translatable Applications.
Control time zone display.	See Understanding Time Zones.
Validate field lengths for the character sets you will use.	See Understanding Character Sets. See Setting Data Field Length Checking.
Design appropriate field formats for character sets you will use.	See Using CJK Ideographic Characters in Name Character Fields.
Set up currency amount and currency control fields.	See Understanding Currency-Specific Settings.
Use appropriate sorts in the objects that you develop.	See Understanding Sort Orders.
Design reports and configure reporting tools for global requirements.	See Using Language-Sensitive Queries. See also discussions of language- and locale-specific formatting considerations in the relevant reporting tools PeopleBooks.
Create related language tables and views for any language-sensitive tables and views that you create or modify.	See Understanding Related Language Tables.

PeopleTools also delivers a number of globalization-related PeopleCode functions.

See PeopleCode Language Reference.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Other Sources of Information

In the implementation planning phase, take advantage of all PeopleSoft sources of information, including the installation and upgrade guides, release notes, red papers, updates and fixes posted to My Oracle Support, PeopleSoft curriculum guides, and other PeopleBooks.

Related Topics

- Getting Started with PeopleTools

Copyright © 1988, 2026, Oracle and/or its affiliates.

2

Controlling International Preferences

Understanding International Preferences

The international preference settings control the language in which users view and use the PeopleSoft system, as well as locale formatting conventions for dates, times, and numbers. The PeopleSoft system treats language preferences and local formatting preferences separately.

Language Preferences

You can set up language preferences for your internet browser, for the PeopleSoft Pure Internet Architecture (by session and by user profile), and for the PeopleSoft Windows-based client. Your user, or session language, helps to determine the language of your PeopleSoft interface, such as field and menu labels, as well as the language of the data that you maintain or view during the session. The internet browser language setting helps to determine the language of your PeopleSoft sign-in page in the PeopleSoft Pure Internet Architecture.

Locale-Based Formatting Preferences

Locale-based formatting determines such preferences as date, time, and number formatting. You can set up locale-based formatting preferences at three levels, each of which overrides the one above it:

1. System-wide
2. Locale-specific
3. User-specific

You can set these formatting preferences for both the PeopleSoft Pure Internet Architecture and for the PeopleTools development environment. The preferences you set up for one is independent of those you set up for the other.

> **Note**
>
> Only one date format is supported per selected language.

Setting Up Language Preferences in the PeopleSoft Pure Internet Architecture

This section discusses how to set the session language, the user profile language and the help documentation language.

Related Topics

- Understanding Related Language Tables
  This section discusses related language tables.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Understanding the Session Language

In the PeopleSoft Pure Internet Architecture, the *session language* is the language selected by the user on the sign-in page before entering a user name and password. The session language controls the language in which the user views language-sensitive data and objects in the application for the duration of that session.

If the session language is the database's base language, then all language-sensitive data, messages, and objects appear in the base language. If the session language is a non-base language, then any language-sensitive data, messages, and objects that have been translated appear in the preferred language; those that have not been translated appear in the base language. For example, if a user's language for a particular session is Spanish, PeopleTools attempts to display menus and field labels in Spanish for each page viewed in that session, and if the table from which the data is being fetched has a related language record associated with it, PeopleTools also attempts to fetch the Spanish translations of the data from this table.

Choosing a Session Language

Users choose a session language when they sign in to the system. Users can change their session language during the PeopleSoft session; however, this is typically done only for development or testing purposes. You can also program applications to change session languages, so that your application has its own language selection options.

When the sign-in page initially appears, the page elements on that page appear in the language designated in the user's browser language settings.

> **Note**
>
> You can prevent a sign-in page from appearing for a language that is not installed in your system by deleting that language from the PSLANGUAGES table using the Maintain Languages page.

To choose a different session language, click one of the available language links before signing in. Clicking the link updates the sign-in screen to reflect the new language choice, and signing in using the updated screen confirms the new session language. Only languages that have been installed and enabled by the system administrator can be used to sign in to the database. If you select a language that has not been enabled in the database to which you are logging in, you receive an error message.

To enable and disable languages for signing in to the database, use the Maintain Languages page. To access this page, select **PeopleTools**, and then **Utilities**, and then **International**, and then **Manage Installed Languages.**

If you will not be using all languages delivered by the PeopleSoft system, or if you need to add a new language to the system, you can modify the sign-in page to add or remove language links. Remember, the base language of your system must always be available.

Related Topics

- Understanding the Addition of New Language Definitions
  PeopleSoft provides translations in several 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, you can

Copyright © 1988, 2026, Oracle and/or its affiliates.

maintain data in all languages supported by the Unicode standard in a single PeopleSoft database.

Changing the Session Language While Signed In

During development, testing, or debugging, you might want to change the language of your session without signing out of the PeopleSoft system.

Use the International Preferences page to change your session language during a session. Any change made using this page is valid only for the remaining duration of the current session.

To change your session language without signing out:

1. Select **PeopleTools**, and then **Utilities**, and then **International**, and then **Reset My Session Language.**

   The International Preferences page appears.
2. Select a language in the **Language Preference** field.
3. Click the Save button.

Changes take effect when a new component is loaded.

Changing the Session Language Programmatically

You can change user language preferences programmatically for the remainder of a PeopleSoft Pure Internet Architecture session. This feature is useful if you are building a front end to your PeopleSoft system that does not involve the sign-in screen, and if you want to enable users to change their language preference at any time.

For example, an internet store application might enable anonymous sign-in by customers of the store to browse a catalog, and therefore would not display the sign-in screen. The PeopleCode SetLanguage built-in function could be linked to a PeopleCode program behind a button on a page to enable users to change the language of the interface after they have already been signed in as guest users.

The SetLanguage function temporarily overrides the user's default language preference and the *%Language* system variable, which evaluates to the language for the current preferred language.

Calling the SetLanguage function is the same as changing the **Language Preference** field on the International Preferences page.

The *%Language_Base* system variable makes it simple to determine the current preferred language and build language-specific functionality in PeopleCode.

If multi-language entry is enabled, *%Language_Data* returns the current data language selected by the user. Use *%Language_Data* if your application must know the language any entered application data is stored as in the component's related language records.

Related Topics

- PeopleCode Language Reference: SetLanguage function
- PeopleCode Language Reference: %Language system variable
- PeopleCode Language Reference: %Language_Base system variable
- PeopleCode Language Reference: %Language_User system variable

Copyright © 1988, 2026, Oracle and/or its affiliates.

Determining the Session Language When the Sign-in Page is Bypassed

The PeopleSoft system offers the following ways to sign in to a PeopleSoft session without seeing the sign-in page:

- Direct access.

  You can configure the web server to enable users to access the PeopleSoft system without having to sign in. In these cases, 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.

- When bypassing the sign-in page, you can override the user profile language preference by explicitly setting the language within a direct query string using the parameter &languageCd=*target_language.*

- Single sign-in.

  You need to select a language preference only once during the single sign-in.

  All subsequent PeopleSoft sessions that are controlled by the single sign-in use the same language preference.

Related Topics

- Security Administration: Understanding Single Signon

Controlling the User Profile Language Preference

In addition to the session language preference that is selected at sign-in time and is valid for the current session only, the PeopleSoft Pure Internet Architecture also maintains a language preference on each user's profile. This language preference is used to determine the user's preferred language when that user is not signed into the PeopleSoft system. For example, when PeopleSoft Process Scheduler runs a Structured Query Report (SQR) program, or if the workflow system sends an email to the user, the user language preference is used.

In these cases, the user might not be signed in to the PeopleSoft system, or another user might have initiated the transaction, so there is no session language that can be used. For example, a manager might be sent an mail by the workflow system requesting approval for a purchase requisition.

To set the language preference for a user, use the User Profiles page. To change your own language preference, use the simplified General Profile Information page.

To set the user profile language preference for another user:

1. Select **PeopleTools**, and then **Security**, and then **User Profiles**, and then **User Profiles.**

   The User Profile search page appears.

2. Use the standard search method to choose a user ID, and click the Search button.

   The User Profiles - General page appears.

3. From the **Language** drop-down list box, select a language.

4. Click the Save button.

To set your user profile language preference:

1. From the menu pagelet, select **My System Profile.**

   The General Profile Information page appears.

Copyright © 1988, 2026, Oracle and/or its affiliates.

2. In the Personalization section, choose a language in the **My preferred language for reports and email is** drop-down list box.
3. Click the Save button.

Related Topics

- Understanding Data Editing in Related Language Tables and Base Tables
  When a user changes data on a language-sensitive page, the system responds by changing the contents of the base table, the related language table, or both, depending on the language of the user and which field was modified. The following table explains at the database table level the actions that are performed when you work with language-sensitive tables:

Determining the Language of Help Documentation

If a user accesses a document from a PeopleSoft application by clicking **Help** in a PeopleSoft page or by pressing the **F1** key from Application Designer, the system opens the version of the document that corresponds to the session language. If the system does not find a version of the documentation in the appropriate language, it brings up the English documentation, if available.

Applying Browser Language Preferences

This section discusses configuring the browser language preferences and the sign-in language.

Understanding Browser Language Preferences

Web browsers enable users to specify a list of preferred languages for web content. PeopleTools uses the first language listed in the browser's preferred language list to determine the default language of the PeopleSoft sign-in page. If the user's preferred language is not available, the default sign-in page is displayed in U.S. English (en-US).

Setting Browser Language Preferences

In Microsoft Internet Explorer, the preferred language list is located in the Language Preference dialog box. You can add new languages using the Add dialog box.

The representation of languages in the browser's preferred language list uses two- to five-character ISO standard locale codes made up of a language and country portion. For example, *es* represents Spanish and *ar* represents Argentina. A two-letter code represents only a language, without reference to a specific country or territory.

Other browsers have a similar dialog box that enables you to create an ordered list of language preferences.

Using Browser Language Settings to Determine the Sign-in Language

For PeopleTools to display the sign-in page in the user's preferred language as determined from browser settings, the ISO codes used by the browser are mapped to three-letter PeopleSoft language codes. The PeopleSoft system reads the PSLANGUAGES table to perform this mapping.

As delivered, the PeopleSoft system includes the following mappings:

Copyright © 1988, 2026, Oracle and/or its affiliates.

PeopleSoft Language Code	Description	Corresponding ISO Locales
ARA	Arabic	ar
BUL	Bulgarian	bg
CFR	Canadian French	fr_CA
CRO	Croatian	hr
CZE	Czech	cs
DAN	Danish	da
DUT	Dutch	nl
ENG	English	en
ESP	Spanish	es
FIN	Finnish	fi
FRA	French	fr
GER	German	de
GRK	Greek	el
HEB	Hebrew	he
HUN	Hungarian	hu
ITA	Italian	it
JPN	Japanese	ja
KOR	Korean	ko
NOR	Norweigan (Bokmål)	no
POL	Polish	pl
POR	Portuguese	pt
ROM	Romanian	ro
RUS	Russian	ru
SER	Serbian	sr
SLK	Slovak	sk
SLV	Slovenian	sl
SVE	Swedish	sv
THA	Thai	th
TUR	Turkish	tr
UKE	UK English	en_GB
ZHS	Simplified Chinese	zh_CN
ZHT	Traditional Chinese	zh_HK, zh_TW

The two-part ISO locale 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, including France (FR), Belgium (BE), Canada (CA), and Switzerland (CH). The full locale code consists of a lowercase language code, an underscore, and an uppercase country code. So the code for Canadian French is fr_CA.

In many cases, the country distinction between languages is not relevant, or the PeopleSoft system does not deliver a country-specific translation. In these cases, many individual ISO locale codes may be mapped to a single PeopleSoft language code. For example, the PeopleSoft system delivers a Canadian French translation with the PeopleSoft language code of CFR. Therefore, the ISO locale of fr_CA is mapped to the PeopleSoft language code of

Copyright © 1988, 2026, Oracle and/or its affiliates.

CFR. However, the PeopleSoft system does not provide a Belgian French or Swiss French translation, so the fr_BE and fr_CH ISO locales fall back to fr, and are both mapped to the FRA PeopleSoft language code.

Setting Up Language Preferences for Microsoft Windows-Based PeopleTools

This section provides an overview of language preferences in PeopleTools for Microsoft Windows and discusses how to control PeopleSoft Configuration Manager language settings.

Understanding Language Preferences for Microsoft Windows-Based PeopleTools

In the Microsoft Windows-based PeopleTools such as Application Designer, the language setting within PeopleSoft Configuration Manager controls the language in which the user views almost all language-sensitive data and objects in the application.

Just like in the PeopleSoft Pure Internet Architecture, if the language selected in PeopleSoft Configuration Manager is the base language, then all language-sensitive data, messages, and objects appear in the base language. If the language in PeopleSoft Configuration Manager 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.

Related Topics

- Understanding Related Language Tables
  This section discusses related language tables.

Controlling PeopleSoft Configuration Manager Language Settings

The language setting in PeopleSoft Configuration Manager is maintained on each PeopleTools development environment workstation. The setting determines the language in which the user views and maintains almost all language-sensitive data and objects in the Microsoft Windows-based PeopleTools.

The following graphic shows PeopleSoft Configuration Manager with the Language field set to *ESP-Spanish*.

To set the PeopleSoft Configuration Manager language:

1. Start PeopleSoft Configuration Manager.
2. Select the Display tab.
3. Select a language in the **Language** field.
4. Click **OK.**

Setting Up Locale-Based Formatting for the PeopleSoft Pure Internet Architecture

This section discusses setting up locale-based formatting.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Related Topics

- Understanding Currency-Specific Settings
  PeopleTools can format a currency amount that appears on a page or a report based on both your numerical format preferences and the currency that the amount represents.
- Security Administration: Understanding System Personalizations

Understanding Locale-Based Formatting

The locale-based personalization architecture in the PeopleSoft Pure Internet Architecture provides a flexible framework for formatting locale-sensitive data. Three levels of personalization settings drive the formatting of locale-based data in the PeopleSoft Pure Internet Architecture:

- User-specific (My Personalizations page).
- Locale-specific (from the browser).
- System-wide.

These levels ensure that each user can specify how to format locale-based data, such as numbers, dates and times, while maintaining an intelligent set of default formats for users who don't set specific preferences.

When determining how to format these types of data, PeopleTools checks for formatting preferences at each of these levels in the order of user-specific, locale-specific, and finally system-wide.

For example, a first-time user may sign in to the PeopleSoft Pure Internet Architecture as a French user and having *fr* set in the browser as the preferred language. Because the user is a first-time user of the system, PeopleTools does not have any personalization information specific to that user. Next, PeopleTools checks the personalizations tables to see if a locale-specific formatting setting exists for the browser preference code of fr. If locale-specific settings for fr users are found, the system formats all numbers, dates, and times for that session using the settings defined by the system administrator for French users.

However, it is possible that the user, even though signed in using the French language and having fr set as the browser preferred language, might prefer another type of formatting for number separators. To register a user-specific preference, the user opens the My Preferences page, accesses personalization options for regional settings, and overrides the **Digit Group Separator** field value. Whenever that user accesses the PeopleSoft Pure Internet Architecture, the decimal separator is that custom value, regardless of session language and browser language preference. All other preferences are read from the locale-specific personalizations mapped to the browser settings.

If a user signs in to the PeopleSoft Pure Internet Architecture using a language for which locale-specific personalizations have not been set up by the system administrator, PeopleTools uses system-wide defaults.

When running a batch program such as an Application Engine program, the browser locale defaults are not available, so there are only two levels of settings: user preferences and system defaults. Therefore, if the user running the Application Engine program has their formatting preferences set (from My Preferences), then the output of that program will be consistent with other outputs for that user.

Using the personalizations system in PeopleTools, the system administrator can restrict which personalization options are driven by a locale and which can be overridden. In certain organizations where a corporate-wide policy exists for formatting numbers, dates, or times, the system administrator might disable the locale sensitivity or user overrides for locale-sensitive

Copyright © 1988, 2026, Oracle and/or its affiliates.

data formats. In this case, the system-wide defaults take effect for each personalization option disabled by the system administrator.

> **Note**
>
> If user-specific formatting is not set, locale-specific formatting will be used, even if the Locale Based check box on the Define Personalizations page is not selected. If locale-specific formatting is not set, system-wide formatting defined on the Define Personalizations page is used.

Defining User-Specific, Locale-Based Formatting

Use the Regional Settings preferences on the My Preferences – General Settings page to specify how PeopleTools should format each field that contains numbers, dates, and times. Each individual formatting element can be overridden on a user-by-user basis.

To override any of these values, enter the appropriate value or select one from the drop-down list box in the **Override Value** column. For example, if you prefer to use a dash (-) instead of a slash (/) to separate the components of dates, enter a dash character in the **Override Value** column for the **Date Separator** personalization option. Values maintained here are specific to your user ID.

If you leave any of these override values blank, they will be evaluated next time you sign in based on your browser language preference, and if locale-specific preferences aren't found for your language, they are automatically populated from the system-wide values, as shown in the **Default Value** column on this page.

> **Note**
>
> If auto-recognize Gregorian dates is set to *Yes* (the default) and the calendar is set to a non-Gregorian calendar, any dates entered in date fields that fall in the range of the Gregorian calendar will be assumed to be Gregorian and will be converted to the specified calendar's dates.

Related Topics

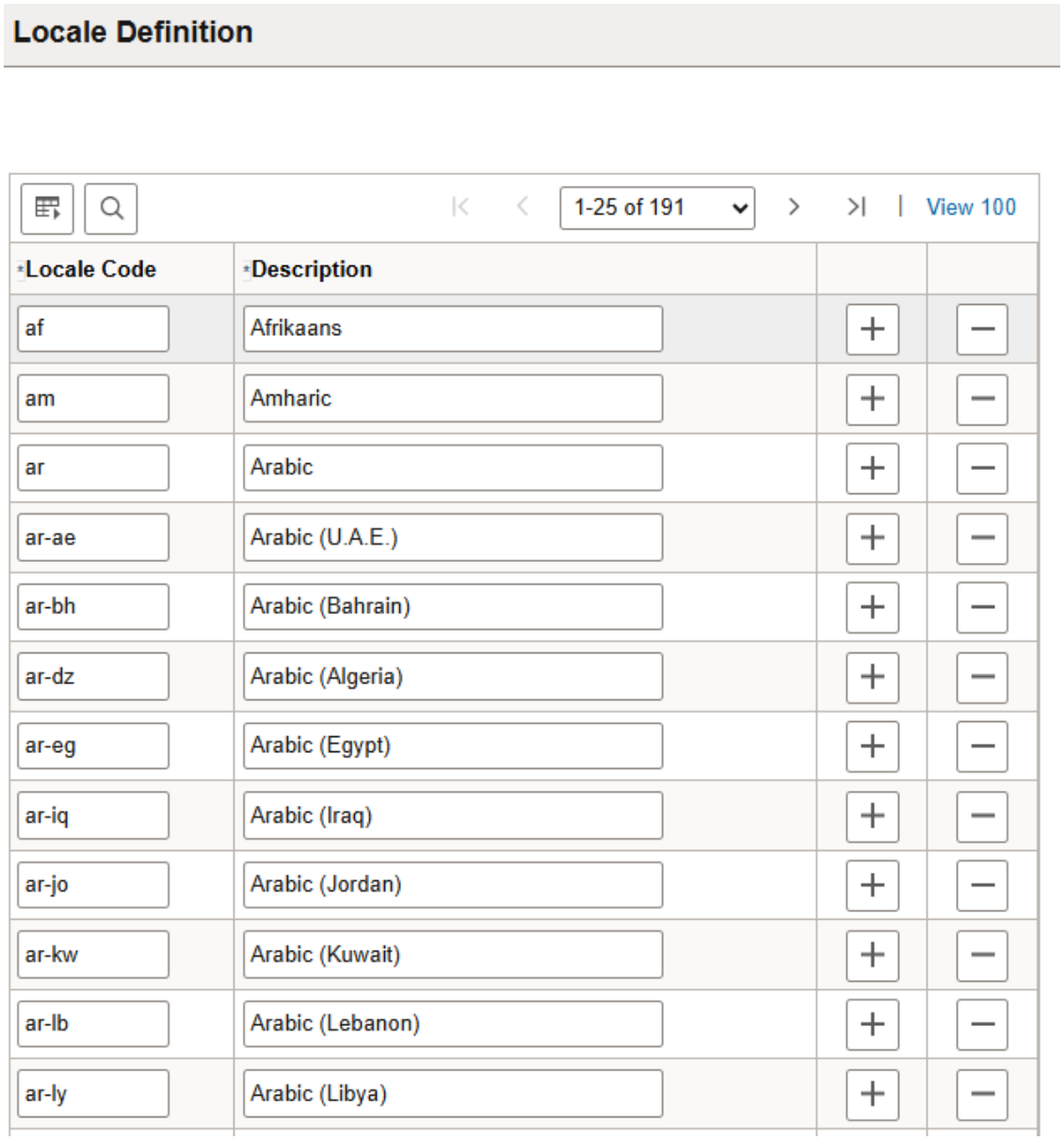
- Security Administration: Understanding System Personalizations
- Security Administration: Creating Custom My Preferences Options

Defining Locales for Locale-Specific Formatting

Access the Locale Definition page (**PeopleTools**, and then **Personalization**, and then **Personalization Locales**).

This example illustrates the fields and controls on the Locale Definition page.

Copyright © 1988, 2026, Oracle and/or its affiliates.



Use this page to add locales that can have defaults specified on the Locale Defaults page. These locales are different from PeopleSoft language codes; they typically contain both language and country information. When formatting data fields, both these elements are typically needed, given that many languages are spoken in different countries, with each country often following different formatting conventions. Having only information on the user's language is not sufficient to determine a default level of locale-sensitive data formatting.

For example, English is spoken in several countries that differ on date formatting. In the U.S., dates are typically formatted with the month first, whereas in the United Kingdom, they are typically formatted with the day first.

The session language code cannot be used to determine the specific locale for a session, due to its lack of country information. Instead, the browser's language preference list is read, and the top (the most preferred) locale in that list is used to look up locale-specific formats in the Personalizations Locale Definitions page. Therefore, you should ensure that each browser locale used by your system's users appears in the Locale Definition page.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Defining Locale-Specific Formatting

Access the Locale Defaults page (**PeopleTools**, and then **Personalization**, and then **Personalization Local Defaults**).

This example illustrates the fields and controls on the Local Defaults page.

Locale Defaults

1-25 of 923 View 100

*Option Category Level	*User Option	*Locale Code	Description	*Override Value
Tools	Afternoon designator (PM, pm	af	Afrikaans	nm
Tools	Afternoon designator (PM, pm	en	English	PM
Tools	Afternoon designator (PM, pm	en-au	English (Australia)	PM
Tools	Afternoon designator (PM, pm	en-bz	English (Belize)	PM
Tools	Afternoon designator (PM, pm	en-ca	English (Canada)	PM
Tools	Afternoon designator (PM, pm	en-nz	English (New Zealand)	PM
Tools	Afternoon designator (PM, pm	en-ph	English (Philippines)	PM
Tools	Afternoon designator (PM, pm	en-tt	English (Trinidad)	PM

Use this page to assign default values for personalization options based on locale code.

To more easily view defaults for a particular locale, click the Locale Code column heading to sort the data on this page by the locale code.

> **Note**
>
> For some international locales, a space character is used as a digit group separator (for example, thousands). However, the PeopleSoft Pure Internet Architecture interprets a single space in a field as a null value. To use the space character as a digit group separator, define the override value as a space between single quotation marks. This personalization can be defined at the system-wide, locale-default, or individual user level. This special treatment of the space character currently only takes effect for the digit group separator user option.

Applying System-Wide Default Formatting

System-wide formatting is applied for the session only if a particular personalization option cannot be found that is specific to the user or the browser's locale. System-wide defaults are defined using the Define Personalizations page. To apply system-wide default formatting, select **PeopleTools**, and then **Personalization**, and then **Personalization Options** and click the Format tab.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Related Topics

- Security Administration: Understanding System Personalizations

Setting Up Locale-Based Formatting for Microsoft Windows-based PeopleTools

The Microsoft Windows Regional Options dialog box (accessed through the Microsoft Windows control panel) enables you to choose the format for the display of numbers, dates, times, and currencies while operating in Application Designer and other Microsoft Windows components of PeopleTools.

> **Note**
>
> Regional Options that you set up in Microsoft Windows do not affect the display or maintenance of data in the PeopleSoft Pure Internet Architecture.
>
> Currency formatting is controlled by PeopleTools and is independent of the Regional Options dialog box.

Related Topics

- Understanding Currency-Specific Settings
  PeopleTools can format a currency amount that appears on a page or a report based on both your numerical format preferences and the currency that the amount represents.

Copyright © 1988, 2026, Oracle and/or its affiliates.

3

Setting and Maintaining Time Zones

Understanding Time Zones

PeopleSoft applications store times based on a system-wide base time zone and support the display of times relative to a user's local time zone or relative to the time zone in which a transaction is entered (called a specified time zone).

For example, if the base time zone is U.S. Pacific time (PST), a time entered as 10 a.m. in U.S. eastern time (EST) is stored as 7 a.m. because of the time difference between the two locations. This allows for easy comparison and manipulation of times so that PeopleCode developers do not have to worry about time zone differences. Additionally, all batch processes, such as structured query reports and COBOL, operate in the base time zone of the system.

However, although the time is stored as 7 a.m., it can still appear to the user as 10 a.m. EST, or 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, the PeopleSoft system delivers an initial global list of time zones, including information about daylight saving time (DST) observances. When a time zone begins or ends in daylight saving 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 how users see times.

> **Note**
>
> If your base time zone is one that observes Daylight Saving Time (DST) , you will get an ambiguous time listed during the changeover hour in the Fall. For example, in US Pacific Time, there are two 1:30 AM, first in PDT and then in PST. Since 1:30 AM PDT or PST is not marked in the timestamp object, if transactions during this time rely on that difference, it should be marked separately whether DST is in effect or not. Choosing a base time zone that does not observe DST, such as Universal Coordinated Time (UTC), would eliminate this ambiguity especially for the customers with multiple branches with different time zones.

Base Time Zone Details

This time zone 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 meta-SQL %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 for your system to function correctly.

The base time zone is also used for the following:

- All effective-date processing.

Copyright © 1988, 2026, Oracle and/or its affiliates.

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.

Specified Time Zone Details

Specified time zones are useful for applications where users must see the same time and time zone as those who enter transactions, such as in call processing systems. To implement a specified time zone field, your record must contain a time zone control field.

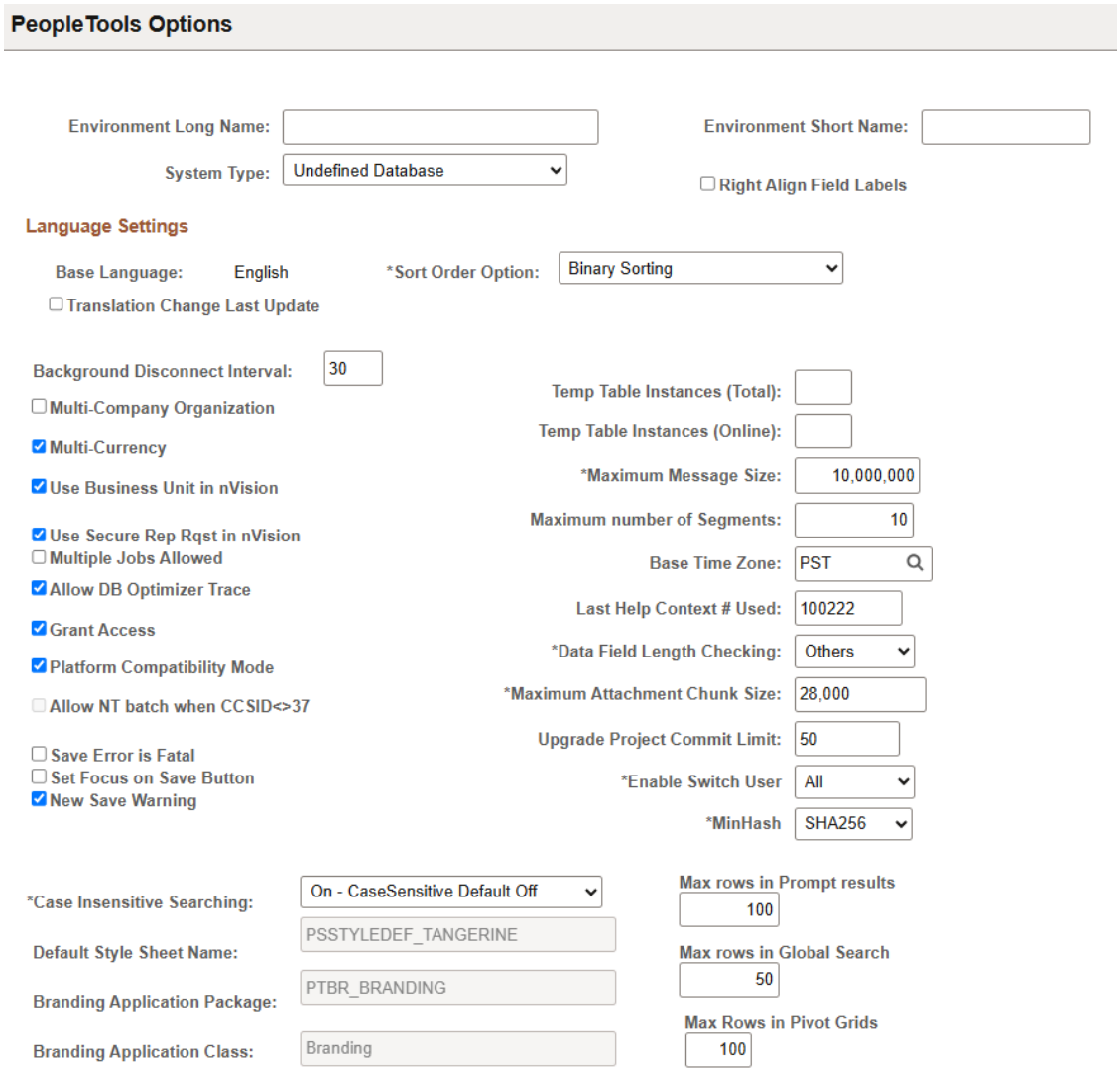
> **Note**
>
> When client DateTime values are to be saved without converting to another time zone, use the Specified Time Zone. This way, there is no need to convert to the base time zone when saving to the database and convert back to the client time zone when obtaining the value from database. Using the Specified Time Zone avoids the complexity of any shifts in Daylight Saving Time if, that is used in the base time zone.

Setting the Base Time Zone

Select **PeopleTools**, and then **Utilities**, and then **Administration**, and then **PeopleTools Options** to access the PeopleTools Options page.

This example illustrates the fields and controls on the PeopleTools Options page.

Copyright © 1988, 2026, Oracle and/or its affiliates.



The base time zone that you define for PeopleTools must match the time zone used by your database. Any discrepancy between the two leads to inaccurate time processing.

To set the base time zone:

1. Select a time zone from the **Base Time Zone** field.

   Values are maintained in the Time Zones component (TIMEZONEPNLGRP).

   See Maintaining Time Zones.

2. Click the **Save** button.

> **Note**
>
> 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 is made to existing time data in your database.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Maintaining Time Zones

This section discusses how to define and maintain time zones.

Understanding Time Zone Maintenance

Oracle delivers a subset of time zone data including information about daylight saving time observances in the time zone table. Typically, you will access time zone data through the PSTIMEZONE view, which shows current entries from the PSTIMEZONEDEFN table. The PSTIMEZONEDEFN table has an effective datetime field, allowing historical time zone definitions to be stored. The Oracle-provided subset of time zone data is consistent with current law at the time of the PeopleTools GA release. Oracle does not provide patches or updates to this time zone data once it is released—for example, to bring it up to date with recently enacted legislation.

You are, therefore, responsible for three tasks:

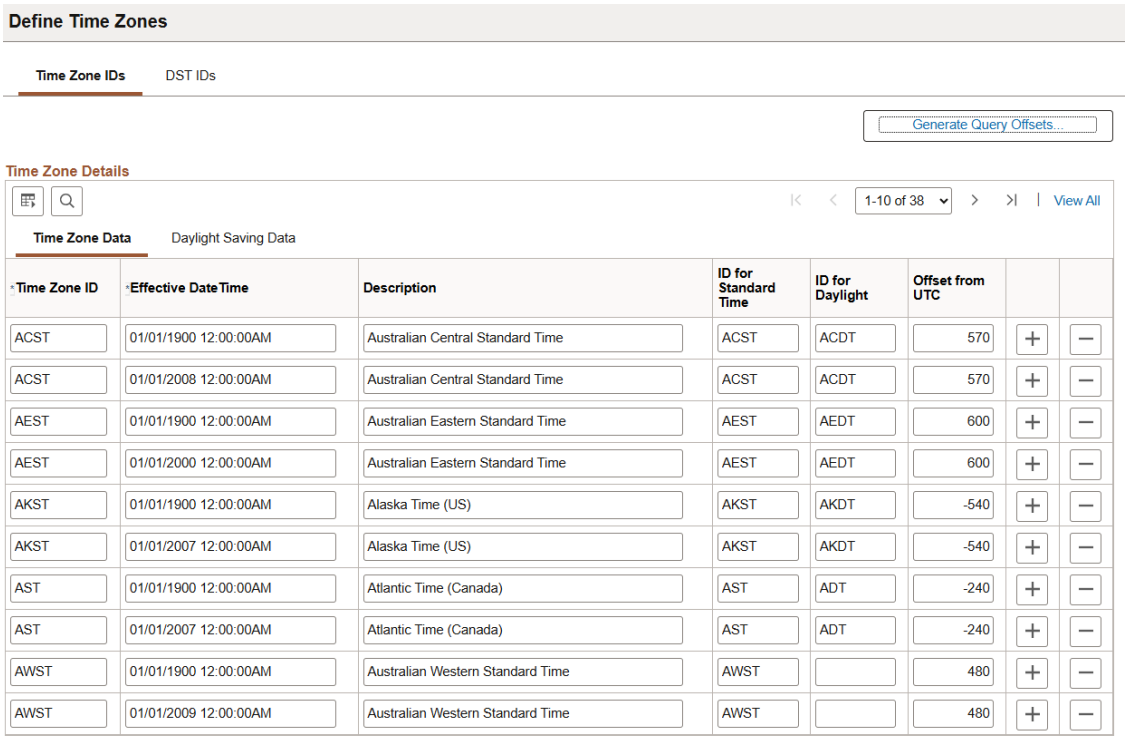
- Create any time zone definitions required by your global operations but not included in the Oracle-provided data.
- Ensure that each time zone has an entry with a PTEFFDTTM of 1900-01-01 00:00:00, plus entries for any regulatory changes you want to track, in the PSTIMEZONEDEFN table.
- Update time zone definitions whenever local regulations change the definition for a time zone; for example, the daylight saving time start or end date.
- Ensure that each time zone ID you added has an entry in the Time Zone Mapping Table to map to an IANA time zone.

Defining Time Zones

Access the Time Zone IDs page (**PeopleTools**, and then **Utilities**, and then **International**, and then **Define Time Zones**).

This example illustrates the fields and controls on the Time Zone IDs page: Time Zone Data tab. You can find definitions for the fields and controls later on this page.

Copyright © 1988, 2026, Oracle and/or its affiliates.



*Time Zone ID	*Effective DateTime	Description	ID for Standard Time	ID for Daylight	Offset from UTC
ACST	01/01/1900 12:00:00AM	Australian Central Standard Time	ACST	ACDT	570
ACST	01/01/2008 12:00:00AM	Australian Central Standard Time	ACST	ACDT	570
AEST	01/01/1900 12:00:00AM	Australian Eastern Standard Time	AEST	AEDT	600
AEST	01/01/2000 12:00:00AM	Australian Eastern Standard Time	AEST	AEDT	600
AKST	01/01/1900 12:00:00AM	Alaska Time (US)	AKST	AKDT	-540
AKST	01/01/2007 12:00:00AM	Alaska Time (US)	AKST	AKDT	-540
AST	01/01/1900 12:00:00AM	Atlantic Time (Canada)	AST	ADT	-240
AST	01/01/2007 12:00:00AM	Atlantic Time (Canada)	AST	ADT	-240
AWST	01/01/1900 12:00:00AM	Australian Western Standard Time	AWST		480
AWST	01/01/2009 12:00:00AM	Australian Western Standard Time	AWST		480

Time Zone Data Tab

Field or Control	Description
Time Zone ID	Displays the name (or ID) for the time zone.
Effective DateTime	Displays the effective date and time for each time zone definition.
Description	Displays the description for the time zone ID.
ID for Standard Time	Displays the name for the time zone during standard time.
ID for DST	Displays the name for the time zone during daylight saving time.
Offset from UTC	Displays the number of minutes the time is offset from universal coordinated time (UTC, sometimes known as Greenwich mean time or GMT). A positive offset indicates a time zone east of UTC; a negative offset indicates a time zone west of UTC. For example, the time zone for India, which is 5 and a half hours east of UTC, has an offset of +330. U.S. pacific standard time (PST), which is 8 hours west of UTC, has an offset of -480. **Note:** There are two business use cases of GMT. GMT as used in the UK is synonymous with WET. It does observe DST, when the label is BST during summer time. GMT as used internationally is synonymous with UTC. It does not observe DST, and does not change the label.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Field or Control	Description
Generate Query Offsets	Click to access the open the Time Zone Offset Generation page. On this page, you can use the information from the Time Zone IDs page to populate the PSTZOFFSET table with offsets for all the time zones and their daylight saving 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 PeopleSoft Query. The PeopleSoft system 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, regenerate the offsets anytime the underlying time zone data is changed.

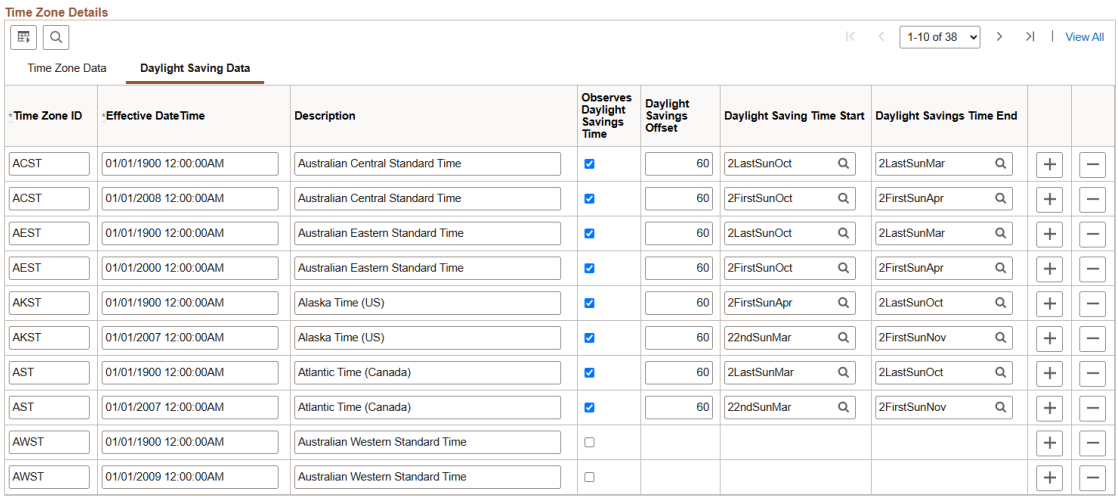
Daylight Saving Data Tab

Select the Daylight Saving Data tab to view daylight saving offset information and the IDs specified for daylight saving time start and daylight saving time end.

> **Note**
>
> When a timezone discontinues observing DST, special handling is required. This statement will audit for issues related to this type of change.
>
> select count(*),timezone from (select distinct timezone,observedst from pstimezonedefn) group by timezone having count(*) > 1;
>
> Any rows that are returned have both Y and N defined, which can cause issues when generating query offsets.
>
> In this case, all rows OBSERVEDST should be Y, and timezones which have discontinued DST should have a DSTOFFSET of 0, and should have TIMEZONEDST label matching TIMEZONESTD label.

This example illustrates the fields and controls on the Time Zone IDs page: Daylight Saving Data tab. You can find definitions for the fields and controls later on this page.

Copyright © 1988, 2026, Oracle and/or its affiliates.



Field or Control	Description
Observes DST	Indicates whether the time zone observes daylight saving times. If this check box is cleared, the other DST values have no effect.
DST Offset	Displays the number of minutes by which the time is offset during daylight saving time.
DST Start ID	Displays the ID for when daylight saving time begins. This field prompts against values from the PSDSTTIME table (the DST IDs page).
DST End ID	Displays the ID for when daylight saving time ends. This field prompts against values from the PSDSTTIME table (the DST IDs page).

Generating Time Zone Offsets

Access the Time Zone Offset Generation page (click the Generate Query Offsets button on the Time Zone IDs page).

To generate query offsets:

1. Enter a start date and an end date.

   Select a reasonable range of dates. With the time zones delivered with the PeopleSoft system, more than 100 rows are generated for each year.

2. Click **OK.**

   The data is generated for the table. Any existing data for the same range of dates is overwritten.

> **Note**
>
> If you receive an error about missing daylight saving time IDs, the error can be corrected by adding entries to the PSDSTTIME table on the DST IDs page.

For example, the following data from the PSTZOFFSET table represents one time zone, EST, for a two-year period, from January 1, 1999, to January 1, 2001. It is based on a system with

Copyright © 1988, 2026, Oracle and/or its affiliates.

PST as the base time zone. When you generate query offsets, the PSTZOFFSET table contains similar data for all the time zones defined in the PSTIMEZONE table.

TIMEZONE	STARTDATETIME	ENDDATETIME	BASEOFFSET	TIMEZONELABEL
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

The first row of data shows that from January 1, 1999, at midnight, until April 3, 1999, 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.*

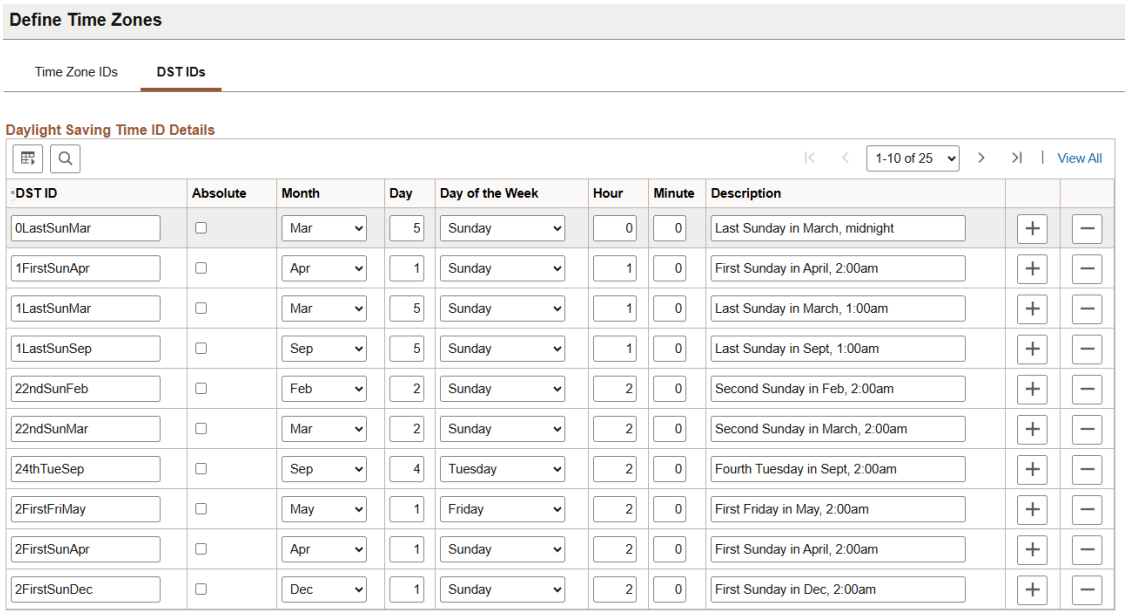
The second row shows that from April 3, 1999, at 11 p.m., until April 4, 1999, at 2 a.m., there is a three-hour time period during which EST is 240 minutes ahead of PST. This is because EST has changed to daylight saving time, but PST hasn't changed yet. The label for EST during this period is *EDT,* for eastern daylight time.

Defining DST IDs

Access the DST IDs page (**PeopleTools**, and then **Utilities**, and then **International**, and then **Define Time Zones**, and then **DST IDs**).

This example illustrates the fields and controls on the DST IDs page. You can find definitions for the fields and controls later on this page.

Copyright © 1988, 2026, Oracle and/or its affiliates.



Field or Control	Description
DST ID	Displays a unique identifier for the combination of date and daylight saving start time. For example, *22ndSunFeb* represents the second Sunday in February at 2:00 am.
Absolute	Clear this check box to indicate a relative date, such as the last Sunday in October.
Month	Displays the month.
Day	For absolute dates, displays the day of the month. For relative dates, the numbers *1*, *2*, *3*, and *4* represent the first, second, third, and fourth occurrence of the day of the week—for example, to describe the second Sunday in February, enter *2*. Use the number *5* for a relative date to indicate the last occurrence of the specified day of the week, even if that day of the week occurs only four times in the month.
Hour	Displays the hour at which the time switch occurs.
Minute	Displays the minute at which the time switch occurs.
Description	Provides a description of the DST ID.

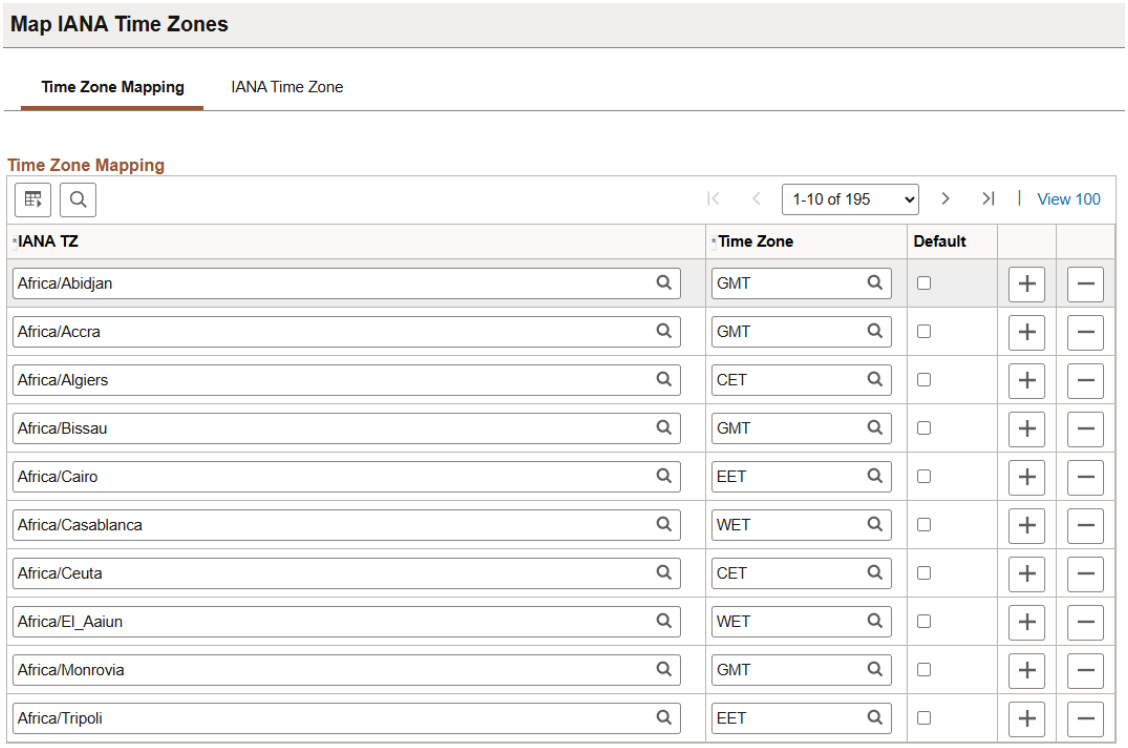
Mapping Time Zone

Access the Time Zone Mapping page (**PeopleTools**, and then **Utilities**, and then **International**, and then **Map IANA Time Zones**).

PeopleSoft has a mapping table to the industry standard IANA time zone IDs that is managed from the Time Zone Mapping page.

Copyright © 1988, 2026, Oracle and/or its affiliates.

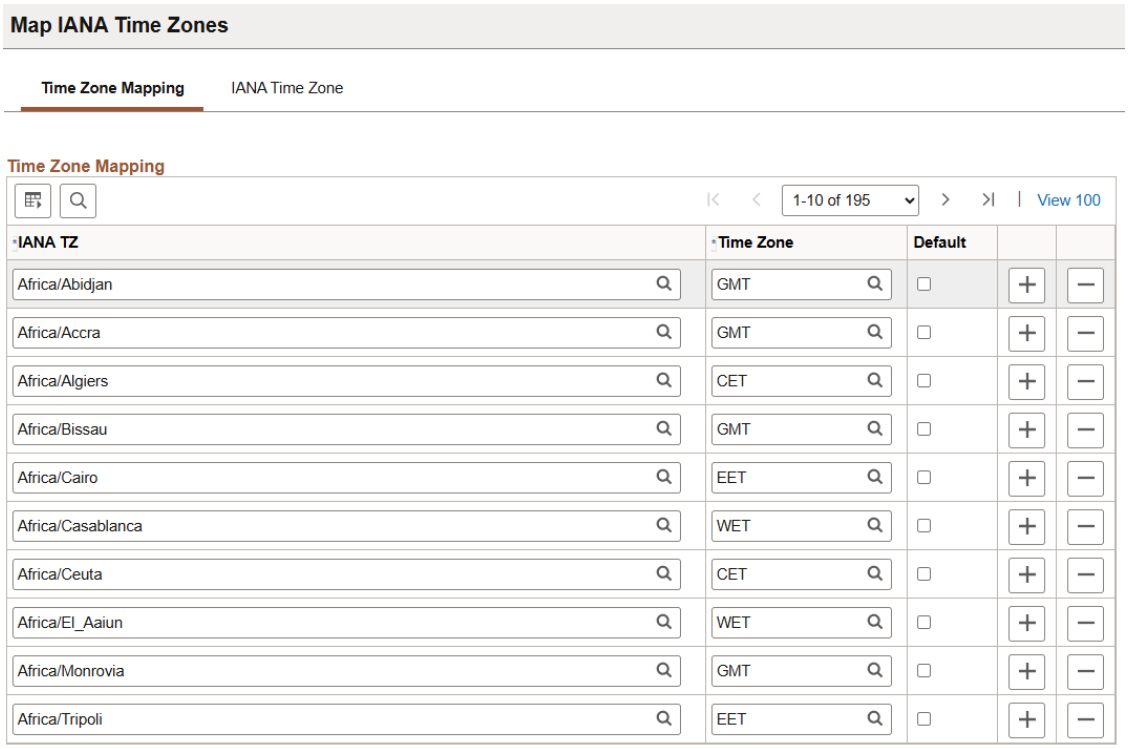
This example illustrates the fields and controls on the Time Zone Mapping Tab in the Time Zone Mapping Page. You can find definitions for the fields and controls later on this page.



Term	Definition
IANA TZ	IANA TZ: Specify the IANA time zone in the format from the tzdata file. This value should match the IANA time zone detected by the JVM running on the application server or process scheduler server. **Note:** The IANA time zone is listed in the server logs as *Detected IANA timezone*.
Time Zone	Specify the PeopleSoft time zone ID this IANA time zone maps to.
Default	Set this to indicate that this is the default for reverse mapping from PeopleSoft time zone ID to IANA time zone ID. This reverse mapping is used in XMLP. See BI Publisher for PeopleSoft: Using Time Zones in BI Publisher Reports

This example illustrates the fields and controls on the IANA Time Zone Tab in the Time Zone Mapping Page. You can find definitions for the fields and controls later on this page.

Copyright © 1988, 2026, Oracle and/or its affiliates.



Term	Definition
IANA Time Zone	The time zone listed in tzdata. When the application server or process scheduler starts, it will be running in one of these time zones. These are the lookup values for the Time Zone Mapping page.

Base Time Zone and Server Time Zone

The base time zone, database server time zone, and application server time zone should all match. This is the reference time zone where all data is stored. The process scheduler time zone can be set to a time zone which is often the same as the base time zone, but can also be set to any local time zone.

When the application server and process scheduler server start up, they map the IANA time zone, as reported by the JVM, to a PeopleSoft time zone ID. This mapping is used by the Query Export to Excel function, and the reverse mapping is used by XMLP.

Tables involved in this mapping are:

- PSTIMEZONEDEFN: Defines the PeopleSoft time zone IDs.
- PSDSTTIME: Defines the day and time when DST rules are in effect.
- PSTIMEZONEIANA: Defines the IANA time zones.
- PSTIMEZONEMAP: Maps between IANA and PeopleSoft time zone definitions.

The IANA time zone mapping is used in Query, when exporting queries with date or time related fields to Excel. To show the IANA time zone in the server logs, set the `LogFence = 5` in psappsrv.cfg or psprcs.cfg. Then, in Query Manager or Query Viewer, export a query having a date, time, or datetime field to Excel. The log will show the IANA time zone detected on that server, and the PeopleSoft time zone it was mapped to.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Adding DST Rules

Example 3-1 Adding DST Rules: PST Example

The delivered DST rules for the PST time zone include effective dates for several time ranges. A more complete list for PST would be the following:

PTEFFDTTM	OBSERVEDST	DSTOFFSET	DSTSTART	DSTEND
1900-01-01	N	60		
1918-01-01	Y	60	2LastSunMar	2LastSunOct
1920-01-01	Y	0	2LastSunMar	2LastSunOct
1942-01-01	Y	60	22ndMonFeb	12LastThuDec
1943-01-01	Y	60	0FirstFriJan	12LastFriDec
1944-01-01	Y	60	0FirstSatJan	12LastSunDec
1945-01-01	Y	60	0FirstMonJan	2LastSunSep
1946-01-01	Y	0	2LastSunMar	2LastSunOct
1948-01-01	Y	60	22ndSunMar	12LastFriDec
1949-01-01	Y	60	0FirstSatJan	2FirstSatJan
1949-01-01	Y	0	2FirstSatJan	2LastSunSep
1950-01-01	Y	60	2LastSunApr	2LastSunSep
1962-01-01	Y	60	2LastSunApr	2LastSunOct
1974-01-01	Y	60	2FirstSunJan	2LastSunOct
1975-01-01	Y	60	2LastSunFeb	2LastSunApr
1976-01-01	Y	60	2LastSunApr	2LastSunOct
1987-01-01	Y	60	2FirstSunApr	2LastSunOct
2007-01-01	Y	60	22ndSunMar	2FirstSunNov

In the PSTIMEZONEDEFN table, there must be a placeholder row for 1900-01-01, even if DST was not observed then. Additional rows are also needed as the DST rules change over the years, but you only need the DST definitions for the date ranges that you require for business reasons.

To add the DST rules for additional time ranges, check the tzdata file to see which Rules were defined that apply to the given time zone, and derive the PeopleSoft PSDSTTIME entries and PSTIMEZONEDEFN entries you will need. Then use the time zone pages to add the data.

For more information on time zone and the latest tzdata file, see the documentation for IANA.

The tzdata file has a Rule for when Daylight Saving Time comes into effect, and a separate Rule for when Daylight Saving Time goes out of effect. The PeopleSoft PSTIMEZONEDEFN table has one effective dated row showing both the DST start and the DST end. So in general, two lines of tzdata Rules will be represented in one PSTIMEZONEDEFN row.

Example 3-2 Adding DST Rules: IST Example

The IANA naming standard for time zones is based on a location. For example, Asia/Kolkata, and the typical time zone abbreviation used locally is IST for India Standard Time. But IST is also the abbreviation used locally for Israel Standard Time and Irish Standard Time. If your business requires all three of those time zones, only one of them can have a PeopleSoft time

Copyright © 1988, 2026, Oracle and/or its affiliates.

zone ID of IST, and the other two will need to have alternative IDs. In this example, IST is only needed for India Standard Time.

India Standard Time does not observe Daylight Saving Time, so it will not need any entries in PSDSTOFFSET, but it will need at least the definition with effective date of 1900-01-01 in PSTIMEZONEDEFN. It will also need entries in PSTIMEZONEMAP for any IANA time zones that the new IST PeopleSoft time zone should map to. In this example, there will be a new row in PSTIMEZONEMAP mapping from Asia/Kolkata to IST, and the column Default will be checked, since the reverse mapping for IST should point to Asia/Kolkata.

Updating Time Zone Definitions to Comply with Legislative and Other Changes

From time to time, legislative changes can impact the definitions of time zones, such as whether a time zone observes daylight saving time and the start or end date for daylight saving time observance. The PeopleSoft system provides updated time zone definitions in the next release following such a legislative change, and new customers as of that release receive the updated time zone data. However, existing customers upgrading to that new release do not receive the benefit of updated time zone data because customer time zone data is not changed during upgrade. In these cases, you can manually modify your time zone definitions to comply with changing rules.

As a specific example, the U.S. Energy Policy Act of 2005 changed time zone definitions in the U.S. and Canada. PeopleSoft-delivered data in the current PeopleTools release already reflects these changes. An appendix provides the procedure that you would use if you needed to update your time zone definitions to comply with this act.

See Understanding the U.S. Energy Policy Act of 2005.

Controlling Time Zone Display When Developing PeopleSoft Application Objects

This section provides an overview of time zone display settings and discusses steps to set time zones in record field and page field properties.

Understanding Time Zone Display Settings

Although times are always stored in the base time zone, you can choose whether a page displays a time in the base time or in another 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, the PeopleSoft system does not use the time component of these fields.

Two settings work together to control the time display on pages in the PeopleSoft Pure Internet Architecture:

- Record field properties.

  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, which determines the time zone.

Copyright © 1988, 2026, Oracle and/or its affiliates.

- Page field properties.

  Choose whether to automatically include the time zone abbreviation in a field that displays time or datetime values.

Use these guidelines to help decide how to implement time zone displays:

- If the record field properties option specifies that time appears in the base time zone, setting the page field properties to show the time zone helps users interpret the time shown on the page.
- If the record field properties option specifies that the time zone of a particular time or datetime field is determined by the time zone control field, then put that control field directly on the page, to enable the user to manually select the time zone for a field.
- If you put the time zone control field on the page where users are permitted to enter a time zone, disable the page field properties option that includes the time zone abbreviation in the time field.

  It is redundant to show the time zone in both places.
- If you do not permit users to enter a time zone manually, for example, if you use the %ClientTimeZone or %BaseTimeZone system variables to set the time zone, then display the time zone either by including the control field on the page or by setting the page field properties appropriately—not both.

In Application Designer, users can display time values in their local time zones or in the system's base time zone, regardless of how the record field properties and page field properties are set. By default, all Application Designer time and datetime fields appear in the base time zone, but each user can override this option by selecting **View**, and then **Time Display**, and then **Local Time** or **View**, and then **Time Display**, and then **Database Time.**

Understanding Time Zone Precedence

Time zone displays can be controlled at three levels.

- Field-specific time zone.

  Set field-specific time zone options in Record Field Properties in Application Designer.
- User-specific time zone.

  Set user-specific time zone options in My Personalization in the PeopleSoft Pure Internet Architecture. To access the options, select My Personalizations. Click the Personalize Option button for Regional Settings.
- System base time zone.

  Set System base time zone options using the PeopleTools Options page in the PeopleSoft Pure Internet Architecture. To access the page, select PeopleTools, Utilities, Administration, PeopleTools Options.

PeopleTools checks for time zone preferences in the order of field-specific, then user-specific, then system base time zone.

A display control can be set for a specific field. The time zone display is controlled for a field via the Time Zone settings in Record Field Properties. The time zone control field overrides the user-specific time zone.

For example: you have four time values on a page, one of which has a time zone control field. This field will display in the time zone specified in the control field. The other three time fields will display in the user-specific time zone if set, and otherwise in the system base time zone.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Users can specify a time zone preference using Use Local Timezone and Local Time Zone options in the PeopleSoft Pure Internet Architecture. To access time zone preference settings, select My Personalizations, Regional Settings.

The Use Local Timezone option means to display all times and datetimes in the time zone specified in the Local Time Zone option

Setting Time Zone Options and Related Date Fields in Record Field Properties

When a record includes a time or datetime field, choose whether the PeopleSoft system 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.

Understanding Related Date Fields

The purpose of a related date field is for accounting for daylight saving time. Related date fields only apply to time fields, not datetime fields. If the user has set their personalization to show times in their local time zone, then the runtime engine calculates the offset of the local time zone from the database time zone and applies that offset to the value before displaying it.

Similarly it does the reverse when converting a user entered time back to database time. To calculate the offset between time zones the system must know if daylight saving time is in effect, so you must provide the date. For a datetime field this is no problem, but for a time field the system needs only the current date. This can lead to some problems where someone enters data, then comes back to look at it at a later date and sees it has shifted by an hour because daylight saving time has gone into affect in the meantime. Related dates resolve this problem.

When there is a related date, that date value is used to calculate the offset between the local and database time zones, therefore the calculation will always be the same and the time value won't change depending on when you look at it. Furthermore the related date field will look at the time field that is related to it to determine if it needs to be offset.

For example, lets say the database time zone is pacific standard time (PST), and the local time zone is eastern standard time (EST). If you enter 1:00 a.m. into a time field and 01/01/2000 into the date field related to that time, the time will be converted to 10:00 p.m. PST internally and for saving to the database, and the date will be converted to 12/31/1999, because 1:00 a.m. EST is 10 p.m. PST of the previous day. The offset is minus three hours.

Setting the Time Zone to Control Display Times

To set the time zone that controls the display time:

1. In Application Designer, open the record, and open the Record Field Properties dialog box for the time field by double-clicking the field that displays the time.

   The Use tab appears.

2. In the **Time Zone** group box, set the time zone property.

   If you leave the **Specified Time Zone** check box clear, the time appears in the database's base time zone by default, unless a user specifies a local time zone in My Personalizations, Regional Options to display times in the user's local time zone.

   If you select the **Specified Time Zone** check box, the field displays the time according to the time zone that is specified in the **Time Zone Control Field** field. You must specify a time zone control field in this record.

Copyright © 1988, 2026, Oracle and/or its affiliates.

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 appear in the same time zone. If you want multiple time or datetime fields to appear 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 is 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. Select which related date field in the current record stores the calendar date to which this field should be adjusted.
4. 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 changing 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 by double-clicking the page control that displays the time.

   The Record tab appears.
2. Select the **Display Time Zone** box.

   When you select this box, the page displays the time zone in the same field as the time and resizes the edit box to allow space for up to 10 additional characters.

   This option has no affect on data entry. Users cannot override the time zone. Users who see the time in GMT must enter the time in GMT.

   If the field that is associated with this page control is not a time or datetime field, selecting the **Display Time Zone** box has no effect.
3. Click **OK.**

Using PeopleCode Time Functions

Three PeopleCode functions convert times to their base time zone equivalent: ConvertTimeToBase, ConvertDatetimeToBase and DayTimeToTimeZone. Another function, IsDaylightSavings, establishes whether daylight saving time is in effect on a particular date.

Additionally, you can use system variables %clienttimezone and %basetimezone in PeopleCode programs or as field default values. The %clienttimezone variable returns the current user's local time zone, and %basetimezone returns the base time zone of the database.

Related Topics

- PeopleCode Language Reference: AddToDate function

Copyright © 1988, 2026, Oracle and/or its affiliates.

- PeopleCode Language Reference: System Variables Reference

Copyright © 1988, 2026, Oracle and/or its affiliates.

4

Selecting and Configuring Character Sets

Understanding Character Sets

This section discusses character sets.

Character Sets

Before you install your PeopleSoft system, Oracle recommends that you choose an appropriate character set for PeopleSoft client workstations, web servers, application servers, and database servers, as well as for file attachment storage locations (that is, FTP sites, HTTP repositories, and database tables).

A *character set,* also known as a *code page,* is an ordered set of characters in which each character is mapped to a numeric index, called a *codepoint.* This codepoint stores character data in a computer system. Many hundreds of character sets exist. Some are international standards, maintained by the International Organization for Standardization (ISO), some are country-specific standards, and others 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 uses several different character sets.

Common Character Sets

Although there is general agreement on the content and arrangement of most character sets, especially those that are maintained by the ISO, many different names are used by vendors and software packages for similar or identical character sets. US-ASCII encodes the basic characters and symbols that are needed to write the English language. However, US-ASCII is limited to 127 characters and cannot represent many characters that are needed by Western European languages, such as French and German, let alone ideographic languages, such as Japanese and Chinese, in which each character represents a word or concept. Many character sets, however, include all US-ASCII characters in addition to their other characters.

The following table illustrates just a few common character sets that you are likely to encounter and some of the names that are used by different vendors to refer to them:

Copyright © 1988, 2026, Oracle and/or its affiliates.

Character Set	Description and Comments	Type	PeopleSoft and SQR Name	Oracle DBMS Name	Microsoft Windows Name
ISO 8859-1	Western European Latin-1. Contains characters that are required to represent Western European languages. However, does not include the euro symbol, the trademark (TM) symbol, or the *oe* ligature.	ISO	LATIN1 or ISO_8859-1	WE8ISO8859P1	CP28591
Microsoft Code Page 1252	Microsoft Code Page 1252 - Western European. Very similar to ISO 8859-1, except for the inclusion of additional characters. Includes the euro symbol, trademark (TM) symbol, and *oe* ligature, but using a different codepoint than ISO 8859-15.	Vendor (Microsoft)	CP1252	WE8MSWIN1252	CP1252
ISO 8859-2	Central/ Eastern European Latin-2. 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	EE8ISO8859P2	CP28592

Copyright © 1988, 2026, Oracle and/or its affiliates.

Character Set	Description and Comments	Type	PeopleSoft and SQR Name	Oracle DBMS Name	Microsoft Windows Name
ISO 8859-15	Western European extended Latin-9. Similar to ISO 8859-1, but contains the euro symbol, *oe* ligature, and several characters that are required for Icelandic.	ISO	LATIN9 or ISO_8859-15	WE8ISO8859P15	CP28605
Shift-JIS	Most common Japanese character set. Defines thousands of characters for writing Japanese.	Country (Japan)	SJIS	JA16SJIS or JA16SJISTILDE	CP932
IBM CCSID 37	IBM Coded Character Set ID 37. Western European Multilingual EBCDIC-based character set.	Vendor (IBM)	EBCDIC	WE8EBCDIC37	CP1140
GB18030	Chinese national character set	Country (China)	GB18030	GB18030	GB18030

Some of these character sets, such as ISO 8859-1 and IBM CCSID 37, require only one byte to represent each character. For example, in ISO 8859-1, the hexadecimal number 61 represents the lowercase Latin letter *a.* However, larger character sets, such as Shift-JIS, may require more than one byte to represent each character.

The Unicode Standard

The most important consideration when dealing with character sets across a system is ensuring that all characters that you plan to represent within the PeopleSoft system exist in the character set that is used by each component of the system.

For example, if you plan to maintain Japanese characters in employee names, you must ensure that:

- The character set that is used by the database system includes Japanese characters.
- Each external system feeding into or out of the PeopleSoft system expects data in a character set that includes Japanese characters.
- Workstations and printers are installed with fonts that include those characters.

For example, the Japanese Shift-JIS character set contains Japanese and many US-ASCII characters; it is sufficient for encoding both English data and the primary characters that are

Copyright © 1988, 2026, Oracle and/or its affiliates.

required in Japanese. However, it does not include the accented Latin characters that are needed for French, German, and other languages, so it is not a suitable character set for implementations that encompass Western European countries.

Given the sample list of common character sets in the previous table and the number of languages that are required by a typical global PeopleSoft implementation, selecting a character set can be daunting, especially when you are planning to support a large list of languages.

To simplify this situation, an industry consortium of vendors devised a universal character standard: the Unicode standard. This internationally recognized character standard represents every character that is required to write virtually every written language. The Unicode standard was developed and is maintained by the Unicode Consortium in conjunction with ISO. This standard shares the character repertoire with ISO/IEC standard 10646: the Universal Multiple-Octet Coded Character Set (UCS), also known as the Universal Character Set for short.

The PeopleSoft system uses Unicode throughout PeopleTools to simplify character handling. The PeopleSoft system allows the use of Unicode within PeopleSoft databases to enable you to maintain a single database with characters from virtually any language.

The Unicode standard and the ISO 10646 standard are available from their respective organizations.

See The Unicode Consortium.

See International Organization for Standardization (ISO).

Unicode Encodings

Unicode defines a code space of more than one million code points (or characters). Unicode code points are referred to by writing “U+” followed by the hexadecimal number—for example, U+0000, U+0061, U+FFFF, U+27741, and so on. To manage such a large repertoire of characters, Unicode defines multiple planes each comprising 65,533 code points, or character positions. Plane 0, covering the range of U+0000 to U+FFFF, is known as the Basic Multilingual Plane (BMP) and is generally sufficient for almost all modern languages. The other planes are intended to encode extended ideographic characters, archaic scripts, and other rarely used characters (such as advanced mathematical symbols). All characters from planes outside the BMP are known as *supplementary characters.*

PeopleTools fully supports the use of characters from the BMP only; supplementary character support is limited to display in the browser, storage in the database, and reporting output in BI Publisher. A tool that can be used to view Unicode character properties is http://www.unicode.org/charts/unihan.html. If a character is at codepoint U+20000 or higher, it is a supplementary character.

Several different Unicode encoding forms have been standardized based on two encoding methodologies. Unicode defines these two encoding methods: the Unicode Transformation Format (UTF), and UCS. An encoding maps the Unicode code points (or perhaps a subset of code points) to sequences of code values of some fixed size. In UTF encodings, the number in the encoding name indicates the number of bits per code value; in UCS encodings, the number indicates the number of bytes per code value.

Four common encodings of Unicode are widely used:

- UTF-32 — a 32-bit, fixed-width encoding; equivalent to UCS-4.
- UTF-16 — a 16-bit, variable-width encoding.
- UTF-8 — an 8-bit, variable-width encoding, which maximizes compatibility with ASCII.

Copyright © 1988, 2026, Oracle and/or its affiliates.

- UCS-2 — a 2-byte, fixed-width encoding; a subset of UTF-16 supporting characters in the BMP only.

Other Unicode encodings—such as, CESU-8, Java's Modified UTF-8, UTF-1, and others—have specific, and sometimes internal, applications and are not widely used for the interchange of information.

While the PeopleSoft system currently supports only the UTF-8 and UCS-2 encodings, the following table presents a brief comparison of all four common encodings:

Encoding	Description	Min. Bytes per Char.	Max. Bytes per Char.	PeopleSoft System Usage
UTF-32	The full, 32-bit (four-byte) encoding of Unicode. Each Unicode character is represented by a four-byte number. For example, the Latin small letter a character is represented in UTF-32 hexadecimal as 0x00000061. UTF-32 was formerly called UCS-4.	4	4	None

Copyright © 1988, 2026, Oracle and/or its affiliates.

Encoding	Description	Min. Bytes per Char.	Max. Bytes per Char.	PeopleSoft System Usage
UTF-16	An extension of UCS-2 in which the application references characters on planes other than the BMP by combining two UCS-2 code units to designate a single, non-BMP 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, known as supplementary characters, require two UTF-16 code units: a low surrogate and a high surrogate, together called a surrogate pair. When no supplementary characters are present, UTF-16 is identical to UCS-2.	2	4	None

Copyright © 1988, 2026, Oracle and/or its affiliates.

Encoding	Description	Min. Bytes per Char.	Max. Bytes per Char.	PeopleSoft System Usage
UTF-8	A transformation of Unicode that encodes each character as one to four bytes, depending on which character is being encoded. All US-ASCII characters are encoded in UTF-8 as one byte, and this byte is identical to the encoding in US-ASCII. UTF-8 data is therefore backward compatible with US-ASCII data. All characters in the BMP are encoded as one, two, or three bytes in UTF-8. Characters in other planes are encoded as four bytes in UTF-8. UTF-8 has three main advantages: it is fully US-ASCII compatible, US-ASCII data can be considered as UTF-8 data, and it does not require that all characters use two or more bytes of storage.	1	4	The PeopleSoft system uses UTF-8 for serving HTML pages in the PeopleSoft Pure Internet Architecture and for inbound and outbound XML.
UCS-2	A 2-byte (16-bit) representation of each Unicode character. As such, it can reference only 65,535 code points and is limited to characters in the BMP.	2	2	The PeopleSoft system uses UCS-2 in memory for the Microsoft Windows development tools and for the application server.

Unicode Encoding Examples

This section includes Unicode encoding examples for the following characters:

Copyright © 1988, 2026, Oracle and/or its affiliates.

Character	Unicode Code Point	Description
a	U+0061	Latin small letter *a*.
ñ	U+00F1	Latin small letter *ñ*.
€	U+20AC	Euro symbol

The following table shows the hexadecimal representation of these characters in each of the four Unicode encodings:

Unicode Encoding	Latin Small Letter *a*(U+0061)	Latin Small Letter *ñ*(U+00F1)	*€* Symbol(U+20AC)
UTF-32	0x0000006	0x000000F1	0x000020AC
UTF-16	0x0061	0x00F1	0x20AC
UTF-8	0x61	0xC3B1	0xE282AC
UCS-2	0x0061	0x00F1	0x20AC

Related Topics

- Selecting Email Character Sets
  The PeopleSoft system supports UTF-8 for outgoing Simple Mail Transfer Protocol (SMTP) email messages from PeopleTools application servers. In addition, the PeopleSoft system supports several additional encodings for outgoing email.

Non-Unicode Character Sets

Although much of the PeopleSoft system runs by using Unicode, you can configure several components with a non-Unicode character set. When making these choices, you should understand the types of character sets other than Unicode that exist.

This section discusses:

- Single-byte character sets (SBCSs).
- Double-byte character sets (DBCSs).

> **Note**
>
> 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. Informally, ANSI character sets refer to non-Unicode character sets, which can be any international, national, or vendor standard character set, such as those that are discussed at the beginning of this topic.

Single-Byte Character Sets

Most character sets use one byte to represent each character and are therefore known as SBCSs. These character sets are relatively simple and can represent up to 255 unique characters. Examples of SBCSs are ISO 8859-1 (Latin1), ISO 8859-2 (Latin2), Microsoft CP1252 (similar to Latin1, but vendor specific), and IBM CCSID 37.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Double-Byte Character Sets

DBCSs use one or two bytes to represent each character and are typically used for writing ideographic scripts, such as Japanese, Chinese, and Korean. Most DBCSs allow a mix of one-byte and two-byte characters, so you cannot assume an even-string byte length. Encoding with a mix of one- and two-byte characters is also known as variable-width encoding, and such a character set is sometimes referred to as a multi-byte character set (MBCS).

The PeopleSoft system supports two types of DBCSs:

- Nonshifting
- Shifting

The difference between these types of DBCSs is in the way in which the system determines whether a particular byte represents one character or is part of a two-byte character.

Term	Definition
Nonshifting DBCSs	Nonshifting DBCSs use ranges of codepoints, specified by the character set definition, to determine whether a particular byte represents one character or is part of a two-byte character. In nonshifting DBCSs, the two bytes that are used to form a character are called *lead bytes* and *trail bytes*. The lead byte is the first in a two-byte character, and the trail byte is the last. Nonshifting DBCSs 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 code page, Shift-JIS, uses this lead byte/trail byte encoding scheme, as do most Microsoft Windows and Unix/Linux ASCII-based double-byte character sets that represent Chinese, Japanese, and Korean characters. Contrary to its name, Shift-JIS is a nonshifting double-byte character set.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Term	Definition
Shifting DBCSs	A shifting DBCS is another double-byte encoding scheme in use that doesn't use the lead byte and trail byte concept. The IBM DB2 UDB for OS/390 and z/OS EBCDIC-based Japanese, Chinese, and Korean character sets use this shifting encoding scheme. Instead of reserving a range of bytes as lead bytes, shifting DBCSs always keep track of whether they are in double-byte or single-byte mode. In double-byte mode, every two 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 expected as 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 two bytes. This continues until a *shift-out* byte is detected, which indicates that the character set should go back to single-byte per characters. This scheme, while more complex than the lead byte and trail byte scheme, provides greater performance, because 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 comprises a mixture of single-byte and double-byte characters could 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. **Important:** In the PeopleSoft system, shifting DBCSs have limited usage, such as for file I/O, and are not supported for use as a database character set.

Character Sets Across the Tiers of the PeopleSoft Architecture

PeopleSoft installations include multiple components, each of which must be able to handle differing character sets.

PeopleSoft application servers and clients (for example, the PeopleTools development environment and PeopleSoft Pure Internet Architecture pages) use Unicode exclusively and do not rely on other character sets to represent and process data. However, depending on your environment, not all system components support Unicode-encoded data. Therefore, you might not be able to run all parts of your system in Unicode. For example, some database platforms and third-party products do not support Unicode. The following table illustrates support for Unicode in the PeopleSoft system.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Tier	Component	Unicode Support
Client	• PeopleTools development environment • PeopleSoft Pure Internet Architecture pages	• Yes • Yes
Web server	Web server	Yes
Application server	Application server	Yes
Database	• Non-Unicode DB (Western European or Japanese) • Unicode DB	• No • Yes
File attachment storage	• FTP server • HTTP repository	• Yes • Yes

Examples of how to configure these tiers are provided in this topic.

See Understanding Character Set Selection.

In addition to the tiers listed in the previous table, PeopleTools enables you to configure these system components to use other character sets:

- COBOL.

  The character set that is used for PeopleSoft COBOL processing must match the character set of the database. If you created a Unicode database for the PeopleSoft implementation, you must also run COBOL in Unicode.

- Direct file I/O operations.

  All direct file I/O operations in PeopleTools, including file layout objects, trace and log files, and file operations from Structured Query Report (SQR) programs can be performed in Unicode or any supported non-Unicode character set. This is useful in situations in which you must interface with an external system that does not support Unicode.

  Some applications refer to the ANSI encoding that inherits the encoding of the OS. For example, applications running on Windows OS inherits the active codepage. Application running on Unix OS inherits the encoding portion of the shell locale.

- Third-party products that are non-Unicode compliant.

  Some third-party products that are supported by PeopleTools do not yet support Unicode. In this case, PeopleTools converts application data to a specific, non-Unicode character set before communicating with these tools. Check the product documentation for your third-party application regarding Unicode compliance before determining how the application and the PeopleSoft system will interoperate.

When Unicode is not used for any of these types of operations or data storage, the PeopleSoft system transparently handles the conversion from Unicode to a non-Unicode character set. The non-Unicode character set that is used depends on several settings, which are discussed in detail later in this topic.

PSCHARSETS Table

The character sets that the PeopleSoft system supports are defined in the PSCHARSETS table. The following table lists these character sets and the names by which they may be referred to in PeopleSoft applications. You may need to know the correct character set name to use in several situations including:

- In PeopleCode programs for manipulating file layout objects.

Copyright © 1988, 2026, Oracle and/or its affiliates.

- In the Unix/Linux application server configuration to determine the default, non-Unicode character set for log files, trace files, and operating system interfaces.
- When creating your database.

  Refer to your hardware and software requirements guide for details about the character sets that are supported for your database platform.

PSCHARSETS Character Set Name	Description and Comments	Character Set Type
ANSI	Current ANSI-based 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 or DBCS, depending on the host operating system.
ASCII	7–bit US-ASCII	SBCS
Big5	Big5 (Traditional Chinese)	Nonshifting DBCS
CCSID1027	IBM EBCDIC 1027 (Japanese-Latin)	SBCS
CCSID1047	IBM EBCDIC 1047 (Latin1)	SBCS
CCSID290[1]	IBM EBCDIC 290 (Katakana)	SBCS
CCSID300[1]	IBM EBCDIC 300 (Kanji)	Nonshifting DBCS
CCSID930[2]	IBM EBCDIC 930 (Kana-Kanji)	Shifting DBCS
CCSID935[2]	IBM EBCDIC 935 (Simplified Chinese)	Shifting DBCS
CCSID937[2]	IBM EBCDIC 937 (Traditional Chinese)	Shifting DBCS
CCSID939[2]	IBM EBCDIC 939 (Latin-Kanji)	Shifting DBCS
CCSID942	IBM EBCDIC 942 (Japanese PC)	Nonshifting DBCS
CP1026	Windows 1026 (EBCDIC)	SBCS
CP1250	Windows 1250 (Eastern Europe)	SBCS
CP1251	Windows 1251 (Cyrillic)	SBCS
CP1252	Windows 1252 (Western Europe)	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 (U.S.)	SBCS
CP500	Windows 500 (EBCDIC 500V1)	SBCS
CP708	Windows 708 (Arabic - ASMO708)	SBCS
CP720	Windows 720 (Arabic - ASMO)	SBCS

Copyright © 1988, 2026, Oracle and/or its affiliates.

PSCHARSETS Character Set Name	Description and Comments	Character Set Type
CP737	Windows 737 (Greek - 437G)	SBCS
CP775	Windows 775 (Baltic)	SBCS
CP850	MS-DOS 850 (Western Europe)	SBCS
CP852	MS-DOS 852 (Eastern Europe)	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 865 (Nordic)	SBCS
CP866	MS-DOS 866 (Russian)	SBCS
CP869	MS-DOS 869 (Modern Greek)	SBCS
CP870	Windows 870	SBCS
CP874	Windows 874 (Thai)	SBCS
CP875	Windows 875 (EBCDIC)	SBCS
CP932	Windows 932 (Japanese)	Nonshifting DBCS
CP936	Windows 936 (Simplified Chinese)	Nonshifting DBCS
CP949	Windows 949 (Korean)	Nonshifting DBCS
CP950	Windows 950 (Traditional Chinese)	Nonshifting DBCS
EBCDIC	IBM EBCDIC CCSID37 (USA)	SBCS
EUC-JP	Extended UNIX code (Japanese)	Nonshifting DBCS
EUC-KR	Extended UNIX code (Korean)	Nonshifting DBCS
EUC-TW	Extended UNIX code (Taiwan)	Nonshifting DBCS
EUC-TW-1986	Extended UNIX code (TW-1986)	Nonshifting DBCS
GB12345	GB 2312 (Simplified Chinese)	Nonshifting DBCS
GB18030	GB18030 (Simplified Chinese)	Nonshifting DBCS
GB2312	GB 2312 (Simplified Chinese)	Nonshifting DBCS
HKSCS	Hong Kong Supplementary Character Set	Nonshifting DBCS
ISO-2022-JP[2, 3]	ISO-2022-JP Japanese	Shifting DBCS
ISO-2022-KR[2]	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

Copyright © 1988, 2026, Oracle and/or its affiliates.

PSCHARSETS Character Set Name	Description and Comments	Character Set Type
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[1]	Japanese Half-width Katakana	Nonshifting DBCS
JIS_X_0208	Japanese Kanji	Nonshifting DBCS
Java	Java (Unicode encoding)	Unicode
Johab	Johab (Korean)	Nonshifting DBCS
Shift_JIS	Shift-JIS (Japanese)	Nonshifting DBCS
UCS2	Unicode UCS-2	Unicode
UTF-8	Unicode UTF-8	Unicode
UTF7[1]	Unicode UTF-7. (An outdated Unicode 7-bit clean transformation that is sometimes used for email that must pass through gateways that do not support 8-bit characters.)	Unicode
UTF8	Unicode UTF-8	Unicode
UTF8BOM	Unicode UTF-8 with BOM (byte-order mark)	Unicode

[1] Not commonly used.

[2] In the PeopleSoft system, shifting DBCSs have limited usage, such as for file I/O, and are not supported for use as a database character set.

[3] To use certain Windows-31J (also known as Microsoft CP932) characters in incoming or outgoing email messages, you must complete additional configuration of your web server (incoming email) and application server or PeopleSoft Process Scheduler (outgoing email).

See Selecting Email Character Sets.

This PeopleBook also contains information about supported character set encodings for globalization when using SQR for PeopleSoft.

See SQR for PeopleSoft-Supported Character Set Encodings.

See, Certifications tab on My Oracle Support.

Related Topics

- Microsoft Code pages
- Code Chart for Unicode

Copyright © 1988, 2026, Oracle and/or its affiliates.

Selecting Character Sets

This section discusses selecting character sets.

Related Topics

- Understanding Character Sets
  This section discusses character sets.

Understanding Character Set Selection

When configuring your PeopleSoft system, you need to consider the character set (or sets) that will be in use on the following tiers:

- Client.
- Web server.
- Application server.
- Database server.
- File attachment storage location (FTP site, HTTP repository, or database table).
- Email.

Some operations of your PeopleSoft system require the interaction of multiple tiers. For example, the uploading of a file attachment involves the browser on the client, the web server, the application server, the database server, and ultimately the file attachment storage location. To ensure the correct transfer of data and files between these tiers, Oracle recommends configuring each server tier (web server, application server, database server, and file storage location) to use the same character set as follows:

- If your PeopleSoft system operates in a multi-language environment, use a UTF-8 character set on each server tier.
- If your PeopleSoft system operates in a single language environment, use the native language character set for that language on each server tier. Alternatively, you could use a UTF-8 character set on each server tier, which would provide more flexibility than using the native language character set.

Clients can always be configured to use the native language of the user of that workstation or browser.

The following table depicts example character set settings across all tiers for three typical configurations—a multi-language environment, a single language environment (Western), and a single language environment (non-Western).

> **Note**
>
> This table shows examples for a particular combination of languages and platforms; your specific configuration could differ.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Tier (Platform)	Multi-Language	Single Language(Western: French)	Single Language(Non-Western: Japanese)	Where to Check
Client (Windows)	Any (for example, English uses CP1252).	French (uses CP1252).	Japanese (uses CP932).	Start, Settings, Control Panel, Regional Options
Web server (Linux) – Shell processes	en_US.utf8	fr_FR.iso88915	ja_JP.sjis	locale command
Application server (Linux) – PSAPPSRV processes	utf-8	latin15	sjis	psappsrv.cfg [PSTOOLS] Character Set
Application server (Linux) – Email processes	utf-8	utf-8	utf-8	psappsrv.cfg [SMTP Settings] SMTP Character Set
Application server (Linux) – Shell processes	en_US.utf8	fr_FR.iso88915	ja_JP.sjis	locale command
Database server (Oracle)	AL32UTF8	WE8ISO885915	JS16SJISTILDE	NLS_DATABASE_PARAMETERS
File attachments: FTP site[4] (Linux) – Shell processes	en_US.utf8	fr_FR.iso88915	ja_JP.sjis	locale command

[4] For file attachments, if the storage location is a database table or an HTTP repository, then the configuration of one of the other server tiers will also configure the character set in use for a file attachment storage location on that tier. Specifically, a database table as a storage location depends on the settings for the database server; an HTTP file repository as a storage location depends on the web server settings if the HTTP repository is deployed on the web server. In the preceding table, information is provided for an FTP site as a storage location only because an FTP site can be deployed independently from the other server tiers.

Failure to configure character sets correctly across server tiers can result in garbled file names. To minimize character corruption issues, try to use the multi-language settings where possible.

Related Topics

- PeopleCode Developer's Guide: Attachments with non-ASCII File Names

Selecting Database Character Sets

The primary character set decision that you must make when installing a PeopleSoft implementation is which character set to use for the database system. Ideally, all databases are encoded in Unicode; however, in some cases Unicode requires several bytes to represent each character when only one byte may be required in a non-Unicode character set. Therefore, the PeopleSoft system enables you to use certain non-Unicode character sets for the database.

By using a Unicode encoded database, you can maintain a single database with data in any combination of languages. 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

Copyright © 1988, 2026, Oracle and/or its affiliates.

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 a few languages. If you create a non-Unicode database, you must ensure that all of the characters for all of the languages that you plan on using can be represented in the character set that you choose.

The following table lists whether a PeopleSoft language is supported in a Unicode or non-Unicode database character set:

Language Code	Language	Database Character Set
ARA	Arabic	Unicode
BUL	Bulgarian	Unicode
CFR	Canadian French	Unicode or non-Unicode
CRO	Croatian	Unicode
CZE	Czech	Unicode
DAN	Danish	Unicode or non-Unicode
DUT	Dutch	Unicode or non-Unicode
ENG	US English	Unicode or non-Unicode
FIN	Finnish	Unicode or non-Unicode
ESP	Spanish	Unicode or non-Unicode
FRA	French	Unicode or non-Unicode
GER	German	Unicode or non-Unicode
HUN	Hungarian	Unicode
ITA	Italian	Unicode or non-Unicode
JPN	Japanese	Unicode or non-Unicode
KOR	Korean	Unicode
NOR	Norwegian	Unicode or non-Unicode
POL	Polish	Unicode
POR	Portuguese	Unicode or non-Unicode
ROM	Romanian	Unicode
RUS	Russian	Unicode
SER	Serbian	Unicode
SLK	Slovak	Unicode
SLV	Slovenian	Unicode
SVE	Swedish	Unicode or non-Unicode
THA	Thai	Unicode
UKE	English	Unicode or non-Unicode
ZHS	Simplified Chinese	Unicode
ZHT	Traditional Chinese	Unicode

Depending on the data that you store and how the database stores Unicode characters, a Unicode database can be significantly larger than a non-Unicode database. However, only the storage of character data is affected; the space that is required for non-character data, such as numbers and dates (which are stored by the database system as numbers), is not affected.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Depending on the database platform, you can use one of the four character set types (SBCS, nonshifting DBCS, shifting DBCS, or Unicode) when creating the database. However, the number of characters that you can store in each column is affected greatly by the type of character set that you choose for the database encoding.

Related Topics

- The Unicode Standard
  The most important consideration when dealing with character sets across a system is ensuring that all characters that you plan to represent within the PeopleSoft system exist in the character set that is used by each component of the system.
- Non-Unicode Character Sets
  Although much of the PeopleSoft system runs by using Unicode, you can configure several components with a non-Unicode character set. When making these choices, you should understand the types of character sets other than Unicode that exist.

Selecting Application Server Character Sets

All data that is stored in memory and processed by the PeopleTools application server is held in Unicode. However, the application server allows files on the server (created through PeopleCode file layout objects) and log and trace files to be Unicode or non-Unicode. 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 character set, UTF-8. If a file operation must create a non-Unicode file, this character set is 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 in which character set the file should be created, the default non-Unicode character set of the application server is used.

Microsoft Windows enables you to change the default character set of the system, although as installed, the default character set matches the default locale of the Microsoft Windows installation. To change the system default locale (and therefore the character set), on Microsoft Windows servers, use the Control Panel's Regional Options menu. In the Language settings for the system section, click the **Set Default** button.

When running on Unix/Linux, the PeopleSoft application server enables you to specify the default non-Unicode character set in the application server's configuration file, which you select by using the PSADMIN tool. Any valid PeopleSoft character set with a character set type of SBCS or nonshifting DBCS is a valid default non-Unicode character set for PeopleSoft application servers that run on Unix/Linux.

Related Topics

- Character Sets Across the Tiers of the PeopleSoft Architecture
  PeopleSoft installations include multiple components, each of which must be able to handle differing character sets.

Selecting and Managing Client Workstation Character Sets

You must consider the client components of PeopleTools when you are planning your language strategy. The requirements for language support on client workstations are different, depending on whether you are using the PeopleSoft Pure Internet Architecture or the PeopleTools development tools for Microsoft Windows.

This section discusses:

- Character sets and fonts in the PeopleSoft Pure Internet Architecture.

Copyright © 1988, 2026, Oracle and/or its affiliates.

- Fonts and the PeopleTools development environment.
- Input methods.

Character Sets and Fonts in the PeopleSoft Pure Internet Architecture

The PeopleSoft Pure Internet Architecture serves all HTML pages in 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. All browsers supported by PeopleTools can support UTF-8 encoded HTML pages.

However, the browser needs other components 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. In addition, you might need alternate keyboard layouts or, in the case of ideographic scripts such as Chinese, Japanese, and Korean, you need input method editors (IMEs) to convert sequences of keystrokes into ideographs. The requirement for alternate keyboard and IMEs is the same for both the PeopleSoft Pure Internet Architecture and the PeopleTools development environment.

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 displayed language, you will most likely see square boxes in place of the appropriate characters. The 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.

The PeopleSoft Pure Internet Architecture includes a set of style sheets, defined with Application Designer, that determine the font that is used to display HTML pages. In some cases, the application data may contain characters that are not present in this font and that require a different font.

The Go Noto TrueType fonts shipped in the *PS_HOME*\fonts\truetype directory support all of the languages supported by the PeopleSoft system. Alternatively, 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. Obtain fonts from the following sources:

- Many Microsoft Windows and other operating system applications are packaged with Unicode fonts containing glyphs covering a large range of languages.

  Microsoft Office is packaged with several fonts containing a large portion of the characters in Unicode, including the Microsoft Sans Serif font. Use these fonts in the PeopleSoft Pure Internet Architecture by specifying them in the Application Designer style sheet definitions or by following the browser-specific instructions in this section.
- Many public domain fonts exist that contain a large character repertoire for use in web browsers. The unifont.org web site is one location to get additional information on public domain fonts.

  See Unicode Font Guide For Free/Libre Open Source Operating Systems.
- Several font foundries license fonts for individual or corporate use.

  Some of these foundries include Monotype, Bitstream, and Tiro Typeworks.

Depending on your browser, you can also download fonts from your browser's manufacturer.

To enable the display of GB18030 characters, you can use either the SimSun-18030 font from Microsoft or the Go Noto fonts shipped in the *PS_HOME*\fonts directory. Both of these fonts have glyphs for the supported ranges of the GB18030 character set.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Fonts and the PeopleTools Development Environment

PeopleTools enables you to specify the font that is used for all graphical components for all PeopleTools modules that run on Windows, such as Application Designer. Use these methods to specify fonts:

- Configuration Manager font setting (Display tab)

  This setting affects the font that is used by all of the designer components of PeopleTools, including all of the text that is contained in the Microsoft Windows resource files

  See Understanding PeopleTools Translation.

  Changing this font setting may be necessary if your workstation's default locale does not contain the characters that are used for the language that you are attempting to display or maintain. For example, if you are attempting to view Japanese characters on an English Microsoft Windows workstation, you can change the PeopleSoft Configuration Manager font setting to select a font that contains the characters for the language that you are trying to display.

  The Go Noto TrueType fonts shipped in *PS_HOME*\fonts directory support all of the languages supported by the PeopleSoft system.

  In addition, several fonts that are shipped with Microsoft Windows and Microsoft Office, including Arial Unicode MS and Microsoft Sans Serif, contain a large number of glyphs covering most of the languages that are supported by the Unicode character set. Microsoft Windows can also be configured with fonts for most worldwide languages by selecting the required languages under the Regional Settings Control Panel menu.

- PeopleCode font

  The PeopleCode editor in 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 that you are working on contain Unicode characters. To set the font in Application Designer, open the PeopleCode program, select Edit, Display Fonts and Color.

Fonts in PeopleSoft Charts and PDF Documents

The operating system on the client workstation provides the fonts for the browser to render text in Peoplesoft Pure Internet Architecture pages. In certain circumstances, the application server provides the fonts when text is rendered on the server. For example, for PeopleSoft charts sent as rendered images and for PDF documents produced by reporting tools such as SQR, or BI Publisher, the application server might need to be configured with fonts that contain the needed glyphs. If the font is not configured correctly, squares or blanks might appear in PeopleSoft charts or PDF documents when the same characters are rendering correctly in the browser.

> **Note**
>
> The most common case in which fonts are rendered on the application server is when Java is used to draw charts or reports. The fonts shipped by PeopleSoft can be found in PS_HOME\fonts\ttf, as well as in PS_HOME\jre\lib\fonts.

See BI Publisher for PeopleSoft: Understanding Report Template Types.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Input Methods

If users will enter translated data by using PeopleSoft Pure Internet Architecture or the PeopleTools development environment, you must ensure that an appropriate keyboard layout or input method editor is installed on the workstation.

Most alphabetic languages can be typed by using a relatively simple keyboard layout. Several specialized keyboard layouts exist for most languages; configure these keyboard layouts through your operating system. For example, a Spanish keyboard layout contains keys for the n-tilde character (ñ) and several other accented characters.

However, certain PeopleSoft hot keys do not work as expected on alternate, non-U.S. keyboard layouts. For example, **Alt+'**, **Alt+**, and **Alt+/** do not produce the expected results on the AZERTY keyboard. This occurs because some keys on non-U.S. keyboards produce different key codes than the same key on a U.S. keyboard (also known as a QWERTY keyboard).

A solution to this problem can be found in the appendix.

See PeopleSoft Hot Keys Do Not Function As Expected on a non-U.S. Keyboard.

There are several ways of entering these characters by using a nonlocalized keyboard. Your operating system manual can help you use specialized keyboard layouts, such as the English international layout, which enables you to enter accented characters by using two keystrokes. The Microsoft web site contains information about keyboards that are supported by Microsoft Windows and instructions for installing and configuring Windows keyboard layouts.

Ideographic 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 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 that is 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. The PeopleSoft Pure Internet Architecture supports any IME that is supported by your browser. The designer tools in Microsoft Windows support all standard Microsoft IMEs.

Selecting Email Character Sets

The PeopleSoft system supports UTF-8 for outgoing Simple Mail Transfer Protocol (SMTP) email messages from PeopleTools application servers. In addition, the PeopleSoft system supports several additional encodings for outgoing email.

PeopleSoft application servers support the following for outgoing email:

- UTF-8 (default).
- ISO-2022-JP, Shift_JIS, EUC-JP (for Japanese).
- ISO-2022-KR, EUC-KR (for Korean).
- GBK, Big5, GB18030 (for Chinese).

Specifying Email Character Sets

You specify an email character set in the SMTPCharacterSet parameter in the application server configuration file, psappsrv.cfg. By default, the SMTPCharacterSet parameter is set to UTF-8.

Copyright © 1988, 2026, Oracle and/or its affiliates.

> **Note**
>
> You should specify a value for the SMTPCharacterSet. If you do not specify a value for the parameter, email is sent as-is, with no encoding. Leave the parameter set to the default value of UTF-8 if you are not certain about which value to use.

For example, to use ISO-2022-JP encoding for outgoing SMTP mail, in the psappsrv.cfg file, set the SMTPCharacterSet parameter to ISO-2022-JP, as shown in the following example:

```
[SMTP Setting]
...
SMTPCharacterSet=ISO-2022-JP
SMTPEncodingDLL=blank
```

You can also write your own SMTPEncodingDLL modules, if necessary.

Converting Between Character Sets

You can use PeopleCode file functions to convert files or text strings from one supported PeopleSoft character set to another supported PeopleSoft character set.

PeopleCode operations such as GetFile, GetTempFile, Open, ReadLine, and WriteLine automatically account for the file encoding. Therefore, you can:

- Convert from a Unicode character set to a non-Unicode character set.
- Convert from a non-Unicode character set to a Unicode character set.
- Automatically handle the Unicode BOM if it is present or needs to be written.

When using a character set such as UCS2 or UTF8BOM, the BOM is added at the beginning of the file contents when using WriteLine. The BOM is skipped when read by the ReadLine PeopleCode function, and not interpreted as a text character. Since the BOM is recognised as meta data and not part of the file's text the BOM is not added to file contents when writing in a non-Unicode character set with the WriteLine PeopleCode function.

In the following example PeopleCode program, the FileEncodingConversion function handles converting files from one supported character set to another. In the body of the program, the function is called to convert from the UTF8BOM character set to the UCS2 character set.:

```
REM this function, FileEncodingConversion() converts character encoding of
input file to another of output file.
REM an example of how to call the function is at the end of this file.

Local File &InputFile, &OutputFile;
Local string &InputDirFile, &OutputDirFile, &InputFilename, &OutputFilename;
Local string &sDirSep, &LogLine;
Local array of string &FIleEncoding;
Local boolean &ret;

Function FileEncodingConversion(&InputEncoding, &InputDirectoryFile,
&OutputEncoding, &OutputDirectoryFile) Returns boolean

   &InputFile = GetFile(&InputDirectoryFile, "R", &InputEncoding,
%FilePath_Absolute);
```


Copyright © 1988, 2026, Oracle and/or its affiliates.

```
   &OutputFile = GetFile(&OutputDirectoryFile, "W", &OutputEncoding,
%FilePath_Absolute);

   If &InputFile.IsOpen And
&OutputFile.IsOpen Then

      While (&InputFile.readline(&LogLine))
&OutputFile.Writeline(&LogLine);
      End-While;

      &InputFile.Close();
      &OutputFile.Close();
      Return True;
   Else
      If &InputFile = Null Then
         WinMessage("Error: PeopleCode: File Encoding I/O: " | "Failed to
open: " | &InputFile.Name);
      Else
         If &OutputFile = Null Then
            WinMessage("Error: PeopleCode: File Encoding I/O: " | "Failed
to open:" | &OutputFile.Name);
         End-If;
      End-If;
      Return False;
   End-If;

End-Function;

/*---------------------------------------------------------------------------*/
/*  Function IsUnix                                                          */
/* check if OS = Unix                                                        */
/*---------------------------------------------------------------------------*/
Function IsUnix Returns boolean
   &DummyFile = GetFile("/bin/sh", "E", %FilePath_Absolute);
   If &DummyFile.IsOpen Then;
      &DummyFile.Close();
      Return True;
   Else;
      Return False;
   End-If;
End-Function;

REM test the function above;
&FIleEncoding = CreateArray("UTF8BOM", "UCS2", "SJIS", "GB18030", "UTF8", ⇒
"a", "u");

REM WinMessage("ret: " | &ret);

If IsUnix() Then
   /* for UNIX */
   &ret = FileEncodingConversion(&FIleEncoding [1], "/home/FS_" |
&FIleEncoding [1] | ".txt", &FIleEncoding [2], "/home/BEFORE/FS_PCode_" | ⇒
&FIleEncoding [1] | "_to_" | &FIleEncoding [2] | ".txt");
Else
   /* for WINDWOS */
   &ret = FileEncodingConversion(&FIleEncoding [1], "D:\TMP\FS_" |
```


Copyright © 1988, 2026, Oracle and/or its affiliates.

```
&FIleEncoding [1] | ".TXT", &FIleEncoding [2], "D:\TMP\AFTER\FS_PCode_" |
&FIleEncoding [1] | "_to_" | &FIleEncoding [2] | ".TXT");
End-If
```

Setting Data Field Length Checking

This section provides overviews of Application Designer field length semantics and field length checking for non-Unicode databases and discusses how to enable or disable data field length checking.

Understanding Application Designer Field Length Semantics

The database character set determines the way that PeopleTools interprets the column length that is defined in Application Designer.

If you create a Unicode database, the field length, as shown in Application Designer, indicates the maximum number of Unicode BMP characters that are permitted in the field, regardless of the Unicode encoding that is used by the database. Some database platforms, such as Oracle with byte semantics, use byte lengths to measure column sizes when operating in a Unicode database, while others use character lengths.

When the database uses byte-sized column lengths, the PeopleSoft system sizes the database columns based on the worst-case ratio between bytes and characters in the Unicode encoding that is used by your database. For example, if the AL32UTF8 character set is used by Oracle with byte semantics, the worst-case character-to-byte ratio when running against an Oracle Unicode database is 1:3. So, column size is tripled when creating a Unicode database on Oracle. A field that is defined in Application Designer as a CHAR(10) is created on an Oracle Unicode database with a type of VARCHAR2(30). This tripling of the maximum column size does not affect the actual size of the database, because variable length character fields do not reserve space in the database.

Other database platforms use character-based column lengths whose sizes represent the maximum number of Unicode characters instead of bytes that may be stored. Examples of this implementation are the NCHAR data type in Microsoft SQL Server and the GRAPHIC data type in DB2 UDB for Linux, Unix, and Microsoft Windows.

If you create a non-Unicode database, the field length in Application Designer represents the number of bytes that are permitted in the field, based on the character set that you used to create the database. Therefore, a PeopleSoft Unicode database enables you significantly more space for character data within the database when dealing with ideographic languages, such as Japanese, that require one or more byte to store per character.

The following tables show some of the possible database encodings for database platforms that the PeopleSoft system supports in Unicode and non-Unicode and their effects on database column sizes. Each table shows the database representation and the worst case number of characters allowed in the character field for a character field defined in Application Designer with a length of 10.

This table shows the information for an Oracle database with byte semantics (used by PeopleSoft 8.9 applications and earlier):

Database Character Set	Database Representation	Number of Characters
Unicode (AL32UTF8)	VARCHAR2(30)	10
Any SBCS	VARCHAR2(10)	10

Copyright © 1988, 2026, Oracle and/or its affiliates.

Database Character Set	Database Representation	Number of Characters
Shift-JIS (JA16SJIS or JA16SJISTILDE)	VARCHAR2(10)	5

This table shows the information for an Oracle database with character semantics (used by PeopleSoft 9.0 applications and later):

Database Character Set	Database Representation	Number of Characters
Unicode (AL32UTF8)	VARCHAR2(10)	10
*Any SBCS	VARCHAR2(10)	10
*Shift-JIS (JA16SJIS or JA16SJISTILDE)	VARCHAR2(10)	10

**: SBCS and Shift JIS are supported only when using BYTE length semantics, while Unicode is supported using CHAR length semantics. SBCS and Shift JIS are listed in the above table for comparison purposes only.*

This table shows the information for an Microsoft SQL Server database with VARCHAR semantics:

Database Character Set	Database Representation	Number of Characters
Unicode (UCS-2)	NVARCHAR(10)	10
Any SBCS	VARCHAR(10)	10
Shift-JIS (CP932)	VARCHAR(10)	5

This table shows the information for a Microsoft SQL Server database with CHAR semantics:

Database Character Set	Database Representation	Number of Characters
Unicode (UCS-2)	NCHAR(10)	10
Any SBCS	CHAR(10)	10
Shift-JIS (CP932)	CHAR(10)	5

This table shows the information for all other databases:

Database Character Set	Database Representation	Number of Characters
Any SBCS	CHAR(10)	10

Understanding Field Length Checking for Non-Unicode Databases

The maximum number of characters that are permitted in a PeopleSoft field varies, depending on the character set of the database. Because all components of PeopleTools use Unicode for internal storage, by default, field length checking occurs in terms of Unicode character counts. This calculation is appropriate for Unicode databases and for any SBCS databases.

However, if you are using a non-Unicode DBCS, special length checking must occur each time you move off a field to ensure that the string that you entered fits in the database column when the string is converted to the database's character set.

Copyright © 1988, 2026, Oracle and/or its affiliates.

For graphically sizing page fields, PeopleTools uses the Unicode length of the field as defined in Application Designer. For example, if a field is defined in Application Designer as a 10-character field, page fields in both the PeopleSoft Pure Internet Architecture and the PeopleTools clients for Microsoft Windows allow 10 characters to be displayed unless manually resized by the developer.

However, if the database is encoded in a non-Unicode DBCS character set, such as Japanese Shift-JIS, special length validation must occur because the database column size is created relative to a byte count, not to 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 nonshifting 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.

Enabling or Disabling Data Field Length Checking

To enable or disable data field length checking:

1. Select **PeopleTools**, and then **Utilities**, and then **Administration**, and then **PeopleTools Options.**

   The PeopleTools Options page appears.

2. From the Data Field Length Checking drop-down list box, select a value based on the character set that you are using for the database:

Field or Control	Description
Others	Select if you are using a Unicode encoded database or a non-Unicode SBCS database. This option prevents special field length checking, which is not required by these types of databases.
MBCS	Select if you are running a non-Unicode Japanese database on any other platform. This option enables field length checking based on a nonshifting DBCS.

Alternatively, you can set this option by altering the PSOPTIONS.DBLENGTHTYPE field in the PSOPTIONS table. In the PSOPTIONS.DBLENGTHTYPE field in the PSOPTIONS table, select a value from the following

Field or Control	Description
N	Select if you are using a Unicode encoded database or a non-Unicode SBCS database. This option prevents special field length checking, which is not required by these types of databases.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Field or Control	Description
M	Select if you are running a non-Unicode Japanese database on any other platform. This option enables field length checking based on a nonshifting DBCS.

> **Note**
>
> The non-Unicode DBCS settings are specifically oriented to Japanese language installations, because Japanese is the only language that the PeopleSoft system supports in a non-Unicode DBCS encoding. All languages other than Western European languages and Japanese are supported by the PeopleSoft system only when using Unicode encoded databases.

> **Note**
>
> The value to specify data field length checking must be set correctly in order for PeopleSoft applications to perform correctly in a browser.

3. Click the Save button.

Using CJK Ideographic Characters in Name Character Fields

This section discusses PeopleSoft standard name conventions, including name conventions for Chinese, Japanese, and Korean (CJK) ideographic characters. PeopleSoft standard name conventions apply when data is entered or displayed in character fields that use Name as the format type. These conventions should be used when a complete name is constructed from multiple name character fields or when all name data is entered into a single name character field.

The PeopleSoft standard name convention is:

```
[Lastname] [Suffix],[Prefix] [Firstname] [Middle name/Initial]
```

The entry can contain alphanumeric characters, spaces, periods, hyphens, and apostrophes. Uppercase and lowercase characters are preserved as entered, that is, mixed case formatting is used automatically.

> **Important**
>
> While a space is a valid character, it is not allowed immediately after the comma separating the last name from the first name.

Examples of typical suffixes include degrees, affiliations, and titles such as MD, PhD, Jr., and III. Examples of typical prefixes include titles and honorifics such as Ms., Mr., Dr., Rev., and Hon.

Valid examples of these conventions include:

Copyright © 1988, 2026, Oracle and/or its affiliates.

Name as Displayed by PeopleSoft Convention	Name Elements Used	Actual Name
O'Brien,Michael	`[lastname],[firstname]`	Michael O'Brien
Jones IV,James	`[lastname] [suffix], [firstname]`	James Jones IV
Phillips MD,Deanna Lynn	`[lastname] [suffix], [firstname] [middle name]`	Deanna Lynn Phillips, MD
Reynolds Jr.,Dr. John Q.	`[lastname] [suffix], [prefix] [firstname] [middle initial]`	Dr. John Q. Reynolds Jr.
Phipps-Scott,Ms. Adrienne	`[lastname],[prefix] [firstname]`	Ms. Adrienne Phipps-Scott
Knauft,Günter	`[lastname],[firstname]`	Günter Knauft

However, if the name contains any CJK ideographic characters, different standard name conventions apply.

If the name contains any Japanese or Korean ideographic characters, the first and last names are separated by a space instead of a comma. In Japanese, a prefix or suffix is optional; in Korean, only an optional prefix can be used. These modified PeopleSoft standard name conventions can be used when a name includes any of the following types of characters:

- Japanese or Korean unified ideographs (Japanese Kanji or Korean Hanja).
- Japanese half-width or full-width Katakana.
- Japanese Hiragana.
- Korean Hangul.

The PeopleSoft standard name convention for Japanese names including these ideographic characters is:

```
[lastname] [firstname][{suffix|prefix}]
```


Copyright © 1988, 2026, Oracle and/or its affiliates.

The PeopleSoft standard name convention for Korean names including these ideographic characters is:

```
[lastname] [firstname][prefix]
```

Valid examples of these conventions include:

Name as Displayed by PeopleSoft Convention	Name Elements Used	English Equivalent
塩次 伸二	`[lastname] [firstname]`	Shinji Shiotsugu
塩次 伸二様	`[lastname] [firstname] [prefix]`	Mr. Shinji Shiotsugu
□ □□	`[lastname] [firstname]`	Hong Gildong
□ □□□	`[lastname] [firstname] [prefix]`	Mr. (or Ms.) Hong Gildong

If the name contains Chinese Hanzi, there is no space or comma between the first name, last name, suffix, and prefix. The PeopleSoft standard name convention for names including Chinese Hanzi characters is:

```
[lastname][firstname][{suffix|prefix}]
```

Valid examples of this convention include:

Name as Displayed by PeopleSoft Convention	Name Elements Used	English Equivalent
陳嘉明	`[lastname][firstname]`	Chen Jiaming
陳嘉明先生	`[lastname][firstname] [prefix]`	Mr. Chen Jiaming

Related Topics

- Application Designer Developer’s Guide: Specifying Character Field Attributes
- PeopleCode Language Reference: Character Processing

Copyright © 1988, 2026, Oracle and/or its affiliates.

Detecting and Converting Between Character Types

PeopleTools also provides PeopleCode string functions that recognize and convert between different characters within the Japanese character set. This enables you to detect, convert, and enforce the types of characters that you can enter in any PeopleSoft field. For example, the PeopleSoft system uses these functions in the development of the Alternate Character Architecture in some PeopleSoft applications. The Alternate Character Architecture is used in several PeopleSoft applications to provide a feature that enables the entry of, and enforces the characters contained in, Japanese phonetic spellings (Furigana) by using the Hiragana or Katakana scripts.

The following PeopleCode string functions can be used to recognize and convert between different characters within the Japanese character set:

- CharType
- ContainsCharType
- ContainsOnlyCharType
- ConvertChar

Related Topics

- PeopleCode Language Reference: CharType function

Copyright © 1988, 2026, Oracle and/or its affiliates.

5

Controlling Currency Display Format

Understanding Currency-Specific Settings

PeopleTools can format a currency amount that appears on a page or a report based on both your numerical format preferences and 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 appears. Those properties include:

- A currency symbol, such as *$* for the Australian dollar.
- A decimal precision.

  For example, the Australian dollar ($5.00) has two decimal positions, but the Korean Won (500 KRW) has no decimal positions.

Currency information is stored in the Currency Code table (CURRENCY_CD_TBL) record definition along with other information about the currencies that is used in the implementation. The CUR_SYMBOL field controls the currency symbol. The DECIMAL_POSITIONS field controls the decimal precision for a currency.

> **Note**
>
> The PeopleSoft Currency Code table (CURRENCY_CD_TBL) contains sample currency and country code data. The Currency Code table is based on ISO Standard 4217, “Codes for the representation of currencies,” and also relies on ISO country codes in the Country table (COUNTRY_TBL). The navigation to the pages where you maintain currency code and country information depends on which PeopleSoft applications you are using. To access the pages for maintaining the Currency Code and Country tables, consult your application PeopleBooks for more information.

Setting Up Currency Amount Fields When Developing Applications

To set up currency amount fields:

1. Access Application Designer.
2. Ensure 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 and hold a valid Currency Code value. This means that it must be a character (Char) field with a length of 3, formatted in uppercase (Upper). 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

Copyright © 1988, 2026, Oracle and/or its affiliates.

PeopleSoft field name. However if the record has only one amount field, you should use the CURRENCY_CD field name for consistency with other PeopleSoft applications.

An example of a record with two currency-controlled amount fields is the currency exchange (CURRENCY_EXCHNG) record. The two amount fields on this record are from amount (FROM_AMT) and converted amount (CONVERTED_AMT). Each has a currency control field of from currency (FROM_CUR) and to currency (TO_CUR), respectively.

3. Ensure that the edit properties of the currency control field are set up correctly.
4. 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.

   In the **Currency Control Field** drop-down list box, select the appropriate currency control field for the currency amount field.
5. Click **OK** to accept the property settings.
6. Save the record definition.

Setting Currency Field Display Properties When Developing Applications

When you place a currency amount field on a page, you can choose whether to display the currency symbol in the amount field along with the numerical amount and whether to display a thousands separator. Regardless of whether you display the symbol in the field, it is not stored in the database. Currency field formatting is performed only if the **Multi-Currency** option is selected on the PeopleTools Option page.

To set currency field display properties:

1. In Application Designer, display the field's Page Field Properties dialog box.

   To open the Page Field Properties dialog box for the currency field, right-click the page field and select Page Field Properties.
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 appear with a thousands separator character. The specific character that is used as the thousands separator is determined by your international preference.
3. Click **OK** to accept the dialog box settings.
4. Save the page definition.

Related Topics

- Setting Up Locale-Based Formatting for the PeopleSoft Pure Internet Architecture
  This section discusses setting up locale-based formatting.
- Using System-Wide Multicurrency Settings
  This section provides an overview of multicurrency settings and discusses how to use the system-wide multicurrency settings.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Using System-Wide Multicurrency Settings

This section provides an overview of multicurrency settings and discusses how to use the system-wide multicurrency settings.

Understanding Multicurrency 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.
- Validation of user entries against the currency's defined decimal precision.

  This setting causes the system to issue an error if the user attempts to enter a decimal precision that is greater than the precision that is allowed by the currency code definition.
- The display of global currency control fields.

  You can design an application so that fields that are specifically related to multiple currencies can appear globally or be hidden, depending on the implementation. Although multicurrency fields are hidden from pages when the multicurrency option is disabled, the fields still exist in the page buffer; so if they are required fields, they must have default values.
- The hiding of all fields.

  If you implement the application in an enterprise that uses only a single currency, you can hide all of the fields by clearing the **Multi-Currency** check box. If the enterprise later begins to use multiple currencies, you can redisplay the multicurrency fields by selecting the check box. Apply this technique only to fields that can be globally hidden without affecting the functionality of the application when working in a single currency. Use this technique for user-operated currency control fields, but not for fields that actually display currency. You also need to set the defaults of the currency display fields to the single currency that is used by the enterprise.

  Unless you specifically want to maintain your entire system in a single currency, the **Multi-Currency** check box remains selected.

Connecting a Currency Control Field to the Multi-Currency Check Box

To connect a currency control field to the system-wide multicurrency check box when developing applications:

1. In Application Designer, display the field's Page Field Properties dialog box.

   To open the Page Field Properties dialog box for the currency field, right-click the page field, and select Page Field Properties. In the Field Properties dialog box, select the Use tab.
2. Select the **Multi-Currency Field** check box on the Use tab to specify that the field can appear or hide from the PeopleTools Options page.
3. Click **OK.**
4. Save the page definition.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Enabling or Disabling System-Wide Currency Settings

To enable or disable system-wide currency settings at installation:

1. Select **PeopleTools**, and then **Utilities**, and then **Administration**, and then **PeopleTools Options.**

   The PeopleTools Options page appears.

2. Select or clear the **Multi-Currency** check box.

   Selecting the check box activates automatic currency formatting; clearing the check box disables 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. Click **Save.**

Examples

The following examples demonstrate the different outcomes of setting the Multi-Currency Field option in a field's properties and on the PeopleTools Options page to control a field's visibility.

- In the field properties, the **Multi-Currency Field** check box is *not* selected. On the PeopleTools Options page, the **Multi-Currency** check box is *not* selected.

  In this case, the field's visibility is not controlled by the PeopleTools Options page, so it appears (or doesn't appear) based on the developer coding.

- In the field properties, the **Multi-Currency Field** check box is *not* selected. On the PeopleTools Options page, the **Multi-Currency** check box *is* selected.

  In this case, the field's visibility is not controlled by the PeopleTools Options page, so it appears (or doesn't appear) based on the developer coding.

- In the field properties, the **Multi-Currency Field** check box *is* selected. On the PeopleTools Options page, the **Multi-Currency** check box *is* selected.

  In this case, the field's visibility is controlled by the PeopleTools Options page, so it appears.

- In the field properties, the **Multi-Currency Field** check box *is* selected. On the PeopleTools Options page, the **Multi-Currency** check box is *not* selected.

  In this case, the field's visibility is controlled by the PeopleTools Options page, so it does *not* appear.

Resizing Currency Fields by Using the International Field Size Utility

PeopleSoft applications are typically shipped with amount fields that are sized at 15 integer positions and 3 decimal positions. However, some older applications may be shipped with smaller field sizes, and to use a greater precision, you may need to run the International Field Size utility when you install your PeopleSoft system. If the PeopleSoft installation guide for your product requires it, use the Application Engine program Set International Field Sizes (TLSINST1) to increase the length and number of decimal position settings of field definitions. For example, you may need to do this if you process low-value currencies that require numeric fields that are longer than those that are provided in the standard application.

Copyright © 1988, 2026, Oracle and/or its affiliates.

PeopleTools is delivered with the International Field Sizes table (PS_INTL_FLDSIZ_TBL) 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.

Keep in mind the limitations on numeric field size and precision imposed by your database platform, and the fact that some numeric fields are used in COBOL and may require COBOL working storage changes in addition to the changes performed by this utility.

To resize international currency fields:

1. Start the PeopleTools International Field Size utility.

   Select **PeopleTools**, and then **Utilities**, and then **International**, and then **Resize Currency Fields** to display the 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, and the proposed new size of the field (stored in the International Field Sizes table).

2. Edit the data in the International Field Sizes table.

   Adjust the new field sizes to meet your own requirements. You can insert or delete rows from the page.

3. Save your changes.

4. Run the Process Request Dialog.

   To do this: click **Run** to open the Process Scheduler Request page. Set the report options, and then click **OK** to run the Set International Field Sizes process.

   The utility updates the field definitions and creates a report showing all page fields that have been affected by the changes in field size.

5. Check all affected pages, and rearrange page fields as necessary to rectify overlaps.

   The process does not update page field information such as overlapping fields that may be caused by an increase in field length; however, the system automatically adjusts the sizes of any page fields with a size property of *Average* or *Maximum.* If the page field size is *Custom,* it is not 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.

6. Rebuild (SQL ALTER) any tables that have been affected by the changes in field sizes.

   Use the Application Designer Build feature, with the Alter Tables option, to build any tables (using SQL ALTER) that have been affected by the changes in field size. Use the Find Object References features to determine which tables have been affected, or create a query against the PSRECDEFN, PSRECFIELD, and International Field Sizes 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
```


Copyright © 1988, 2026, Oracle and/or its affiliates.

6

Running COBOL in a Unicode Environment

Understanding COBOL in a Unicode Environment

This section discusses COBOL in Unicode environment.

Unicode Encodings in PeopleSoft COBOL

The character set that is used for PeopleSoft COBOL processing must match the character set for your database. If you created a Unicode database for the PeopleSoft system, you must also run COBOL in Unicode.

> **Note**
>
> 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.

The Unicode standard provides several methods of encoding Unicode characters into a byte stream. Each encoding has specific properties that make it suitable for use in different environments. The two main encodings that are important to understanding how PeopleSoft COBOL operates when running in Unicode are:

- UCS-2 (2-byte Universal Character Set) — which is the Unicode encoding that PeopleTools uses internally for data that is held in memory on the application server.

  UCS-2 encodes all characters into a fixed storage space of two bytes.

- UTF-8 (8-bit Unicode Transformation Format) — which is the encoding that the PeopleSoft system uses in COBOL.

  UTF-8 uses a format that varies from one to four bytes per character. Currently, the PeopleSoft system supports Unicode's Basic Multilingual Plane (BMP), which requires one to three bytes per character. Four-byte UTF-8 characters are required to represent *supplementary characters* that are outside Unicode's BMP.

In UTF-8, the actual number of bytes to encode a character can be determined by the first three bits of the first, and sometimes only, byte of a character. The following table shows how the bit pattern of the first byte is related to the number of bytes needed to encode the UTF-8 character.

Unicode Code Point Range	UTF-8 Bit Pattern	UTF-8 Character Length
U+0000 – U+007F	0*xxxxxxx*	One byte
U+0080 – U+07FF	110*xxxxx* 10*xxxxxx*	Two bytes
U+0800 – U+FFFF	1110*xxxx* 10*xxxxxx* 10*xxxxxx*	Three bytes

The *x* bit positions are filled with the bits of the character code number in binary representation. The rightmost *x* is the least-significant bit (a big-endian representation). In multi-byte

Copyright © 1988, 2026, Oracle and/or its affiliates.

sequences (for Unicode code points greater than U+007F), the number of leading 1 bits in the first byte is identical to the number of bytes in the entire sequence. In addition, each byte in a multi-byte sequence has the most significant bit set.

This section includes Unicode encoding examples for the following characters:

Character	Unicode Code Point	Description
a	U+0061	Latin small letter *a.*
ñ	U+00F1	Latin small letter *ñ.*
€	U+20AC	Euro symbol

The following table shows the difference in how UCS-2 and UTF-8 encode several characters:

Character	Unicode Code Point	UCS-2 Byte Values	UTF-8 Byte Values
a	U+0061	0x00 0x61	0x61
ñ	U+00F1	0x00 0xF1	0xC3 0xB1
€	U+20AC	0x20 0xAC	0xE2 0x82 0xAC

The PeopleSoft system transparently handles the conversion between UCS-2 and UTF-8 when data is passed into the COBOL program from the database. If you are reading or writing files directly from a COBOL program, the input and output files are UTF-8 encoded when running PeopleSoft COBOL programs in Unicode.

Related Topics

- The Unicode Standard
  The most important consideration when dealing with character sets across a system is ensuring that all characters that you plan to represent within the PeopleSoft system exist in the character set that is used by each component of the system.

Expanded Storage Space Requirements

Moving to a COBOL Unicode environment means that character data can potentially require three times the storage space that is required in a single-character environment, given the variable length of a character that is encoded in UTF-8 from one to three bytes. 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. This expansion is critical because in a Unicode PeopleSoft database, column sizes are calculated based on a character length, not a byte length. So, a CHAR(10) column on a Unicode database allows the storage of 10 characters, regardless of how many bytes each character takes to store. Given the three-bytes-per-character maximum requirement of UTF-8 (four-byte UTF-8 characters are not yet supported by the PeopleSoft system), the maximum byte size of this CHAR(10) column is 30 bytes. Therefore, a COBOL type of PIC X(30) may be required to store the contents of a CHAR(10) field on a Unicode database.

The PeopleSoft system provides a COBOL conversion utility to automatically expand the character-data fields in the working storage to accommodate the number of bytes in the UTF-8 encoding scheme.

Related Topics

- Running the COBOL Conversion Utility
  This section discusses the COBOL conversion utility.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Special Logic for Parsing Unicode Strings

In a non-Unicode COBOL installation, parsing through a string is easy because you can assume that all characters coming in are one byte in length. But in UTF-8, a character can vary between one byte and three bytes in length. Therefore, you must incorporate special logic to handle string parsing when you are dealing with characters in UTF-8 format.

Related Topics

- Identifying Unicode and Non-Unicode Data
  A COBOL program may need to determine whether it's dealing with non-Unicode or Unicode data. For example, if the program parses a string character, it must apply different logic depending on whether the string is non-Unicode or Unicode. The program can get this information from the ENCODING-MODE-SW in the PTCSQLRT copy library (ANSI-Mode is the same as non-Unicode):
- Defining Single Character Arrays
  Some COBOL programs define single-character arrays to parse or examine a string of characters, one character at a time. In a Unicode environment, be sure that you're examining the string one character at a time, not one byte at a time.

COBOL Sorting

Any in-memory sorting that is performed by using COBOL functions is performed as a binary sort in the current character encoding that is used for COBOL processing and may not necessarily match the sort order that is returned by the database in response to an ORDER BY clause. If you require the database to return data that is sorted by using a binary sort of its encoding rather than the default linguistically correct sort, you must use the %BINARYSORT meta-Structured Query Language (meta-SQL) function around each column in the WHERE or ORDER BY clause where binary ordering is important.

However, for DB2 UDB for OS/390 and z/OS implementations, this binary sorting is equivalent only when the COBOL program is run on a DB2 UDB for OS/390 and z/OS server. For example, the binary sort that is produced in COBOL differs from the binary sort that is produced by the database, because the database is encoded in EBCDIC and the client is in an ASCII-based encoding. Therefore, use the %BINARYSORT meta-SQL function only in COBOL programs that are not run through RemoteCall (the DB2 UDB for OS/390 and z/OS platform is not supported as a RemoteCall server).

When running against non-z/OS systems, the %BINARYSORT function can be used in both RemoteCall and non-RemoteCall programs.

For example:

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

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

Copyright © 1988, 2026, Oracle and/or its affiliates.

Related Topics

- PeopleCode Language Reference: %BINARYSORT meta-SQL element

Unicode-Specific Error Messages

These 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 a non-Unicode API to access a Unicode database.
- Attempting to use a non-Unicode COBOL with a Unicode database.
- Attempting to use a Unicode API to access a non-Unicode database.
- Fetch failed: the converted Unicode length of *length* exceeds the allocated buffer length *length* for column *column_number.*

These messages appear in the COBOL output log file.

Running the COBOL Conversion Utility

This section discusses the COBOL conversion utility.

Understanding the COBOL Conversion Utility

As delivered, PeopleSoft COBOL programs are configured to run only on non-Unicode databases. To run the PeopleSoft-delivered COBOL on a Unicode database, it first must be converted by using the PeopleTools COBOL conversion utility. This utility is typically called automatically by the PeopleSoft installation process; however in certain circumstances, such as when you adapt COBOL code or apply a PeopleSoft-provided patch to a COBOL program, you may need to run the COBOL conversion utility manually.

Moving to a COBOL Unicode environment means that character data can potentially require three times the storage space that is required 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.

Adapt and apply patches to only one set of COBOL source code—non-Unicode source. It is much easier to write COBOL programs without having to remember to triple the size of your working storage as you go. Once your adaptation or patch is complete and you are ready to compile the program, first run it through the COBOL conversion utility, then compile it. This approach has several benefits over customizing the converted code:

- You maintain a single source tree for all of your COBOL—the non-Unicode source.

  This way you don’t run the risk of accidentally adapting both the non-Unicode COBOL programs *and* the Unicode-converted COBOL programs and potentially losing the modifications to the converted programs the next time you run the converter.
- Although PeopleSoft developers test all delivered COBOL programs and patches in both Unicode and non-Unicode environments, only non-Unicode versions of the source are delivered.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Therefore, any time you apply a PeopleSoft COBOL patch to a Unicode system, the patched source code must be run through the COBOL converter. If you had already modified the post-converted source, the reconversion would obliterate your modifications.

If the COBOL conversion utility makes modifications to your code that are undesirable, instead of modifying the postconverted code, the PeopleSoft system provides a series of directives to the utility that can tell it how specific lines of code should (or should not be) converted. This enables you to limit your changes only to the nonconverted code and to make the conversion completely automated.

In a non-Unicode (also known as ANSI) implementation, 1 character typically occupies one 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 are required for any character field. Therefore, in the Unicode environment, you must define this same 10-character field with a PICTURE clause of PIC X(30).

To accommodate the number of bytes in the UTF-8 encoding scheme, the PeopleSoft system provides a COBOL conversion utility to expand the character fields in the working storage.

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 non-Unicode 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 in which it was found. Example: -t:d:\PT8\SRC\CBLUNICODE
-r or -rd:*Temp_directory*	See Viewing Error Logs. -r generates only the summary log file; -rd generates all of 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, the PeopleSoft batch utilities that compile your COBOL programs include logic to convert all programs and copybooks before compiling. This logic is triggered only when the Unicode version of PeopleTools is installed.

Compiling COBOL in Microsoft Windows

Use the *PS_HOME*\setup\cbl2uni.bat command to convert all of the COBOL programs and copybooks that are found in the *PS_HOME*\src\cbl directory. After the conversion, *PS_HOME*\src\cbl Unicode contains the expanded COBOL source codes.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Compiling COBOL in Unix/Linux

Use the *PS_HOME*/install/pscbl.mak command to trigger 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:

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

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 to make these decisions.

The conversion utility expands all PIC X[(N)] data fields to triple their original size, with the following exceptions:

- Exception 1: SQL buffer setup data.
- Exception 2: Redefined character fields.
- Exception 3: Fields that appear to be dates.
- Exception 4: Arrays comprising a single character.

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 to avoid converting copybooks twice.

> **Note**
>
> The COBOL conversion utility ensures that edited lines do not go 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: SQL Buffer Setup Data

SQL buffer setup data that refers to the numeric or date data types of 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

Copyright © 1988, 2026, Oracle and/or its affiliates.

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 and 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(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).
```

In Unicode, the only fields that should be expanded are the two character fields (EMPLID and NAME). Numeric data is never affected by Unicode, and (according to the PeopleTools definition), dates are not affected 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 non-Unicode 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'.
```


Copyright © 1988, 2026, Oracle and/or its affiliates.

```
            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'.
```


Copyright © 1988, 2026, Oracle and/or its affiliates.

Exception 2: Redefined Character Fields

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 *are* expanded.

Example 1: In this example, the DB-PIC-PRECIS-CHAR is not 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 is not 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 of the character fields, including the original definition, are 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: Fields That Appear to Be Dates

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.

The following table describes the criteria that are 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 considers that a group-level field may contain REDEFINE fields. The length of the REDEFINE field is not included when determining the total length of the group field.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Example 1: The field in this example is not expanded:

```
10  START-DATE          PIC X(10)   VALUE SPACES.
```

Example 2: The fields in this example are not 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 fields in this example are not 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 fields in this example are not 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 expands 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 Comprising a Single Character

For arrays that comprise 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.

Copyright © 1988, 2026, Oracle and/or its affiliates.

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. 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 following this line (the directive line is not 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.

The following examples use existing PeopleSoft COBOL programs to illustrate possible uses for the utility directives.

NOCBGN, NOCEND, and NOCLN Directives

One of the COBOL programs for PeopleSoft Human Resources 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 that are defined in the 88-level field would ever be true.

```
NOCBGN          03  PAY-PERIODS.
    88  PAY-PERIODS-ALL             VALUE 'YYYYY'.
```


Copyright © 1988, 2026, Oracle and/or its affiliates.

```
                    88  PAY-PERIODS-ALL-BIWEEKLY    VALUE 'YYYNN'.
                    88  PAY-PERIODS-ALL-SEMIMONTHLY VALUE 'YYNNN'.
                    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 doesn't normally expand the size of the array in this line from one of the PeopleTools COBOL programs. Using 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 following example, 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.
```


Copyright © 1988, 2026, Oracle and/or its affiliates.

```
88  FIELDNAME-TSE                VALUE 'TSE_'.
03  FILLER              PIC X(14)   VALUE SPACE.
```

Viewing Error Logs

The COBOL conversion utility produces a set of error and warning logs with messages that identify nonstandard code styles and inconsistencies. The utility also logs expansion actions that may require manual review.

The utility produces the following logs:

Field or Control	Description
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 that are detected by the utility.
Summary log	This log provides general statistics regarding the number of programs that are processed.

When you specify the -4 flag, you see only the summary log. Set the -rd flag on the conversion utility command line if you want the utility to produce all of the detail logs: exception, BIND/SELECT, and exception date.

Messages from the Exception Log

The following tables summarize all of the messages that can appear in the three exception log files. Errors indicate problems that are encountered by the conversion utility.

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	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 the 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	None.
Unable to find the copy library *copy library name*.	Error	Couldn't locate the copy library file that the program references.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Messages from the Exception BIND/SELECT Log

The following table lists messages from the BIND/SELECT log:

Message	Type	Note
Didn't find the corresponding DATA section for *DATA field name* in line *line number*.	Error	Detected either a BIND-DATA or a SELECT-DATA, but cannot find the SETUP group field.
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	Couldn't locate a copy library that DATA or SETUP references.
The *group field name* found in line *line number* has a mismatch count.	Warning and 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

The following table lists messages from the exception date log:

Message	Type	Note
Date/time/datetime detected and will not be expanded in line.	Warning	None.
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.

Fine-Tuning 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.

This section discusses how to fine tune COBOL programs.

Identifying Unicode and Non-Unicode Data

A COBOL program may need to determine whether it's dealing with non-Unicode or Unicode data. For example, if the program parses a string character, it must apply different logic depending on whether the string is non-Unicode or Unicode. The program can get this information from the ENCODING-MODE-SW in the PTCSQLRT copy library (ANSI-Mode is the same as non-Unicode):

```
03  ENCODING-MODE-SW    PIC X(3)    VALUE SPACE.
                   88  ANSI-MODE                  VALUE 'A'.
                   88  UNICODE-MODE               VALUE 'U'.
```


Copyright © 1988, 2026, Oracle and/or its affiliates.

The ENCODING-MODE-SW value is set by the COBOL application programming interface (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 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 must specify the column lengths for bind or select fields before calling the PTPDYSQL program. Within a COBOL program, there are two ways that you can assign bind parameters and select buffers of dynamic SQL statements:

- By using a predefined working storage area with the dynamic SQL statement.

  This method is similar to the method that is 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 the COBOL programs.

  Because PTPDYSQL sends the correct length to PTPSQLRT, no changes to the COBOL program are necessary.

- By using a buffer array.

  At runtime, this array is partitioned based on the properties of all of the fields that are referenced by the dynamic SQL statement. The properties of those fields are retrieved from the PSDBFIELD table, and 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 that is specified for a character field. Adjust the length by a factor of three.

To adjust the length of the character field appropriately, the program must 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.

This example 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.
```


Copyright © 1988, 2026, Oracle and/or its affiliates.

```
               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
```

Related Topics

- Identifying Unicode and Non-Unicode Data
  A COBOL program may need to determine whether it's dealing with non-Unicode or Unicode data. For example, if the program parses a string character, it must apply different logic depending on whether the string is non-Unicode or Unicode. The program can get this information from the ENCODING-MODE-SW in the PTCSQLRT copy library (ANSI-Mode is the same as non-Unicode):

Defining Single Character Arrays

Some COBOL programs define single-character arrays to parse or examine a string of characters, one character at a time. In a Unicode environment, be sure that you're examining the string one character at a time, not one byte at a time.

This example shows 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
```


Copyright © 1988, 2026, Oracle and/or its affiliates.

The intent of the code in the previous example is to examine each character of the array, looking for the first delimiter character. When that character is found, the code displays the position of the delimiter.

In a non-Unicode environment that uses only the Latin1 character set, this works because there is one byte (one array element) per character. In a Unicode environment (or in a non-Unicode environment that allows double-byte character sets), this fails because what could potentially be examined is the second or third byte of a two- or three-byte character. It's 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.

To correct this situation, you must know the length (in bytes) of each character that is being processed. 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 sets and ANSI double-byte characters sets.

The PTPSTRFN subroutine offers two ways of retrieving the byte length of a character:

- By requesting the length of a single character.
- By requesting a map of an entire character string.

  Choose this option if the application program needs to get the length information of all characters within a string.

Requesting the Length of a Single Character

The input parameters to the PTPSTRFN function are:

Parameter	Value	Notes
ACTION-TYPE	ACTION-CHARLEN	None.
CHAR-CD	The character whose length you want to verify.	This variable is included in the PTCSTRFN.CBL copy library.

These values are returned:

Parameter	Value	Notes
CHAR-LENGTH	The subroutine returns one of the following values, representing the length of the character that is referenced by CHAR-CD: • ONE-BYTE • TWO-BYTES • THREE-BYTES	This variable is included in the PTCSTRFN.CBL copy library.
STRFN-RC	Returns one of the following values: • STRFN-RC-OK • STRFN-INVALID-ACTION	This variable is included in the PTCSTRFN.CBL copy library.

At the beginning of this section, 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
```


Copyright © 1988, 2026, Oracle and/or its affiliates.

```
          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 the code is modified for Unicode, it 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 modification ensures that the code continues to function properly in a Unicode environment. However, we can be sure that modification works only when the delimiter character for which the program is searching is one byte in length.

Consider the following code fragment in which the delimiter character that the program is searching for may be longer than one 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
```


Copyright © 1988, 2026, Oracle and/or its affiliates.

```
        WHEN CHAR-POS(CHAR-IDX)  =  W-DELIMITER
    SET W-OFFSET  TO  CHAR-IDX
    DISPLAY ‘FIELD DELIMITER FOUND AT POSITION ‘ W-OFFSET
END-SEARCH
```

For this code to work in a Unicode environment, a Unicode-specific search algorithm must be used. Ensure that the program always compares the correct bytes from the array (up to three bytes, based on the current character length) to the fixed three-byte field containing the search value.

The proper search method 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
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
```


Copyright © 1988, 2026, Oracle and/or its affiliates.

```
    END-IF
END-PERFORM
```

As you can see from the previous example, searching a string array for a particular value that may be an extended character can be difficult; if possible, avoid such a search.

Requesting a Map of an Entire Character String

The input parameters to the PTPSTRFN function are:

Parameter	Value	Notes
ACTION-TYPE	ACTION-STRMAP.	None.
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.	None.
String Parameter 2	The buffer area to be updated by the subroutine.	None.

This table lists the values that are returned:

Parameter	Value	Notes
String Parameter 2	This buffer is updated with the appropriate values. This field contains at least one of these values: • 1 The next character position is part of a one-byte character. • 2X The next two character positions are part of a two-byte character. • 3XX The next three character positions are part of a three-byte character.	Refer to the example following this table to see how the function works.
STRFN-RC	Returns one of the following values: • STRFN-RC-OK • STRFN-INVALID-ACTION	This variable is included in the PTCSTRFN.CBL copy library.

The following sample COBOL code provides an example of how the PTPSTRFN COBOL function can be used to map the byte length of an entire character string:

```
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.
```


Copyright © 1988, 2026, Oracle and/or its affiliates.

```
           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 ' '
```


Copyright © 1988, 2026, Oracle and/or its affiliates.

```
           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 following table provides sample Unicode text as input for the PTPSTRFN function:

Language	Sample Unicode Text
Catalan	Quan el món vol conversar, parla Unicode
Chinese (Simplified)	当世界需要⌧通⌧，⌧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
Finnish	Kun maailma haluaa puhua, se puhuu Unicodea
French	Quand le monde veut communiquer, il parle en Unicode

For each row in the table, the code performs the following functions:

- Calls PTPSTRFN to get the string mapping of the UTF-8 character string of the text that is retrieved for the UNITEXT field.
- Displays the UTF-8 string equivalent of the text.
- Tallies the number of one-byte, two-byte, and three-byte characters of the text.

The output of this program appears as follows:

> **Note**
>
> Certain strings appear to be garbled. This is because the system running the program has printed the output by individual bytes and not by multi-byte characters.

```
LANGUAGE = Catalan
   UTF8 TEXT: (LENGTH = 0041)
          Quan el mÃ³n vol conversar, parla Unicode
   UTF8 BYTE MAPPING:
          1111111112X11111111111111111111111111111
   TALLY:
     NUMBER OF ONE-BYTE CHAR:    039
     NUMBER OF TWO-BYTES CHAR:   001
     NUMBER OF THREE-BYTES CHAR: 000
 LANGUAGE = Chinese (Simplified)
   UTF8 TEXT: (LENGTH = 0043)
```


Copyright © 1988, 2026, Oracle and/or its affiliates.

```
          å½"ä¸-ç•Œé©?è¦?æ²Ÿé?šæ–¶ï¼Œè¯·ç"¨Unicodeï¼?
  UTF8 BYTE MAPPING:
          3XX3XX3XX3XX3XX3XX3XX3XX3XX3XX3XX11111113XX
  TALLY:
    NUMBER OF ONE-BYTE CHAR:    007
    NUMBER OF TWO-BYTES CHAR:   000
    NUMBER OF THREE-BYTES CHAR: 012
LANGUAGE = Chinese (Traditional)
  UTF8 TEXT: (LENGTH = 0055)
          ç•¶ä¸-ç•Œé©?è¦?æº?é?šæ™?ï¼Œè«?ç"¨çµ±ä¸?ç¢¼ï¼Unicodeï
  UTF8 BYTE MAPPING:
          3XX3XX3XX3XX3XX3XX3XX3XX3XX3XX3XX3XX3XX3XX11111113XX
  TALLY:
    NUMBER OF ONE-BYTE CHAR:    007
    NUMBER OF TWO-BYTES CHAR:   000
    NUMBER OF THREE-BYTES CHAR: 016
LANGUAGE = Danish
  UTF8 TEXT: (LENGTH = 0039)
          NÃ¥r verden vil tale, taler den Unicode
  UTF8 BYTE MAPPING:
          12X1111111111111111111111111111111111
  TALLY:
    NUMBER OF ONE-BYTE CHAR:    037
    NUMBER OF TWO-BYTES CHAR:   001
    NUMBER OF THREE-BYTES CHAR: 000
LANGUAGE = Dutch
  UTF8 TEXT: (LENGTH = 0045)
          Als de wereld wil praten, spreekt hij Unicode
  UTF8 BYTE MAPPING:
          111111111111111111111111111111111111111111111
  TALLY:
    NUMBER OF ONE-BYTE CHAR:    045
    NUMBER OF TWO-BYTES CHAR:   000
    NUMBER OF THREE-BYTES CHAR: 000
LANGUAGE = English
  UTF8 TEXT: (LENGTH = 0047)
          When the world wants to talk, it speaks Unicode
  UTF8 BYTE MAPPING:
          1111111111111111111111111111111111111111111111111
  TALLY:
    NUMBER OF ONE-BYTE CHAR:    047
    NUMBER OF TWO-BYTES CHAR:   000
    NUMBER OF THREE-BYTES CHAR: 000
LANGUAGE = Finnish
  UTF8 TEXT: (LENGTH = 0043)
          Kun maailma haluaa puhua, se puhuu Unicodea
  UTF8 BYTE MAPPING:
          1111111111111111111111111111111111111111111111111
  TALLY:
    NUMBER OF ONE-BYTE CHAR:    043
    NUMBER OF TWO-BYTES CHAR:   000
    NUMBER OF THREE-BYTES CHAR: 000
LANGUAGE = French
  UTF8 TEXT: (LENGTH = 0052)
          Quand le monde veut communiquer, il parle en Unicode
  UTF8 BYTE MAPPING:
```


Copyright © 1988, 2026, Oracle and/or its affiliates.

```
       1111111111111111111111111111111111111111111111111111
TALLY:
  NUMBER OF ONE-BYTE CHAR:    052
  NUMBER OF TWO-BYTES CHAR:   000
  NUMBER OF THREE-BYTES CHAR: 000
```


Copyright © 1988, 2026, Oracle and/or its affiliates.

7

Running COBOL in a z/OS Unicode Environment

Understanding Running COBOL in a z/OS Unicode Environment

This section discusses running COBOL in a z/OS Unicode environment.

See the product documentation for *PeopleSoft 9.2 Application Installation for DB2 for z/OS*.

Unicode Encodings in PeopleSoft COBOL on z/OS

The character set that you use for PeopleSoft COBOL processing must match the character set for your database. If you create a Unicode database for a PeopleSoft system, you must also run COBOL in Unicode.

> **Note**
>
> In this topic, the word *character* refers to a single character in any language, regardless of how many bytes are required to store the character.

The Unicode standard provides several methods of encoding Unicode characters into a byte stream. Each encoding has specific properties that make it suitable for use in different environments. Two main encodings that are important to understand are:

Encoding	Description
UCS-2	The Unicode encoding that PeopleSoft COBOL on z/OS Unicode environment uses. PeopleTools uses the encoding internally for data that is held in memory on the application server.
UTF-8	The Unicode encoding that PeopleSoft systems use in COBOL running on Unix and Microsoft Windows servers.

On Unix and Microsoft Windows servers, PeopleSoft uses UTF-8 encoding to support Unicode in COBOL. PeopleSoft uses UTF-8 since it is a superset of the US-ASCII character set, which is the native character set of these operating systems. PeopleSoft can support Unicode in these environments without requiring any special support from the COBOL compiler.

However, in the z/OS environment the native character set is EBCDIC. UTF-8 is not compatible with EBCDIC, and the COBOL runtime environment on z/OS does not recognize the data in UTF-8 correctly as it expects data encoded in EBCDIC. As a result, PeopleSoft supports Unicode COBOL on z/OS by relying on the IBM Enterprise COBOL compiler's native Unicode support, which uses UCS-2.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Related Topics

- The Unicode Standard
  The most important consideration when dealing with character sets across a system is ensuring that all characters that you plan to represent within the PeopleSoft system exist in the character set that is used by each component of the system.

Data Usage for Unicode Storage

Moving to a z/OS COBOL Unicode environment means that character data in COBOL programs, including numbers, dates and time, is stored and processed in the UCS-2 encoding of Unicode. IBM Enterprise COBOL uses special usage NATIONAL to store Unicode data, instead of usage DISPLAY.

The usage NATIONAL should be used for all storage and literal declarations, except for limited special cases. PeopleSoft has worked with IBM to ensure that all the statements that process string data can work with data fields declared with usage NATIONAL.

The PeopleSoft system provides a COBOL Unicode conversion utility for z/OS that automatically converts the data field usage for Unicode storage. When the file server is installed with the “Unicode database” option selected, the conversion utility runs behind the server transfer process, and JCL files pre-configured for compiling Unicode COBOL programs are copied to the server. The Unicode support is provided out-of-the box, with no additional configuration required.

Unlike Unicode COBOL on Unix and Microsoft Windows, there is no need to explicitly expand storage size for Unicode processing in the z/OS Unicode environment. To store ten characters on Db2 for z/OS, declare the database column as VARGRAPHIC(10). This column can be fetched into PIC N(10) field in Unicode COBOL programs running on z/OS.

Related Topics

- Running the COBOL Unicode Conversion Utility for z/OS
  This section discusses running the PeopleTools COBOL Unicode conversion utility for z/OS.

Input/Output in the z/OS Unicode Environment

Even when character data in a PeopleSoft z/OS COBOL program is stored and processed internally in Unicode, there are some cases where data is needed in a native non-Unicode character set of the system. This is especially true for the purpose of input and output operations, and because external systems may not fully support Unicode.

Database Input/Output

PeopleSoft uses COBOL mainly for high-performance SQL batch jobs. Db2 for z/OS supports communication with COBOL programs in Unicode using GRAPHIC and VARGRAPHIC data types.

All PeopleSoft-delivered COBOL programs are designed to communicate through the single central database access program PTPSQLRT. This program is enhanced to support communication with databases using Unicode.

Copyright © 1988, 2026, Oracle and/or its affiliates.

> **Note**
>
> You may need to manually modify custom COBOL programs if they do not use PTPSQLRT or PTPDYSQL for database communication. The PeopleSoft-provided COBOL Unicode conversion utility for z/OS does not update SQL communication logic automatically.

Logging

PeopleSoft COBOL programs output logs using DISPLAY statements. The logs are mainly for displaying informational and debug messages. As these may need to be displayed on z/OS terminal screens which cannot display Unicode data, the PeopleSoft COBOL Unicode conversion utility inserts the DISPLAY-OF() COBOL intrinsic function to convert data to the system EBCDIC character set in the context of the DISPLAY statement.

> **Note**
>
> Because logs are generated in the system EBCDIC code page, not all the Unicode characters display on the log file. Characters not included in the system code page will be lost. Do not attempt to display international characters on the COBOL log. For reporting purpose, use the Structured Query Report (SQR) language instead of COBOL.

File Processing

A small number of PeopleTools COBOL programs use file input/output using READ and WRITE statements, for parameter input from JCL or writing logs. As these input/output can be better done using the system EBCDIC character set rather than Unicode, the PeopleTools COBOL programs performing file input/output are updated to convert to and from EBCDIC when accessing files. Because of the very limited use of file input/output, the COBOL Unicode conversion utility does not perform any special work for file input/output.

> **Note**
>
> Use Application Engine for processes that involve file input and output.

Miscellaneous Input/Output Operations

There are other cases where you should use the EBCDIC character set for string data, such as when accessing environment variables. PeopleSoft delivered programs are modified to convert internal Unicode representation of strings just before passing it to external data area, or convert from EBCDIC to internal Unicode representation, when required. However, the COBOL source conversion utility will not automatically update them. If you have custom COBOL programs that access external data, you may need to test them thoroughly after running the conversion utility, and insert DISPLAY-OF() or NATIONAL-OF() COBOL intrinsic functions where data conversions are required.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Related Topics

- Running the COBOL Unicode Conversion Utility for z/OS
  This section discusses running the PeopleTools COBOL Unicode conversion utility for z/OS.

Understanding the COBOL Unicode Conversion Utility for z/OS

This discusses the PeopleTools COBOL Unicode conversion utility for z/OS and applying patches.

COBOL Unicode Conversion Utility for z/OS

As delivered by PeopleSoft, all COBOL programs are written to run on non-Unicode databases. To run the PeopleSoft-delivered COBOL on z/OS against a Unicode database, you must use the PeopleTools COBOL conversion utility for z/OS.

This utility is typically called automatically by the PeopleSoft server transfer process; however in certain circumstances, such as when you adapt COBOL code or apply a PeopleSoft-provided patch to a COBOL program, you may need to run the utility manually.

All internal data definitions for character-type data in COBOL programs, including data field groups and literals, must have usage NATIONAL instead of usage DISPLAY. The conversion utility changes the usage of the applicable data fields and also makes related changes in procedure division of the COBOL program source.

Applying Patches

Adapt and apply patches to only one set of COBOL source code—non-Unicode source. It is much easier to write COBOL programs without having to remember to use national data items as you go. Once your adaptation or patch is complete and you are ready to compile the program, first run it through the COBOL Unicode conversion utility, then compile it. This approach has several benefits over customizing the converted code:

- You maintain a single source tree for all of your COBOL—the non-Unicode source.

  This way you don't run the risk of accidentally adapting both the non-Unicode COBOL programs and the Unicode-converted COBOL programs and potentially losing the modifications to the converted programs the next time you run the converter.

- Although PeopleSoft tests all delivered COBOL programs and patches in both Unicode and non-Unicode environments, only non-Unicode versions of the source are delivered.

  Therefore, any time you apply a PeopleSoft COBOL patch to a Unicode system, you must run the patched source code through the COBOL converter. If you had already modified the post-converted source, converting again would obliterate your modifications.

Under very limited conditions, you may still need to customize your COBOL source code after the conversion and maintain the Unicode version of the source. In this case, you may need to manually apply patch delivered by PeopleSoft to your source code.

Running the COBOL Unicode Conversion Utility for z/OS

This section discusses running the PeopleTools COBOL Unicode conversion utility for z/OS.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Automatically Running the COBOL Unicode Conversion Utility for z/OS

By default, the PeopleTools COBOL Unicode conversion utility for z/OS runs in the background of the server transfer process if you select the *Unicode database* option during installation.

Manually Running the COBOL Unicode Conversion Utility for z/OS

You may want to run the utility manually in situations such as when converting PeopleSoft-delivered COBOL program patches or converting custom COBOL programs.

Use the following command syntax to run the COBOL Unicode conversion utility on the Microsoft Windows file server:

```
            PS_HOME\bin\client\winx86\pscblucvrtz.exe -s:Source_Directory -
t:Destination_⇒
Directory -r:Log_Directory [-psfs]
```

The following table provides a description of the commands:

Command	Description
-s: *Source_Directory*	Specify the source directory where the non-Unicode version of COBOL resides. For the directory, you must specify where the COBOL subdirectories reside (\BASE, \MVS, and so on.), unless you specify –psfs flag to process all the PeopleSoft COBOL source subdirectories at once. Example: `-s:d:\PT8\SRC\CBL` **Note:** The utility processes COBOL source under *Source_Directory*\Unicode, but it does not modify the source found under the \Unicode subdirectory. The \Unicode subdirectory may be used to maintain the customization to the source already converted to Unicode.
-t: *Target_Directory*	Specify where you want to place the converted version of COBOL. The utility puts the modified source file in the same COBOL subdirectory in which it was found when you specified –psfs flag. Example: `-t:d:\PT8\SRC\CBLUNICODEZ`
-r: or –rd: *Log_Directory*	-r generates only the summary log file; -rd generates all of the log files. See Using Error, Exception and Summary Logging.
-psfs	Specify when you are processing conversion against PeopleSoft file server directory. If this flag is specified, the conversion utility processes both \BASE and \MVS subdirectories under the source directory.

The utility produces a new source file for each .CBL file that is found. These new files are placed under the specified target directory. If you specify the –psfs flag, the files are placed in \BASE and \MVS subdirectories under the target directory.

Copyright © 1988, 2026, Oracle and/or its affiliates.

When you manually run the conversion utility you must transfer the generated .CBL files under the target directory to SRCLIB and COPYLIB data sets on the z/OS server, and compile the programs by submitting JCLs.

> **Note**
>
> The PeopleSoft COBOL conversion program for the Unicode environment on Unix and Microsoft Windows is named pscblucvrt. The COBOL conversion program for the z/OS environment is named pscblucvrtz. Do not use pscblucvrt for COBOL programs running on z/OS because the conversion logic used for z/OS COBOL programs is completely different from that used for Unix and Microsoft Windows.

Identifying Converted COBOL Programs

When the COBOL Unicode conversion utility for z/OS runs, it places a comment at the beginning of each COBOL program that it converts:

```
******* Converted for Unicode (national support)  (date converted)
******************** DO NOT MODIFY THIS FILE *******************
**** ALL SOURCE CODE CHANGES MUST BE DONE TO THE ANSI VERSION ****
******************** FOUND IN %PS_HOME%/SRC/CBL ******************
```

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 converting this program file again.

> **Note**
>
> Conversion utility comments for the z/OS Unicode environment include “(national support)” on the first line. The first line of comments generated by the COBOL conversion utility for Unix and Microsoft Windows environments read “Converted for Unicode”.

Understanding Converted Data

The PeopleTools COBOL Unicode conversion utility for z/OS reads COBOL sources under the source directory, analyzes their structures and statements, and applies a set of conversion rules to each statement.

Copybook Conversion

The utility converts copybooks on the fly: the first time that a copybook is referenced inside data division or procedure division of any program, it is processed immediately. The utility processes an entire set of COBOL modules in a single run. It maintains a record of what it has converted to avoid converting copybooks twice.

Copyright © 1988, 2026, Oracle and/or its affiliates.

The utility assumes that the COBOL programs are using PeopleSoft-standard COBOL program naming conventions, including that program sources are named “??P?????.cbl” and copybook sources are named “??C?????.cbl” The convention ? means any single character.

> **Note**
>
> The PeopleTools COBOL Unicode conversion utility for z/OS is designed to process an entire set of COBOL modules, including all programs and copybooks. The utility processes the COBOL COPY statement and checks for the existence of a copybook called from the program. If a copybook called from a program cannot be found in the source directory, the conversion process stops with an error, since the content of the copybook may affect the conversion result of the calling program.

Data Division Conversion Rules

All data field declarations with usage DISPLAY are converted to usage NATIONAL.

- A PIC X field is converted to a PIC N field.
- A string literal in the context of PIC N field is prefixed with N symbol.
- A PIC 9 field whose usage is DISPLAY is added USAGE NATIONAL clause to make its usage NATIONAL.
- A group that all of its members can be converted to usage NATIONAL is added GROUP USAGE NATIONAL clause to make the group NATIONAL item.

Example 1: PIC X Field Converted to PIC N and String Literal Prefixed with N Symbol

The following example shows the PIC X field converted to PIC N. In addition the prefix N is added to the VALUE clause.

```
02  RECNAME-TBLB           PIC X(9)    VALUE 'APPL_TBLB'
```

Is converted to:

```
02  RECNAME-TBLB           PIC N(9)    VALUE N'APPL_TBLB'
```

Example 2: Usage NATIONAL clause added to PIC 9 Field

The following example shows the Usage NATIONAL clause added to the PIC 9 field.

```
02  CURR-COUNT             PIC 9(4)
```

Is converted to:

```
02  CURR-COUNT             PIC 9(4) USAGE NATIONAL
```

Example 3: GROUP-USAGE NATIONAL Clause Added to Group W-LIT

The utility adds the GROUP-USAGE NATIONAL clause to the group W-LIT, because all the members of the group can be converted to the national data type. W-CALC-FIELDS is not added GROUP-USAGE NATIONAL because it includes item that cannot be converted to

Copyright © 1988, 2026, Oracle and/or its affiliates.

national data type. Packed decimal, binary, pointer items are not converted to national type, and groups including these data types are not added GROUP-USAGE NATIONAL.

```
01  W-LIT GROUP-USAGE NATIONAL .
02  PROGRAM-NAME            PIC N(8)    VALUE N'PTPDEC31'.
02  JOBID                   PIC N(10)   VALUE N'PTPDEC31'.
02  DESCR                   PIC N(16)   VALUE
N'TEST DEC31'.
01  W-CALC-FIELDS.
02  INIT-CHAR-DEF           PIC N(10).
02  COMP-REDEF-OF-CHAR REDEFINES INIT-CHAR-DEF
PIC S9(8) COMP.
```

Exceptions to Data Division Conversion Rules

There are a few exceptions to the data division rules to make the program compile and work correctly. For the utility to recognize these exceptional cases, strict adherence to the PeopleSoft COBOL coding standards is required. The utility looks for certain code-style patterns to make these decisions.

This section discusses the following areas where there are exceptions to data division rules:

- SQL buffer setup area.
- SQL buffer data area.
- File status field.
- FD entry in file section.

SQL Buffer Setup Area

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 PTPSQLRT is designed to process the descriptors as alphanumeric character arrays (instead of national character arrays), the conversion utility does not convert the data type of the descriptor fields. However, it doubles the size of the descriptor field that is representing character-type data. Each data field in the descriptor area has associated field in SELECT-DATA or BIND-DATA field, and the byte size of the corresponding field in –DATA area doubles because of the change from PIC X to PIC N. Because of this implicit field size expansion, the corresponding field in –SETUP area should be doubled in size. Example: The following BIND-SETUP area is converted to match with BIND-DATA in byte size. Character fields are converted to double of the original size, to match with the size of the field in BIND-DATA area. Numeric and delimiter (‘Z’) items are not converted.

```
05  BIND-SETUP.
10  FILLER              PIC X(22)   VALUE ALL 'C'.
10  FILLER              PIC X(8)    VALUE ALL 'H'.
10  FILLER              PIC X(02)   VALUE ALL 'S'.
10  FILLER              PIC X(01)   VALUE 'Z'.

05  BIND-DATA.
10  EMPLID              PIC N(11).
10  ACAD-CAREER         PIC N(04).
```


Copyright © 1988, 2026, Oracle and/or its affiliates.

```
10  STDNT-CAR-NBR-REAL  PIC S9(04)              COMP.
10  FILLER              PIC N(01)   VALUE N'Z'.
```

SQL Buffer Data Area

In the SQL buffer data area the conversion utility does not convert character data that is redefining numeric data. By convention in PeopleSoft application COBOL programs, this type of redefinition is used only for the purpose of using one data area for different number of variables. This type of character field can be in PIC X type that the conversion utility will not change the data type to PIC N.Example: PAGE-NO-FILLER, which is redefining binary field and in SQL buffer data area, is not converted to PIC N.02 BIND-DATA.

```
03  COMPANY             PIC N(10).
03  PAYGROUP            PIC N(10).
03  PAY-END-DT          PIC N(10).
03  OFF-CYCLE           PIC N.
03  PAGE-NO             PIC 9999                COMP.
03  PAGE-NO-FILLER REDEFINES PAGE-NO
                        PIC XX.
03  FILLER              PIC N       VALUE N'Z'.
```

File Status Field

The conversion utility does not convert a file status field (that includes “FILE-STAT” in name), because file status field (specified in FILE STATUS clause of FILE-CONTROL paragraph) must have the form of two-character alphanumeric or numeric item. It cannot be two-character national item.

Example: FILE-STAT-CTLFILE is not converted because it is file status field.

```
INPUT-OUTPUT SECTION.

FILE-CONTROL.

     SELECT CTLFILE  ASSIGN TO UT-S-CTLFILE
     FILE STATUS IS FILE-STAT-CTLFILE.
DATA DIVISION.
WORKING-STORAGE SECTION.

01  PROGRAM-IDENTITY       PIC N(8)    VALUE N'PTPCTL'.
01  FILE-STAT-CTLFILE      PIC XX      VALUE '00'.
01  MORE-INPUT             PIC N(1)    VALUE SPACES.
```

FD Entry in File Section

A line containing the RECORD CONTAINS clause in the FD data field will be commented out. If this line is not commented out, the program may cause an error at compile time because the record field is normally converted to a PIC N field and the record size will not match with the size specified in RECORD CONTAINS clause.

Copyright © 1988, 2026, Oracle and/or its affiliates.

> **Note**
>
> This conversion eliminates compile errors. To read and write files either in Unicode or in EBCDIC, you may need to further modify the program manually to guarantee the correct file operation.
>
> Example: Line containing “RECORD CONTAINS” clause is commented out in FD field declaration.

```
FD  TESTFILE
*   RECORD CONTAINS 80 CHARACTERS
     RECORDING MODE IS F
     LABEL RECORDS OMITTED.
01   TESTFILE-RECORD              PIC N(80).
```

Procedure Division Conversion Rules

In procedure division, the COBOL Unicode conversion utility for z/OS applies the following conversion rules to ensure that the correct operations on data fields convert to usage NATIONAL. These conversion rules are based on statement type.

Statement(s)	Conversion Rule Description
• ADD • DIVIDE • EVALUATE • WHEN • ENTRY • IF • INITIALIZE • INSPECT • INVOKE • MULTIPLY • PERFORM • SEARCH • SET • STOP • STRING • SUBTRACT • UNSTRING • EXEC	Add “N” in front of every literal in the context of these statements.
DISPLAY	Surround each national data item in its context with FUNCTION DISPLAY-OF(), so that EBCIDIC data is printed on the log. If the data item is national numeric item (PIC 9 USAGE NATIONAL), add FUNCTION DISPLAY-OF() and reference modification to the item.
• CALL • CANCEL	Add FUNCTION DISPLAY-OF() if the first parameter is NATIONAL data field.
COMPUTE	Add FUNCTION DISPLAY-OF() to the parameter of FUNCTION NUMVAL().

Copyright © 1988, 2026, Oracle and/or its affiliates.

Example 1: Prefix “N” Added to Literals in the Context of STRING Statements

IBM Enterprise COBOL compiler complains if both national and non-national items are contained in the context of some statements, including STRING, UNSTRING, and INSPECT. Since SQL-STMT of ICST-DYSQL is a national character field, the statement cannot be compiled unless we add “N” prefix to the literals.

```
STRING
     N'%ROUND(TLP_COST,2), ' DELIMITED BY SIZEN'%ROUND(LLP_COST,2), '
DELIMITED BY SIZE
     INTO SQL-STMT OF ICST-DYSQL
     WITH POINTER SQL-STMT-LEN OF ICST-DYSQL
     ON OVERFLOW PERFORM ZS000-STRING-ERROR
END-STRING
```

Example 2: Insert DISPLAY-OF() Intrinsic Function Around National Data Items

The conversion utility inserts DISPLAY-OF() intrinsic function around national data items in the context of DISPLAY statements, so that the data is converted to EBCDIC.

The DISPLAY-OF() function only accepts the PIC N parameter. To convert the PIC 9 USAGE NATIONAL item to non-Unicode using the DISPLAY-OF() function, you must add a reference modification (in the following example (1:)) to the PIC 9 USAGE NATIONAL ITEM, so that the item is first converted to a PIC N item.

```
DISPLAY 'RUNID= ' FUNCTION DISPLAY-OF(RUNID OF KBDPR)
DISPLAY 'OPRID= '  FUNCTION DISPLAY-OF(OPRIDX OF KBDPR)
DISPLAY 'RUNID OF SQLRT=' FUNCTION DISPLAY-OF(BATCH-RUN-ID OF SQLRT)
DISPLAY 'PROCESS-INSTANCE = ' FUNCTION
DISPLAY-OF(PROCESS-INSTANCE OF KBDPR(1:))
```

Example 3: Add FUNCTION DISPLAY-OF() if the First Parameter is National Data Field

If the first parameter of a CALL statement, which specifies the called subprogram, is a national data item, the conversion utility adds the DISPLAY-OF() function to the parameter, because the national data item cannot be used for the CALL parameter.

```
CALL FUNCTION DISPLAY-OF(COBOL-PROG OF W-WK) USING SQLRT
                                                   LOGMS
                                                   BMSET
                                                   DYSQL
                                                   SPARM
                                                   SBIND
                                                   SCACH
                                                   ECURS
```


Copyright © 1988, 2026, Oracle and/or its affiliates.

Example 4: Add FUNCTION DISPLAY-OF() to the Parameter of FUNCTION NUMVAL()

If the data field passed to function NUMVAL() is national data type, the conversion utility adds function DISPLAY-OF() to convert the data item to EBCDIC data, as NUMVAL() function cannot process national data.

```
COMPUTE W-COMPARE-GPA =
            FUNCTION NUMVAL ( FUNCTION DISPLAY-OF(W-CONDITION-DATA
))
```

Columns

The COBOL Unicode conversion utility ensures that edited lines do not go past the column 72. If the conversion would normally cause a line to exceed column 72, the utility inserts a line break before column 72 to make the long line into two separate lines.

The COBOL Unicode conversion utility for Microsoft Windows and Unix environments accepts various directives in the first six columns of COBOL code to that COBOL programs can use to tailor the conversion for specific code. The COBOL Unicode conversion utility for z/OS does not use any directive, therefore, the above rules applies to all the COBOL code.

Using Error, Exception and Summary Logging

This section discusses using error. exception and summary logging.

Understanding Error, Exception and Summary Logs

The COBOL Unicode conversion utility for z/OS produces a set of error and informational logs with messages that identify inconsistencies and problems. The utility also logs conversion exceptions that may require manual review.

Log	Description
Error log	This log is produced when serious error occurred in the conversion, and provides detailed information of the error.
Exception log	This log lists all conversion exceptions found in the programs.
Summary log	This log provides general statistics regarding the number of programs that are processed.

Setting Logging Parameters

To set logging parameters, set the following parameters on the conversion utility command line:

Parameter	Description
-r	Produces the summary log.
-rd	Produces the summary log and the exception log.

Copyright © 1988, 2026, Oracle and/or its affiliates.

When called from the server transfer process, the conversion utility produces the summary log in the staging directory, but does not produce an exception log.

Viewing Exception Log Messages

The following table summarizes all of the messages that can appear in the exception log file.

Message	Note
File Name: Not converted because this is Unicode file.	COBOL source that is already converted (has “Converted for Unicode” marker on the first line), or COBOL source files under \unicode subdirectory under the source directory, will not be converted.
File Name: Not converted because it is not used in DATA or PROCEDURE division of any program.	The utility converts copybooks on the fly when they are used in COPY statement in the data division or procedure division of the program. Copybooks not used from data division or procedure division of program sources will not be modified.
File Name(*Line Number*): *Field Name* not converted because it is file status field.	The utility does not convert file status field because COBOL requires file status field to be a two-byte alphanumeric field.
File Name(*Line Number*): Size of *Field Name* expanded but not converted to PIC N because it is -SETUP data.	In the SELECT-SETUP or BIND-SETUP group, the utility does not change data field usage. Instead, character data field size is expanded to match the size of the corresponding PIC N field in the SELECT-DATA or BIND-DATA group.
File Name(*Line Number*): *Field Name* not converted because of redefinition.	In the SELECT-DATA or BIND-DATA group, the character field redefining binary, packed decimal, or pointer data field will not be converted to PIC N.
File Name(*Line Number*): Removed GROUP-USAGE form group *Group Field Name*	The utility removes the GROUP-USAGE clause from a group declaration if the group is used in the context where GROUP-USAGE NATIONAL cannot be used. This message typically appears when a group in a copybook can be USAGE NATIONAL in the context of one program, but later found that it cannot be NATIONAL in the context of another program.
File Name(*Line Number*): Data item *Field Name* not found.	The utility checks the data field when processing procedure division of COBOL programs, typically to determine if it should add the DISPLAY-OF() function. This message appears on the log when the utility failed to look up the data field.

Viewing Summary Log Messages

The summary log lists the number of programs and copybooks read, modified, and not modified. The following is an example of the summary log.

If you are converting PeopleSoft- delivered application program sources, it is typical that not all the programs and copybooks are modified. The utility converts a copybook when it is called from data division or procedure division of any program source using COPY statement. There are several PeopleSoft delivered copybooks that are not called from data division or procedure

Copyright © 1988, 2026, Oracle and/or its affiliates.

division of any program. PeopleSoft also provides Unicode-specific COBOL source code that the utility does not modify and counts as a “not modified” program or copybook.

Fine-Tuning COBOL Programs for the z/OS Unicode Environment

This section discusses fine tuning COBOL programs for the z/OS Unicode environment.

Understanding Fine-Tuning Converted COBOL Programs for the z/OS Unicode Environment

Although the COBOL Unicode conversion utility for z/OS makes most of the changes that are needed to run COBOL in a Unicode environment on z/OS, some manual fine-tuning may still be necessary.

Identifying Unicode Data for z/OS, Unicode Data for Microsoft Windows/Unix and Non-Unicode Data

A COBOL program may need to determine whether it’s dealing with non-Unicode data, Unicode data for Microsoft Windows/Unix in UTF-8 format, or Unicode data for the z/OS environment in UCS-2 format.

The conversion utility can get this information from the ENCODING-MODE-SW data field in the PTCSQLRT copy library. ANSI-mode is the same as non-Unicode. Unicode-mode represents Unicode processing on Microsoft Windows/Unix using UTF-8, and National-mode represents Unicode processing on z/OS using UCS-2.

```
03  ENCODING-MODE-SW    PIC X(1)    VALUE SPACE.
88  ANSI-MODE                       VALUE 'A'.
88  UNICODE-MODE                    VALUE 'U'.
88  NATIONAL-MODE                   VALUE 'N'.
```

The ENCODING-MODE-SW field value is set by default during installation. If when installing the file server you select the Unicode database option and build the PeopleSoft Process Scheduler environment on z/OS by running the PeopleSoft server transfer process, the value is automatically set to National-mode in the z/OS Unicode environment.

> **Note**
>
> It is very important to distinguish Unicode processing on Microsoft Windows/Unix from Unicode processing on z/OS. Unicode processing on Microsoft Windows/Unix requires expanding storage size by explicitly updating the field length. In the z/OS environment, Unicode values are stored using different data type (national), and you need not expand the storage size explicitly.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Understanding Character Fields and Byte Size

The COBOL Unicode conversion utility for z/OS changes the usage of alphanumeric character fields (PIC X) to usage NATIONAL (PIC N). This change works for the majority of the data fields, but may have side effects.

As in UCS-2 encoding, any character takes up two bytes. The size of one PIC N element is two bytes. The conversion from PIC X to PIC N implicitly doubles the field size. This can be a problem when a field must be a specific byte size. For example, in the following example, the PIC X field is used as filler to extract a portion of byte stream from numeric data field. This type of PIC X field should not be converted to PIC N.

```
01  W-WORK.

           03  NUM-OUT-AREA            PIC X(16).
           03  NUM-OUT-0               REDEFINES NUM-OUT-AREA
                                       PIC S9(31) COMP-3.
           03  NUM-OUT-1               REDEFINES NUM-OUT-AREA
                                       PIC S9(30)V9(1) COMP-3.
           03  NUM-OUT-2               REDEFINES NUM-OUT-AREA
                                       PIC S9(29)V9(2) COMP-3.
           03  NUM-OUT-3               REDEFINES NUM-OUT-AREA
                                       PIC S9(28)V9(3) COMP-3.

           03  FILLER                  REDEFINES NUM-OUT-AREA.
               05  FILLER              PIC X(12).
               05  NUM-OUT-INT         PIC S9(8) COMP.
           03  FILLER                  REDEFINES NUM-OUT-AREA.
               05  FILLER              PIC X(14).
05  NUM-OUT-SMALLINT    PIC S9(4) COMP.
```

The utility converts all alphanumeric character fields to PIC N, except for a very limited case where it can detect it should not convert this field data type. For the cases like that shown in the previous example, you must manually modify the code and test it.

Additional Considerations

Consider the following:

- Be very careful when deciding not to convert a PIC X field. COBOL does not allow moving data between PIC X and PIC N field without character set conversion between EBCDIC and Unicode, which adds extra complexity in your program and can cause performance overhead.
- You cannot use national and non-national data in some statements, such as STRING and UNSTRING.
- PIC N is two-byte field but content is not limited to Unicode strings. You can store binary, packed decimal or pointer values in the PIC N field (using redefinition) and can copy the value to another PIC N field. Therefore, it is not always necessary to use PIC X field to store non-Unicode values. Use the PeopleSoft-delivered COBOL Unicode conversion utility for z/OS to convert all fields to PIC N unless the field cannot be converted to PIC N (instead of trying to convert only the field that can contain international characters).

Copyright © 1988, 2026, Oracle and/or its affiliates.

Understanding Character Set Conversion

The COBOL Unicode conversion utility for z/OS updates COBOL programs to use Unicode for all character data for internal processing, including numbers and date. However, there maybe cases where the Unicode data must be converted to non-Unicode character set, typically when communicating with external systems that do not fully support Unicode.

The following table lists IBM Enterprise COBOL statements and functions that convert between Unicode and non-Unicode character sets.

Statement/Function	Description
MOVE	This statement converts data in system EBCDIC character set to Unicode in the following syntax: `move PIC X field or alphanumeric string literal ⇒ to PIC N field` **Note:** Using the statement `move PIC N field or national literal to PIC X` causes a compiler error. When converting from Unicode data item to non-Unicode, you must insert the DISPLAY-OF() function to convert the Unicode data to non-Unicode first.

DISPLAY-OF()	This function converts data from Unicode to a non-Unicode character. The DISPLAY-OF() function has two forms:
	• The following converts from Unicode to system EBCDIC character set.
	
	`FUNCTION DISPLAY-OF(PIC N field or national⇒ literal)`
	
	• The following converts from Unicode to a character set specified in the CCSID parameter.
	
	`FUNCTION DISPLAY-OF(PIC N field or national⇒ literal, CCSID)`

Copyright © 1988, 2026, Oracle and/or its affiliates.

Statement/Function	Description

NATIONAL-OF()	This function converts data from a non-Unicode character set to Unicode. NATIONAL-OF() function has two forms:
	• The following converts from the system EBCDIC character set to Unicode.
	`FUNCTION NATIONAL-OF(PIC X field or alphanumeric⇒ literal)`
	• The following converts from a character set specified in the CCSID parameter to Unicode.
	`FUNCTION NATIONAL-OF(PIC X field or alphanumeric⇒ literal, CCSID)`

Note

To change the newline characters, use one of transformations - NewLine LF, NewLine CR, or NewLine CRLF at the end of the command. This utility is available to use in the bin directory.

Copyright © 1988, 2026, Oracle and/or its affiliates.

8

Sorting in PeopleTools

Understanding Sort Orders

This section discusses sorting.

Sorting Overview

Sorting data in English is reasonably simple given the well-defined sorting rules of the language. Additionally, most character sets are based on the ASCII standard, which allocates characters to numerical codes in English alphabetical order. Therefore, when sorting ASCII data by its binary representation, you automatically get a sort that makes sense in English; data is sorted from *A* to *Z* and numerics are sorted from 0 to 9.

However, sorting non-English languages is significantly more complex; some languages have special rules for sorting characters with diacritic marks; others, such as Japanese and Thai, can be sorted in several different orders depending on the usage or context of the sort.

In linguistic circles, sorting is also known as *collation.* In this book, the two terms are used interchangeably.

A sort order identifies how PeopleTools assembles, compares, and displays data. For example, a sort order specifies whether *A* is less than, equal to, or greater than *Z.* The simplest way of sorting data in a computer system is to sort it in the order that the characters appear in the character set. This is known as a *binary sort,* because it sorts the numerical codes of each character as they are stored in memory without any special sorting cases or linguistic considerations. A binary sort works well for sorting English language data; English sorting rules can be implemented as a binary sort as long as the underlying character set is laid out from *A* to *Z.* English characters in US-ASCII, EBCDIC, and Unicode are all laid out in this fashion, so a binary sort on data in any of these character sets is sufficient for sorting English data.

However, when sorting data in languages other than English, you must consider how to sort:

- Characters with diacritic marks (such as á, ñ, and ö): after the letter Z, or after the base character without the diacritic.

  In most character sets, these characters appear after the letter *Z* in binary order; however, in most languages they sort after the base form of the character.

- Special characters and ligatures (such as æ and œ) and representative characters (such as ß) in some European languages.

  In most cases, these characters must be expanded to their full form before being sorted. For example, æ is often expanded to *ae* when sorted.

- Ideographic languages.

  The Chinese, Japanese, and Korean languages use a large repertoire of characters in their written languages, such that it would be impossible for the average person to remember an arbitrary sort order. Instead, several schemes exist for sorting Chinese, Japanese, and Korean characters, including sorting based on core, common parts of each

Copyright © 1988, 2026, Oracle and/or its affiliates.

character (radicals), or by counting the number of brush or pen strokes that it takes to write the character (stroke count).

Unfortunately, most of these sorting schemes are language-specific and sometimes even specific to a country in which a language is spoken. For example, the German sharp-S character (ß) is sorted in Germany as if it were written as *SS,* but in Austria it is sorted as if it were written as *SZ.* Other times, multiple sort orders can be in use in a single country. In Spain, it is common to sort the *ch* character sequence after *cz* but before *d.* However, in more recent times many Spanish organizations have reverted to sorting the *ch* sequence as individual characters between *cg* and *ci.* Which is correct depends on personal or organizational preference.

In the majority of cases where data is sorted in PeopleTools, the host database management system performs the sort through an ORDER BY clause in SQL, and the result is displayed directly to the user. It is therefore critical, when you create a database, that you select an appropriate sort order based on the languages that you plan to maintain in the database, the countries in which you plan to operate, and any specific preferences or policies that your organization maintains relating to sorting (such as which Spanish *ch* sort is preferred). Most database systems require you to choose a sort order during database creation, because it affects the way that SQL indexes are stored on disk to optimize sorting performance.

> **Note**
>
> Due to limitations in most database packages and for performance reasons, PeopleTools supports only one sort order per database.

In some cases, PeopleTools also sorts data in memory and must use internal tables to determine the appropriate order for character data. This is particularly prevalent when sorting drop-down lists on pages displaying translate values and within PeopleSoft Query. It is therefore important that you not only create the database with the appropriate sort order, but you also tell PeopleTools which sort order you have chosen for the database so it can emulate this sort for lists that it sorts in memory.

> **Note**
>
> Sorting in PeopleTools is case sensitive (for example, the lowercase letter *a* sorts after the uppercase letter *Z*) and accent sensitive (for example, the accented letter *á'* is considered distinct from the unaccented letter *a*). Sorting in PeopleTools is also kana sensitive for Japanese data, meaning that certain forms of Japanese characters (Hiragana and Katakana) are considered distinct rather than equivalent. PeopleTools does not support case-insensitive, accent-insensitive, or kana-insensitive sorting.

PeopleTools Sorts

Many different components of PeopleTools sort character data, most of which rely on the database system to perform sorting by using a SQL ORDER BY clause. However, PeopleTools does perform some sorting in memory.

See PeopleTools In-Memory Sorts.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Other parts of PeopleTools support only a binary sort for logistical or performance reasons. The following table indicates each of the common areas of PeopleTools that presents sorted lists of data to the user and the mechanism that each uses to perform the sort:

Functional Area	Sorting Engine
Component search dialog boxes.	Database-level SQL ORDER BY sort.
Scroll buffers.	Database-level SQL ORDER BY sort.
SQL operations in PeopleSoft Query, PS/n Vision, and SQR.	Database-level SQL ORDER BY sort.
SQL functions that are called from PeopleCode (SQL Objects, ExecSQL) containing BETWEEN, <, >, MIN, MAX, and so on.	Database-level SQL ORDER BY sort.
PeopleCode ScrollSelect( ) / ScrollSelectNew( ), RowScrollSelect( ), and RollScrollSelectNew( ) functions.	Database-level SQL ORDER BY sort.
Column-specific re-sorts in PeopleSoft Pure Internet Architecture (by clicking on a column heading).	PeopleTools in-memory sort.
PeopleCode SortScroll( ).	PeopleTools in-memory sort.
Drop-down list boxes of translate values.	PeopleTools in-memory sort.
PeopleCode binary comparison functions (<, >, =).	Binary sort.
%BINARYSORT meta-SQL functions.	Binary sort.
Greater than and less than COBOL operations.	Binary sort.
Greater than and less than comparisons in SQR.	Binary sort.

Database-Level SQL ORDER BY Sorts

PeopleTools relies on the sorting capabilities of the host database system for each functional area that is identified in the table in the previous section as using a database-level SQL ORDER BY sort. In this case, the SQL statement that is used to retrieve the data to be displayed to the user is coded to include an ORDER BY statement, and the sort order of the database determines in what order data is retrieved. Obviously, this relies on the database system being correctly configured for the appropriate linguistic sort that you determined is required for your database.

Each database management system has a different mechanism of determining the sort order for SQL ORDER BY statements. For example, Microsoft SQL Server requires the user to select a sort order when the server is installed, or when each database is created. Oracle enables the database administrator to specify the appropriate sort order in the init.ora parameter file.

See the *PeopleSoft 9.2 Application Installation* for your database platformand the product documentation for your database delivered by your database vendor.

PeopleTools In-Memory Sorts

To avoid round-trips to the database, PeopleTools performs some sorting in memory, typically of short lists, such as a list of translate values. PeopleTools supports a significant number of linguistic sorts for in-memory sorting. The sort order that is used for PeopleTools in-memory sorting is controlled by the sort order option on the PeopleTools Options page. While PeopleTools cannot emulate exactly each sort order that is offered by your database vendor, it

Copyright © 1988, 2026, Oracle and/or its affiliates.

provides an appropriate sort order for most popular business languages. Select the sort order that most closely corresponds to the sort order of your database. Conflicting sort order selection between the database and PeopleTools results in lists of values that are sorted by PeopleTools appearing in a significantly different order than lists of values that are sorted by the database.

See Setting the Sort Order.

Binary Sorts

When comparing values in PeopleCode, SQR, and COBOL that use less than (<), greater than (>), or other character comparison operators, a binary sort is used. The specific sort that is performed depends on the character set of data in memory at the time. For example, all binary sorts that are performed in PeopleCode, SQR syntax, and Unicode COBOL take place based on the Unicode binary order. Binary sorts that are performed in non-Unicode COBOL take place in the non-Unicode character set of the batch server. Binary sorts that are performed by the %BINARYSORT PeopleCode meta-SQL function take place in the character set of the database engine.

Therefore, you should be careful not to write code that depends on the sorting of data by using binary operators matching the sorting of data by the database as the result of sorting operators in SQL statements.

For example, the following PeopleCode statement is performed as a Unicode binary comparison, and *á* is located in the Unicode tables after the character *z,* so it always returns *True:*

```
if 'z' < 'á' then. . .
```

However, when the same comparison is run in SQL (as in the following example), the database's sort order determines which character is greater. In a database that is configured for binary sorting, it returns *true;* however, in a database that is configured for French sorting (where *á* is sorted after *a* but before *b*), it returns *false.*

```
SELECT . . .
WHERE 'z' < 'á'
```

It is important that these functional areas use binary sorting instead of attempting to perform a linguistically-sensitive sort because:

- Any linguistically-sensitive sort that is performed in memory can be only an approximation of the sort that the database system would perform in the same situation given the large number of sort orders that are provided by database vendors and the significant variations to these orders in minor versions of the database software.

  Do not assume that a binary sort that is performed in PeopleTools will match the sort of the same characters in a SQL statement that uses the less than, greater than, or BETWEEN operators.
- Performing a guaranteed database-compatible sort for each comparison in PeopleCode would require a round-trip to the database to perform the sorting and would affect the performance of PeopleCode operations.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Setting the Sort Order

As noted previously in These topics, some components of PeopleTools cannot rely on the database to sort data and must do so in memory. The sort order option on the PeopleTools Options page enables you to select which sort order should be used by PeopleTools when sorting data in memory.

Generally, you should set this option soon after you have completed the installation of the database; choose the option that most closely approximates the sort order that you selected when creating the database.

To set the sort order:

1. Select **PeopleTools**, and then **Utilities**, and then **Administration**, and then **PeopleTools Options.**
2. Select an option from the **Sort Order Option** drop-down list box.
3. Click **Save.**

Forcing a Binary Sort in SQL

When writing PeopleCode and other procedural logic, it is sometimes expected that sorting a list of data in memory produces the same results as sorting the same list in the database through a SQL ORDER BY statement. When working in some languages, such as English, whose sorting logic is relatively simple, this equivalence may be taken for granted.

However, when running PeopleTools against a database that is using a linguistic sort, it is likely that a greater than, less than, or between comparison of values in memory may produce different results than if the same comparison were performed by the database.

Take, for instance, the following PeopleCode syntax:

```
If START_NAME < END_NAME then
   Error("Start name must be less than end name");
End-If
```

Because all character comparisons in PeopleCode are performed based on the binary value of the character in Unicode (and not on the linguistic weight of the character), this code may produce unexpected results in languages where binary sorting is not sufficient. For example, if START_NAME had a value of *Über* and END_NAME had a value of *Zeifang,* this code produces the error as in the Unicode character set, the uppercase *U* with umlaut (*Ü*) appears after the uppercase *Z.* However, in a German sort, *Ü* should precede *Z.* If the database were created with a German sort order, this would be reflected by the database's sort if the same statement were reflected in SQL that is called from PeopleCode:

```
SQLExec("SELECT 'X' FROM PSLOCK WHERE :1 < :2"
        ,START_NAME, END_NAME, &X);
If &X != 'X' then
   Error("Start name must be less than end name");
End-If
```

The example shows that when you use a linguistically sorted database, these string comparisons may return different results when they are run through the database by using SQL than when they are performed in PeopleCode. Of course numerical values and dates

Copyright © 1988, 2026, Oracle and/or its affiliates.

always sort equivalently—this behavior is limited to the sorting of characters and character strings.

In some situations, you may want to disable the linguistic sort that is performed by the database on a statement-by-statement level and have the database perform the comparison at a binary level.

To force a SQL query to return sort-sensitive results based on a binary sort instead of a linguistic sort, a meta-SQL token, %BINARYSORT, is provided. Use this token is to wrap each column in an ORDER BY, less than, greater than, or BETWEEN operation where a binary comparison is required. For example, to return all employee names, ordered by last name in binary ordering, use the following SQL:

```
SELECT NAME FROM PS_PERSONAL_DATA
ORDER BY %BINARYSORT(NAME)
```

This may be useful if you are building an ordered array of names in memory that you plan to parse or manipulate with PeopleCode less than or greater than operators.

Similarly, to fetch a list of names from the database where the NAME field is greater than START_NAME by using a binary comparison that is parallel to that used in PeopleTools, use the following syntax:

```
SELECT NAME FROM PS_PERSONAL_DATA
WHERE %BINARYSORT(NAME) < %BINARYSORT(:1)
```

Similar constructs can be used with the BETWEEN predicate.

> **Note**
>
> The %BINARYSORT meta-SQL token ensures that the database evaluates the column that is wrapped by the token based on its binary value in the character set of the database. The sorting of this representation matches its binary sort position in PeopleTools only if the character set of the database contains the character in the same order as Unicode, which is used for binary representation of data in PeopleTools memory. Therefore, if you are running a US-ASCII, Latin-1 (ISO 8859-1), or Unicode database, the sorting of all alphabetic characters matches the Unicode sort in memory. However, if you are running a database that is encoded in EBCDIC or in Japanese Shift-JIS, the order may be markedly different, and you cannot rely on the %BINARYSORT meta-SQL token to match the binary order that is used within PeopleTools.

Related Topics

- PeopleCode Language Reference: %BINARYSORT meta-SQL element

Sorting in COBOL

Character string comparisons that are performed in COBOL are performed based on the binary representation of each character in the character set that is being used by the COBOL program. Typically, this matches the character set of the database. Use the %BINARYSORT meta-SQL token to wrap any ORDER BY clause or string comparison that is performed by the

Copyright © 1988, 2026, Oracle and/or its affiliates.

SQL statements that are called by the program that you want to sort the same as it would sort in COBOL memory.

Sorting in SQR Programs

Like PeopleCode, all string comparisons and sorting in SQR are performed based on a Unicode binary sort. However, because SQR does not process PeopleTools meta-SQL tokens, you cannot use %BINARYSORT in SQR programs. You should make allowances for SQR programs to not depend on having memory sorting match the sort that the database performs in ORDER BY and similar SQL clauses.

Performing Linguistic Sorting as a Customization

Some users may require linguistic sorting in limited contexts, such as French sorting for a particular table or report. As shipped, PeopleTools supports binary sorting only. You can implement linguistic sorting but you may need to do index tuning and testing, and recheck these after any upgrades.

For Oracle databases only, when selecting a linguistic sort, specify the sort order by setting the NLS_SORT and NLS_LANGUAGE environment variables, and update the Sorting Option in PeopleSoft Options.

For example, to enable French linguistic sorting, set the environment variables to the following:

```
NLS_SORT=FRENCH
NLS_LANGUAGE=FRENCH
```

> **Note**
>
> The linguistic sort must be case sensitive and accent sensitive.

On PeopleSoft Options Page, change Sorting value from *Binary* to *French*.

For certain queries, you may need to create a linguistic index to avoid performance penalties as a result of using the linguistic sort.

The following is an example for setting French Linguistic Index:

```
ALTER SESSION SET NLS_SORT='FRENCH';
CREATE INDEX test_idx ON test4(NLSSORT(name, 'NLS_SORT=FRENCH'));
SELECT * FROM test4 ORDER BY col;
ALTER SESSION SET NLS_COMP=LINGUISTIC;
SELECT * FROM test4 WHERE name > 'Henri';
```

> **Note**
>
> This step is optional and is for performance tuning.

Please consult your Oracle database documentation for more information on linguistic indexes.

Copyright © 1988, 2026, Oracle and/or its affiliates.

9

Using Global Reporting and Data Analysis Tools

Using 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 through you queried the base language record (COUNTRY_TBL) and not the related language record (COUNTRYTBL_LANG). PeopleSoft 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.

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, the output appears 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.

> **Note**
>
> 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. This is because the English description of *Allemagne* is *Germany.*

The automatic language join features in PeopleSoft Query are also used when you access your PeopleSoft database via the PeopleSoft Open Query ODBC interface.

Using Scheduled Queries

For scheduled queries, the system uses the language specified in the user's profile. It does not use the language selected during sign-in. The system will also use the international and regional settings the user has specified using the My Personalizations page. If no personal setting have been specified, the system uses the default installation international settings.

Copyright © 1988, 2026, Oracle and/or its affiliates.

> **Note**
>
> Most PeopleSoft components appear by default from international settings on the browser if the user has not set any user-specific settings. However, this is not available for scheduled queries or any PeopleSoft Process Scheduler processes.

Related Topics

- Query

Using the Strings Table for Language-Sensitive Text in Reports

The PeopleTools Strings table (STRINGS_TBL) stores text strings used for language-sensitive labels and other text for PS/nVision and SQR reports. This avoids hard-coding labels into the report files themselves. The use of strings rather than hard-coded text in reports enables translators to translate the report layout in the database without editing 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 and report logic.

The Strings table stores string text of three different string types:

- A short field description (RFT Short).
- A long field description (RFT Long). Only strings of this type can have translated values.
- A free-form text string (Text).

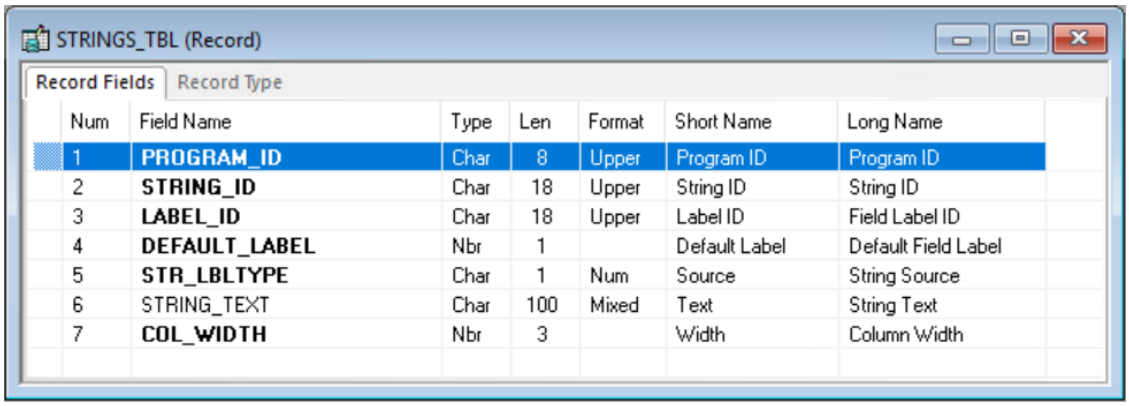
The Strings table is keyed by PROGRAM_ID plus the STRING_ID, which enables you to classify strings into groups that are used in similar reports. PROGRAM_ID can refer to a specific SQR or PS/nVision report name, or it can be a mnemonic for a group of common strings that are shared between reports. For RFT Short or RFT Long, the STRING_ID is the field name (validated against the PSDBFIELD or PSDBFIELDLANG table).

For example, PS/nVision string variables include the string ID and the program ID as follows:

```
%.StringID,ProgramID%
```

The following example shows the Use Display view for the STRINGS_TBL record definition in Application Designer with **PROGRAM_ID** and **STRING_ID** identified as the key fields:

This example illustrates the fields and controls on the STRINGS_TBL record definition.



Copyright © 1988, 2026, Oracle and/or its affiliates.

If the language of the requested report is a non-base language and only if the requested string is a long field description type, the system returns the translated string from the related-language table (STRINGS_LNG_TBL) if a translation is available. If no translation exists, the base language string is returned.

The return of language-sensitive string data follows this sequence as shown in the following diagram:

1. The system locates a string definition in the base language Strings table by using the program ID and string ID.

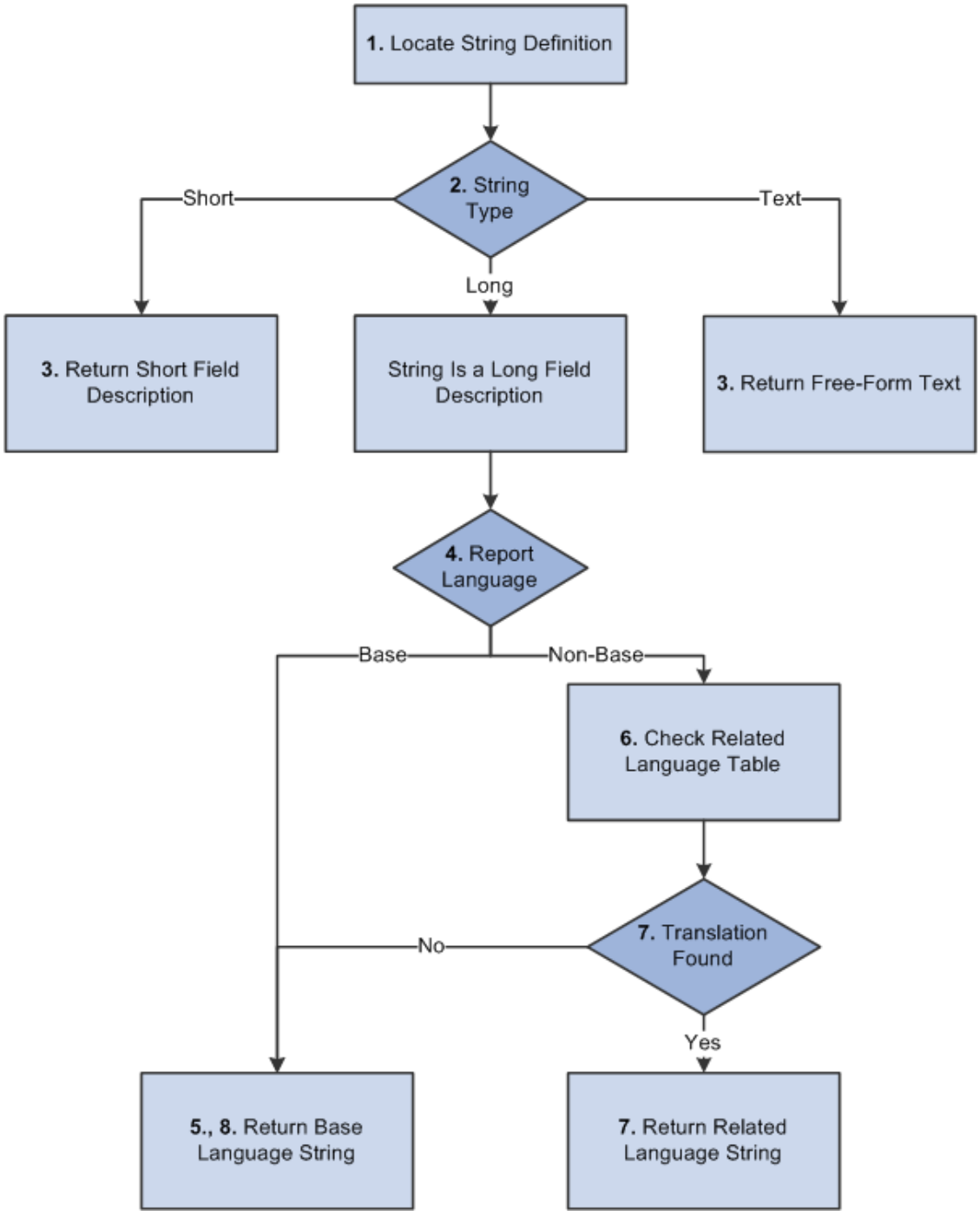
   The text string stored in the base language Strings table can be a short or long field description or a free-form text string that is not associated with a field.
2. The system determines the string type: short, long, or text.
3. If the string type is short or text, the system returns the string text directly from the base language Strings table.
4. The system determines the language of the report request.

> **Note**
>
> If the value is unspecified, the system uses the user's current language preference setting.

5. If the language is the system's base language, the system returns the string value from the base language Strings table.
6. If the language of the report is a non-base language, the system looks in the related-language table (STRINGS_LNG_TBL) for a row with the program ID, string ID, and correct language code.
7. If an appropriate row exists in the related-language table, the system returns the translation.
8. If no translation exists in the related-language table, the system returns the string from the base language table.

This diagram illustrates the return of language sensitive string data

Copyright © 1988, 2026, Oracle and/or its affiliates.



Related Topics

- Using SQR for PeopleSoft in Global Implementations
  SQR for PeopleSoft provides a range of features to allow not only for reports in multiple languages, but also for handling international date/time formatting, paper sizes, numeric formatting and much more. This section discusses some of the internationalization features of SQR for PeopleSoft.
- Strings
  The following table lists where to translate strings:

Copyright © 1988, 2026, Oracle and/or its affiliates.

Using SQR for PeopleSoft in Global Implementations

SQR for PeopleSoft provides a range of features to allow not only for reports in multiple languages, but also for handling international date/time formatting, paper sizes, numeric formatting and much more. This section discusses some of the internationalization features of SQR for PeopleSoft.

Related Topics

- SQR Language Reference for PeopleSoft: SQR Command Overview

Printing for A4 Paper

PeopleTools supports printing A4, legal, and U.S. letter sized paper. Changing the **PAPER_SIZE** setting in SETENV.SQC changes the paper size for all reports that share the same file/report/batch server directory.

To print Legal paper, uncomment the following line in the SETENV.SQC file:

```
#define PAPER_SIZE LEGAL
```

To print A4 paper, uncomment the previous line and edit it to be:

```
#define PAPER_SIZE A4
```

To ensure that you are printing the desired paper size, you must comment out or delete any other #define PAPER_SIZE statements.

Currency Precision

NUMBER.SQC enables you to make use of currency precision, both in terms of character string values (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
```

See the comments block at the beginning of the NUMBER.SQC SQR include file for documentation of these functions.

Date and Time Formatting

The DATETIME.SQC program provides several procedures to aid in the 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.

Copyright © 1988, 2026, Oracle and/or its affiliates.

During the 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 the 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. This table identifies these formats:

SQR Date Format	Description
{DEFDMY}	DD/MM/YYYY
{DEFMDY}	MM/DD/YYYY
{DEFKAN}	Japanese Kanji-format dates (using the Japanese Imperial calendar).
{DEFROM}	Japanese Romaji-format dates (using the Japanese Imperial calendar).

The following table includes examples for SQR for PeopleSoft date formats for the date December 14, 2000:

Field or Control	Description
14/12/2000	{DEFDMY}
12/14/2000	{DEFMDY}
+	{DEFKAN}
H.12.12.14	{DEFROM}

For details about 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 the SQR report, remember to include *#define JapaneseDates* at the top of the SQR report.

Report Translation

PeopleTools enables you to print or format SQR 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.

To enable the use of the Strings table in your SQR program, you must 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 you can call from the report:

- Init_Report_Translation
- Get_Field_Information
- Append_Report_Translation

Copyright © 1988, 2026, Oracle and/or its affiliates.

- Add_Report_Translation

Init_Report_Translation

Call the Init_Report_Translation procedure from your SQR program before using any of the String table information. Typically, you should call Init_Report_Translation in the Init-Report section of your SQR program. Init_Report_Translation takes two parameters:

Parameter Name	Description
$Report_ID	$Report_ID is normally the name of the SQR report. 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 attempts to load all strings in the language specified; however, if a translation for any string does not exist, it loads 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 program, 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
$Report_ID	$Report_ID is normally the name of your SQR report. This parameter is used as the program ID when looking up strings in the Strings table. You must have already called Init_Report_Translation specifying this $Report_ID before passing it to Get_Field_Information.
$Field_ID	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 $Report_ID that you specified.
$Field_Text	$Field_Text is the output variable. Get_Field_Information populates this variable with the text that corresponds to the $Report_ID and $Field_ID that are specified in the preferred language or in the database's base language (if a translation doesn't exist in the preferred language).
$Field_Width	$Field_Width is an output variable that Get_Field_Information populates with the width of the text string that is returned.

Copyright © 1988, 2026, Oracle and/or its affiliates.

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 is used across many of your SQR programs. You can group these strings under a generic program ID and use them in multiple SQR programs.

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 program, this function calls 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 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 is initialized or appended correctly.

Sample Strings Table Enabled SQR Program

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', $EXPTEXT,       #DWET)
PRINT $REPORTTITLE       (1)    CENTER
PRINT $EXPTEXT           (+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 || ':'
```


Copyright © 1988, 2026, Oracle and/or its affiliates.

```
   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
```

Related Topics

- Using the Strings Table for Language-Sensitive Text in Reports
  The PeopleTools Strings table (STRINGS_TBL) stores text strings used for language-sensitive labels and other text for PS/nVision and SQR reports. This avoids hard-coding labels into the report files themselves. The use of strings rather than hard-coded text in reports enables translators to translate the report layout in the database without editing 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 and report logic.

The PSSQR.INI and PSSQR.UNX Files

Most of the parameters that affect globalization of SQR for PeopleSoft are set in the configuration file. The PeopleSoft system delivers default configuration files, which are located in the *PS_HOME*\sqr directory. The files are named pssqr.ini on Microsoft Windows, and pssqr.unx on Unix/Linux. Because different configurations are needed for some of the supported languages, PeopleSoft delivers eight configuration files on each platform. These language-specific configuration files are named pssqr*language_code*.ini or pssqr*language_code*.unx. When the SQR process is started from PeopleSoft Process Scheduler, one of these configuration files is selected based on the user's language preference set in the My Personalization page. When language-specific files are not found, PeopleSoft Process Scheduler selects pssqr.ini/unx without *language_code.*

By default, these configuration files are assigned a character set and font that are correct for each language. However, it is also possible to customize the files to better suit your integration needs and report requirements. The following sections describe the details of configurations you can specify in the pssqr.ini/unx files.

> **Note**
>
> On DB2 UDB for OS/390 and z/OS initialization files, PSSQR*LANGUAGECD*INI and PSSQRINI are provided as members of the SQRSRC dataset on TSO. PeopleSoft Process Scheduler selects a language-specific initialization file if it exists in the SQRSRC dataset by replacing the %SQRINI% meta-string on shell JCT at process runtime. Because of the limitation in PeopleSoft Process Scheduler running on z/OS, it is not possible to use a different character set on language-specific INI files on z/OS. The language-specific configuration files are provided to make different font configurations for some languages. If you need to process multiple character sets for a single database using PeopleSoft Process Scheduler on z/OS, you must set up multiple PeopleSoft Process Schedulers.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Related Topics

- SQR Configuration for Processing International Text
  SQR for PeopleSoft uses Unicode for internal storage of character data regardless of whether you are running against a Unicode encoded PeopleSoft database. This enables a single instance of SQR to process data in virtually any language.
- SQR Configuration for Printing International Text
  SQR for PeopleSoft is a Unicode application. It can generate reports that contain multiple languages, such as English, French, Japanese, and Thai, on a single report and even on a single page, if you use advanced reporting output types like PDF, SPF, HTML, or Enhanced HTML. However, for other output types that SQR supports, printing or generating reports in languages other than the Western European languages might require special considerations. This section provides some hints for customizing or resolving international text printing issues with SQR for PeopleSoft.

SQR Configuration for Processing International Text

SQR for PeopleSoft uses Unicode for internal storage of character data regardless of whether you are running against a Unicode encoded PeopleSoft database. This enables a single instance of SQR to process data in virtually any language.

Although SQR runs using Unicode internally, it automatically converts data read from and written to files to the appropriate non-Unicode encoding based on configurations made in the pssqr.ini/unx file. You can modify the delivered configuration to use different non-Unicode encoding based on your integration and reporting needs. Further, you can use Unicode as the input and output character set if your printer or external system communicating with SQR by file can process Unicode data.

The PeopleSoft system delivers pssqr.ini/unx files with configurations for each supported language and database connection. Manual configuration is not required for basic processing of supported languages or to connect to a Unicode database. The information in this section will help you customize SQR for PeopleSoft to better suit your integration and reporting requirements.

See The PSSQR.INI and PSSQR.UNX Files.

Advanced PSSQR.INI/UNX Settings

The following parameters under the [Environment: Common] or [Environment: <Database type>] sections of the pssqr.ini/unx file control the character set SQR uses for specific operations. The ENCODING parameter sets the default encoding for all types of operations. You can, however, control encodings for each operation independently by adding the parameters in the [Environment: Common] section of the file.

```
ENCODING
ENCODING-SQR-SOURCE
ENCODING-FILE-OUTPUT
ENCODING-FILE-INPUT
ENCODING-REPORT-OUTPUT
ENCODING-CONSOLE
```

If an individual encoding parameter is explicitly included, this specific setting overrides the default encoding specified by the ENCODING parameter.

- The ENCODING-SQR-SOURCE parameter specifies the encoding in which the SQR source (*.SQR, *.SQC) files are encoded.

Copyright © 1988, 2026, Oracle and/or its affiliates.

- 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. If SQR detects a byte-order mark (BOM) at the top of file, it always reads as a Unicode file, regardless of this setting.

- The ENCODING-REPORT-OUTPUT parameter determines the encoding used for report output types LP, HP, PS, and CSV.

  The parameter should be set to a character set that contains all the characters you expect to print using SQR and, if you intend to print the output on a printer, it should also be set to the character set that the printer supports. Output in SPF, HTML, Enhanced HTML, and PDF format is not controlled by this parameter. SPF HTML, and Enhanced HTML always use Unicode for their output, and PDF output is controlled exclusively by the font configuration.

- The ENCODING-CONSOLE parameter determines the character set that is used to write progress and other messages to the console during an SQR run.

  In Microsoft 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 Microsoft Windows. See Microsoft's web site, OEM Code Pages, for a list of Microsoft Windows OEM code pages. For Unix/Linux systems, this setting should match the character set that is supported by your terminal device.

- If the ENCODING parameter is not set, SQR determines the encoding to use based on the machine locale.

- If the same parameter is not set in the [Environment: Common] and [Environment: <Database Type>] sections of the pssqr.ini/unx file, the configuration in the [Environment: <Database Type>] section overrides the configuration in the [Environment: Common] section.

Related Topics

- The PSSQR.INI and PSSQR.UNX Files
  Most of the parameters that affect globalization of SQR for PeopleSoft are set in the configuration file. The PeopleSoft system delivers default configuration files, which are located in the *PS_HOME*\sqr directory. The files are named pssqr.ini on Microsoft Windows, and pssqr.unx on Unix/Linux. Because different configurations are needed for some of the supported languages, PeopleSoft delivers eight configuration files on each platform. These language-specific configuration files are named pssqr*language_code*.ini or pssqr*language_code*.unx. When the SQR process is started from PeopleSoft Process Scheduler, one of these configuration files is selected based on the user's language preference set in the My Personalization page. When language-specific files are not found, PeopleSoft Process Scheduler selects pssqr.ini/unx without *language_code.*

SQR Configuration for Printing International Text

SQR for PeopleSoft is a Unicode application. It can generate reports that contain multiple languages, such as English, French, Japanese, and Thai, on a single report and even on a single page, if you use advanced reporting output types like PDF, SPF, HTML, or Enhanced HTML. However, for other output types that SQR supports, printing or generating reports in languages other than the Western European languages might require special considerations. This section provides some hints for customizing or resolving international text printing issues with SQR for PeopleSoft.

This section discusses how to:

- Set up encoding parameters.

Copyright © 1988, 2026, Oracle and/or its affiliates.

- Set up fonts.
- Set up PDF fonts.
- Set up files for PCL and line printing.
- Set up files for PostScript printing.

This section also discusses limitations of SQR for PeopleSoft to print international text.

Setting Up Fonts

SQR for PeopleSoft represents fonts internally as numbers, and the pssqr.ini/unx file controls the mapping between the font numbers for some printer types. The [Fonts] section in the pssqr.ini/unx file controls the font mappings of the following output types:

- SPF Viewer
- WP (Microsoft Windows Printer)
- Enhanced HTML

By default, this section is not configured with language-specific fonts. This is because Microsoft Windows and HTML browsers have a font linking mechanism that allows them to select an appropriate language-specific font automatically when the font specified does not have characters to show. However, Microsoft Windows and the browser's font linking works differently based on versions, availability of fonts on the system, and the configuration of Microsoft Windows and the browser. If you experience problems showing characters for these output types, you should try the configuration described in this section

To set up fonts:

1. Open the appropriate pssqr.ini/unx file for your language, located in *PS_HOME*\sqr.
2. In the [Fonts] section, the following default values appear.

   Use the chart to replace each of the entries for your target language.

```
3=Courier New,fixed
300=Courier New,fixed,bold
4=Arial,proportional
400=Arial,proportional,bold
5=Times New Roman,proportional
500=Times New Roman,proportional,bold
6=AvantGarde,proportional
8=Palatino,proportional
800=Palatino,proportional,bold
11=Symbol,symbol
900=unknown,proportional
901=Times New Roman,proportional
28825=MS UI Gothic,proportional
```

Font Number	Japanese	Simplified Chinese	Traditional Chinese	Thai	Korean
3	MS Gothic, proportional	SimHei, proportional	MingLiu, proportional	CordiaUPC, proportional	GulimChe, proportional
300	MS Gothic, proportional, bold	SimHei, proportional, bold	MingLiu, proportional, bold	CordiaUPC, proportional, bold	GulimChe, proportional, bold

Copyright © 1988, 2026, Oracle and/or its affiliates.

Font Number	Japanese	Simplified Chinese	Traditional Chinese	Thai	Korean
4	MS Gothic, proportional	SimHei, proportional	MingLiu, proportional	CordiaUPC, proportional	GulimChe, proportional
400	MS Gothic, proportional, bold	SimHei, proportional, bold	MingLiu, proportional, bold	CordiaUPC, proportional, bold	GulimChe, proportional, bold
5	MS Mincho, proportional	SimSun, proportional	MingLiu, proportional	AngsanaUPC, proportional	BatangChe, proportional
500	MS Mincho, proportional, bold	SimSun, proportional, bold	MingLiu, proportional, bold	AngsanaUPC, proportional, bold	BatangChe, proportional, bold
6	MS Gothic, proportional	SimHei, proportional	MingLiu, proportional	CordiaUPC, proportional	GulimChe, proportional
8	MS Gothic, proportional	SimHei, proportional	MingLiu, proportional	CordiaUPC, proportional	GulimChe, proportional
800	MS Gothic, proportional, bold	SimHei, proportional, bold	MingLiu, proportional, bold	CordiaUPC, proportional, bold	GulimChe, proportional, bold
901	MS Mincho, proportional	SimSun, proportional	MingLiu, proportional	CordiaUPC, proportional	GulimChe, proportional
28825	MS UI Gothic, proportional	N/A	N/A	N/A	N/A

Setting Up PDF Fonts

For PDF output, SQR for PeopleSoft has an internal font linking mechanism by which you can link up to 10 fonts for a single font number. This allows you to get character coverage that is not possible by using single font. With this feature, you can print reports containing multiple languages without rewriting your program to show each language in a different font. PeopleSoft delivers the pssqr.ini/unx file with world-ready configuration; therefore, you usually do not need to modify the pssqr.ini/unx file to print non-Western European languages. Many major business languages can be printed in PDF independent of the user's language settings on the My Personalization page. For example, you can output Greek data in PDF even when your My Personalizations language is set to English.

Although it is possible to print Chinese, Japanese, or Korean text in a PDF report without modifying the pssqr.ini/unx configuration, printing Chinese, Japanese, and Korean text in a single report requires special configuration. Font linking data is configured in the pssqr.ini/unx file and is transparent to the user. Depending on your reporting requirements, you can add or remove fonts from the delivered configuration, and you can control which actual font to use down to a single character level. This section explains how to add fonts for PDF output.

The [PDF Fonts] section in the pssqr.ini/unx file supports multiple fonts mapped to a single font number.

The [PDF Fonts: Exclusion Ranges] section of the file specifies exclusion ranges for fonts listed in the [PDF Fonts] section. If an exclusion range is set, then when a character is covered by a font, that font is not used to print the character. Exclusion ranges are set in Unicode format, in hexadecimal or decimal. The base configuration does not contain an exclusion range.

The [TrueType Fonts] section of the file specifies the mapping from TrueType font names used in the [PDF Fonts] configuration section, along with the physical file path of the font on the

Copyright © 1988, 2026, Oracle and/or its affiliates.

operating system. For TrueType collection (.ttc) files, the font directory number should also be specified (in the format *font name=file path, directory number*.

The font path is the directory where the font resides. The default is *SQRDIR*. On Microsoft Windows, the font directory is looked up as well. Fonts residing in other directories must be specified by their full physical pathnames.

The [PDF Fonts] information in the pssqr.ini/unx file is language-specific for Chinese, Japanese, and Korean languages; the information is the same for other supported languages due to Unicode characteristics. If you want to add another language (for example, Russian), you should refer to the documentation for adding new languages.

See the *PeopleSoft 9.2 Application Installation* for your database platform.

See Understanding the Addition of New Language Definitions.

See SQR Language Reference for PeopleSoft.

Sample Steps for Adding a Font for PDF Output: User Defined Font for Japanese

For computing in the Japanese language, it is common to use user-defined characters. A user-defined character is a character that is not encoded in standard character sets like JIS X 0208 or Unicode, and sometimes variants of glyphs existing in standard character sets. This kind of character is often used for a person's name. Unicode has space allocated to encode such user-defined characters, and users can create fonts that include characters not defined in Japanese fonts.

On Microsoft Windows, users can create user-defined fonts using a program called EUDCEDIT to design and define private characters. This program creates a TrueType font file named "eudc.tte" under the Microsoft Windows font folder. As an example of adding fonts for PDF output from SQR for PeopleSoft, perform the following steps to add the "eudc.tte" font to the PeopleSoft-delivered configuration file for Japanese, pssqrjpn.ini:

1. Open the pssqrjpn.ini file from *PS_HOME*\sqr and add the following line under [TrueType Fonts] section:

   ```
   [TrueType Fonts]
   Font Path=...
   Cumberland=...
   GaijiFont=eudc.tte
   ```

   As explained in this section, you do not need to specify a full path for the font residing in the Microsoft Windows font folder, and you can specify a name different from the actual name of the font ("GaijiFont" means a user defined font in Japanese. The actual font name that "eudc.tte" itself has is "EUDC").

2. Under the [PDF Fonts] section, add "GaijiFont" at the end of the font list.

   For example, for font number 3:

   ```
   3=Courier,HeiseiKakuGo-W5,Cumberland,MHei-Medium,STSong-Light,HYGoThic-
   Medium,
   Angsana,GaijiFont
   ```

3. Define exclusion ranges in the [PDF Fonts: Exclusion Ranges] section, to exclude higher priority fonts to show characters in the Unicode private use range where characters in eudc.tte are defined.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Chinese fonts have glyphs defined in the Unicode private use range mainly to include characters defined in national character standards like GBK but not defined in earlier versions of Unicode. Thai fonts also have Thai presentation form characters mapped in the Unicode private use range. As we are working on a Japanese configuration and Chinese characters and Thai presentation forms are rarely needed in a Japanese context, we will configure these fonts not to print characters in the Unicode private range, and private range characters are always printed using "GaijiFont," which we just added. The Unicode private use range is from 0xE000 to 0xF8FF. Chinese fonts are third and fourth in the font link list, and the Thai font is seventh. Therefore, the exclusion definition looks like the following (exclusion range setting for second font comes from original configuration of pssqrjpn.ini, that excludes extended Latin characters from Japanese font):

```
3=,0x0000-0x02FF,,0xE000-0xF8FF,0xE000-0xF800,0xE000-0xF800,,
```

Considerations for Printing Chinese, Japanese, and Korean Text in a Single PDF Output

Unicode application printing of Chinese, Japanese, and Korean text in a single context requires special consideration in the use of fonts, because Unicode encodes ideographs used in Chinese, Japanese, and Korean languages in unified codepoints. A Unicode character is unique in meaning, but the glyph (the shape of the character) might be different depending on the language context. A commonly used technique in resolving this glyph difference is to use a language-specific font for Chinese, Japanese, and Korean, based on external context.

The PeopleSoft system delivers different font linking configurations for Chinese, Japanese, and Korean languages in language-specific pssqr.ini/unx files. In the Simplified Chinese configuration, the Simplified Chinese font is given higher priority than other Chinese, Japanese, or Korean fonts, so that a Simplified Chinese glyph is used for most Chinese ideographs. At runtime, PeopleSoft Process Scheduler picks up language-specific pssqr.ini/unx files with appropriate font linking data for the language, based on the user's My Personalization language setting. If you set your language as Traditional Chinese, the configuration file that uses the Traditional Chinese font for the majority of ideographic characters will be picked up. It is recommended that you set up the language in the My Personalization page when you are printing reports that include Chinese, Japanese, and Korean data. If you do not use any Chinese, Japanese, and Korean languages for the My Personalization page, the default configuration is used and in this case the printed glyph image might be different from what you expect to see.

If you have a reporting requirement to print a mix of Chinese and Japanese in a single report and you need to show each language in the correct glyph image for that language, you will need to use the ALTER-PRINTER command to programmatically change the font in your SQR program based on external data, such as the Language Code of the data. For this purpose, pssqr.ini/unx files are provided with the following font numbers, each of which uses fonts specific for each Chinese, Japanese, and Korean locale.

- 28825, 28752 (Japanese).
- 37110, 37058 (Simplified Chinese).
- 33269 (Traditional Chinese).

Setting Up Files for PCL and Line Printing

Most of the configurations for printing international text from SQR for PeopleSoft are performed by setting options in the pssqr.ini/unx file. However, for some of the printer-specific settings, you must set parameters in SQR for PeopleSoft programs because of the SQR for PeopleSoft syntax.

Copyright © 1988, 2026, Oracle and/or its affiliates.

For example, SQR for PeopleSoft enables users to control printer parameters for PCL format in DEFINE-PRINTER or ALTER-PRINTER commands. PeopleTools provides default setup values for PCL format in three SQC files (ptset01.sqc, ptset02.sqc and ptset02a.sqc). If you intend to print languages in PCL format other than those covered by Latin1, you must add printer setup parameters to the SQC files.

Before you perform the configuration tasks described in this section, make sure that you have properly configured your printers to print in the target language.

If your printer does not have the fonts for the languages you want to print, you might need to purchase fonts in ROM from your printer vendor and install them on your printer.

The parameters you set for PCL printing must be consistent with the ENCODING parameters set in the pssqr.ini/unx file. You must set up the ENCODING parameters for each language before you set the parameters for PCL printing.

To set up PCL printing parameters for Japanese:

1. Open setenv.sqc file under *PS_HOME*\sqr.
2. Uncomment the following parameter:

```
#define PRINT_JAPANESE
```

3. Save and close the file.

To set up PCL printing parameters for languages other than Japanese:

> **Note**
>
> If you intend to print a language other than those covered by Latin1 and if it is not Japanese, follow these steps to set up the SQC files. The values presented here are examples and can vary depending on your printer. You should check your printer manual for the appropriate fonts and symbol sets.

1. Open the ptset01.sqc file located under *PS_HOME*\sqr.
2. Between the declare printer DEFAULT-HP and end-declare statements, add the following information with the appropriate values:

```
font
symbol-set
```

The values you set differ from language to language. The following table shows sample values for several languages commonly used by PeopleSoft customers.

Language	Font	Symbol Set
Central/Eastern European	No need to set.	2N
Traditional Chinese	33269	18T
Simplified Chinese	37058	18C

Copyright © 1988, 2026, Oracle and/or its affiliates.

For example, if you are setting ptset01.sqc for Simplified Chinese printing, try the following setting:

```
declare-printer DEFAULT-HP
#ifdef PRINT_JAPANESE
        init-string=<27>&t31P
        font=28825
        symbol-set=19K
#endif
        font=37058symbol-set=18C
        point-size=7.2
        pitch=17
end-declare
```

3. Under the declare printer DEFAULT-LP statement, locate the init-string parameter.

   Modify the init-string parameter based on the language to print. The following example shows sample values for HP LaserJet printer's line printer mode. Consult your printer manual for correct values for this parameter.

Language	init-string
Eastern European	<27>E<27>(2N<27>&l0O<27>&l6C<27>&l0E<27>&l95F<27>(s16.66H<27>&k2G
Traditional Chinese	<27>E<27>(18T<27>&l0O<27>&16C<27>&10E<27>&195F<27>(s16.66H<27>&k2G<27>(s1p7.25v0s0b33269T
Simplified Chinese	<27>E<27>(18C<27>&l0O<27>&16C<27>&10E<27>&195F<27>(s16.66H<27>&k2G<27>(s1p7.25v0s0b37058T

4. Save and close the file.
5. Open ptset02.sqc from *PS_HOME*\sqr and repeat steps 2 through 4.
6. Open ptset02a.sqc *PS_HOME*\sqr and repeat steps 2 through 4.

Setting up Files for PostScript Output

In addition to PCL and line printer format, SQR for PeopleSoft supports output in PostScript format, which can be printed with a printer that has PostScript interpreter. However, you configure fonts in a setup file provided for PostScript printing.

The parameters you set for PostScript printing must to be consistent with the ENCODING parameters you set in the pssqr.ini/unx file.

By default, the setup file provided for PostScript printing is configured for printing Western European languages. Depending on the encoding parameter in the pssqr.ini/unx file, you must either "re-encode" fonts, or use different fonts that support specific languages other than those set in the configuration file by default (you may need to use printer pre-installed fonts that support specific languages, or install external fonts on your printers). Consult your printer manual for the character sets your printer supports.

To set up files for PostScript output:

1. Open the postscri.str file from *PS_HOME*\bin\sqr*db_platform*\binw or from the directory where the SQR executable resides.
2. If you are using select character sets, such as Latin9, it is possible to print the entire repertoire of the character set just by re-encoding PostScript fonts.

Copyright © 1988, 2026, Oracle and/or its affiliates.

The PeopleSoft system delivers PostScript code and re-encoding data for Microsoft Windows CP1252 and CP1250 on Microsoft Windows, and Latin1, Latin9, and Latin2 on Unix/Linux. The default is CP1252 on Microsoft Windows and Latin1 on Unix/Linux. If you are using other character sets, the PeopleSoft system provides re-encoding data and you will need to modify part of the setup file. For example, the following example is for using the Latin9 character set on Unix/Linux platforms:

Locate the following section:

```
% ISO Latin1
/CODE []
160 /space
...
```

Comment out the “CODE” section entirely, by adding a percent sign (%) at the beginning of each line.

Locate the following section:

```
% ISO Latin9
% (enable this array when you set ENCODING/ENCODING-REPORT-
OUTPUT=ISO-8859-15
%  and comment out above array for ISO Latin1)
% /CODE [
% 160 /space
…
```

Remove the percent signs in front of each line for the entire “CODE” section.

> **Note**
>
> If you plan to use CP1250 on Microsoft Windows or Latin2 on Unix/Linux, you will need to ensure that fonts installed on printer have the characters needed for these character sets. Depending on the printer and the fonts installed, you may not be able to print the entire repertoire of these Eastern European character sets using provided re-encoding data. Note also that if you plan to use character sets other than those discussed previously in this section, you will need to ensure that the printer has fonts supporting the specific character set you want to use. You may need to install additional fonts, and may also need to re-encode the fonts by adding your own font re-encoding data.

3. Some other character sets, including character sets commonly used for Japanese, Korean, and Chinese, can be printed by using PostScript fonts supporting the specific language. You can replace the font names in the setup file in the following way:

   Locate the following section:

   ```
   /Fonts [   % Array of point sizes and font names
   ```


Copyright © 1988, 2026, Oracle and/or its affiliates.

Replace each font with the name of the appropriate font installed on your printer that can print the target language. For CJK languages (Chinese, Japanese, and Korean), PostScript printers accept the font names in the following format:

```
(Font Name)-(CMap name)
```

Font Name represents the name of the font installed on your printer that supports the language that you want to print. *CMap* is a file the PostScript printer used to find the glyph that corresponds to the codepoint passed to the printer. You need to use the *CMap* that matches the encoding specified in the ENCODING parameter in the pssqr.ini/unx file.

The following example shows setup values for each of the CJK languages. Supported font names and CMap files may vary depending on your printer. For details, refer to your printer documentation.

Language	Report Output Encoding	Serif Typeface	San Serif Typeface
Japanese	SJIS	Ryumin-Light-RKSJ-H	GothicBBB-Medium-RKSJ-H
Traditional Chinese	Big5	MSung-Light-B5-H	MHei-Medium-B5-H
Simplified Chinese	CP936	STSong-Light-GBK-EUC-H	STHeiti-Regular-GBK-EUC-H
Korean	CP949	HYSMyoeongJo-Medium-KSCms-UHC-H	HYGoThic-Medium-KSCms-UHC-H

If you do not need to re-encode the fonts you have set, you must comment out the following section entirely by adding a percent sign (%) in front of each line. (For CJK character sets, you do not need to re-encode fonts; however, you might need to do so for other languages.)

```
%
% Re-encode all the fonts except for symbol/dingbats
% (do not re-encode CJK fonts - if you intend to use CJK fonts comment
%  out the below section entirely)
%
/Courier /Courier CODE ReEncode
/Courier-Bold /Courier-Bold CODE ReEncode
/Helvetica /Helvetica CODE ReEncode
/Helvetica-Bold /Helvetica-Bold CODE ReEncode
…
```

4. Save and close the file.

Limitations in Printing International Text from SQR for PeopleSoft

The following limitations apply when printing international text from SQR for PeopleSoft:

- SQR for PeopleSoft does not support languages that require bi-directional text rendering, such as Arabic and Hebrew.
- The default configuration of report formats other than PDF, SPF, WP, HTML, and Enhanced HTML is to use non-Unicode character sets (ISO or commonly used national standard character sets on Unix/Linux, and the Microsoft Windows character set on Microsoft Windows).

Copyright © 1988, 2026, Oracle and/or its affiliates.

The ability to print language data for LP, HP, PS, CSV, and HT output types is limited by the character set used for output. For example, you can print Western European languages like English, French, Spanish, and Portuguese using the Latin1 character set (which is used in pssqr.unx), but you cannot add Polish data using Latin1 because the Polish language uses characters not encoded in the Latin1 character set. You can configure the pssqr.ini/unx files to generate output types other than HT in Unicode-based character sets like UCS-2 or UTF-8, but in these cases you need to ensure that your printer or viewer application supports Unicode natively.

- If you intend to print output using the LP, HP, or PS format, SQR for PeopleSoft's ability to print international characters is also limited by the fonts installed on the printers. For example, you can print Japanese on a PostScript printer if a Japanese font like Ryumin-Light is installed on the printer. However, if your printer does not natively provide font support, the printer output with international text may not be printed as intended. Unlike with PDF output, SQR for PeopleSoft currently does not support downloading host-based fonts, including True Type fonts, to the printer as part of the print job. Printing from Adobe Reader or SPF Viewer is not affected by this limitation.

International Text in SQR for PeopleSoft Programs

This section discusses:

- System variables to check encoding settings.
- String length.
- OPEN command.

System Variables to Check Encoding Settings

You can reference the values set for the ENCODING parameters in pssqr.ini/unx from your SQR for PeopleSoft programs using the following system variables:

- *$sqr-encoding*

  Stores value set to the ENCODING parameter of pssqr.ini/unx. If no value is set for this parameter, SQR automatically determines the appropriate encoding for input and output based on the operating system locale and stores values in this variable.

- *$sqr-encoding-database-api*

  Stores value set to ENCODING-DATABASE-API parameter of pssqr.ini/unx file. Except for DB2 UDB for Linux, Unix, and Microsoft Windows, the value for this parameter is automatically determined by the system and cannot be overridden by editing pssqr.ini/unx. In this case, this variable stores the encoding value automatically determined and used by system. For example, on an Oracle platform, this variable value is always *UTF-8.*

- *$sqr-encoding-file-input*

  Stores value set to ENCODING-FILE-INPUT parameter in pssqr.ini/unx file. If no value is set for this parameter, the input files are considered to be encoded in the encoding specified in the ENCODING parameter and this variable stores the same value as $sqr-encoding.

- *$sqr-encoding-file-output*

  Stores value set to ENCODING-FILE-OUTPUT parameter in pssqr.ini/unx file. If no value is set for this parameter, the same encoding set in the ENCODING parameter is used for file output encoding and this variable stores the same value as $sqr-encoding.

- *$sqr-encoding-report-output*

Copyright © 1988, 2026, Oracle and/or its affiliates.

Stores value set to ENCODING-REPORT-OUTPUT parameter in pssqr.ini/unx file. If no value is set for this parameter, the same encoding set in the ENCODING parameter is used for report output encoding and this variable stores the same value as $sqr-encoding.

- *$sqr-encoding-source*

  Stores value set to ENCODING-SQR-SOURCE parameter in pssqr.ini/unx file. If no value is set for this parameter, SQR source files are considered to be encoded in the character set encodings specified in the ENCODING parameter and this variable stores the same value as $sqr-encoding.

- *$sqr-encoding-console*

  Stores value set to ENCODING-CONSOLE parameter in pssqr.ini/unx file. If no value is set for this parameter, the same encoding set in the ENCODING parameter is used for console output encoding and this variable stores the same value as $sqr-encoding.

> **Note**
>
> The system variables are read-only and cannot be overwritten at runtime. They can be used with functions and commands that take encodings as parameters, such as substrt(), lengtht(), and open.

String Length

When you work with strings in SQR, you must 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 two bytes; however, your database may use a different encoding that requires a different number of bytes for each character.

SQR provides the following functions to help you manage string length according to each of these criteria.

> **Note**
>
> Although SQR uses Unicode internally, it can still read/write with non-Unicode (ANSI) databases and files. Refer to the OPEN command in the SQR documentation for details.

- Length() and Substr().

  Length() and Substr() deal with the number of characters in a string. As the PeopleSoft system field lengths (as defined in 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.

Copyright © 1988, 2026, Oracle and/or its affiliates.

For example, the following code determines whether string *&abc* will fit into a database field that is 10 characters long. If the string won't 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 won't fit into the print area, the code truncates the string to use only the first 10 columns 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 *$sqr-encoding-file-output* is used to reference the *SQR.INI ENCODING-FILE-OUTPUT* variable, which determines the default character set of any file that is written to by the SQR OPEN command. You can substitute any valid PeopleSoft system encoding for the *$sqr-encoding-file-output* variable.

```
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 programs, 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, the SQR program uses the character set specified in the pssqr.ini/unx parameter ENCODING-FILE-OUTPUT or ENCODING-FILE-INPUT, depending on whether you are opening the file for reading or writing.

To integrate with a third-party system, specify the character set for SQR programs to match the target 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 pssqr.ini/unx settings and enables the SQR program 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).

As another example, if you are generating text output for a mail merge in Microsoft Word, you can specify UCS-2 or Unicode-Little-Endian to the ENCODING parameter of the OPEN statement, so that Microsoft Word can import the Unicode file including international text.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Related Topics

- Understanding Character Sets
  This section discusses character sets.
- SQR Language Reference for PeopleSoft: OPEN

SQR for PeopleSoft-Supported Character Set Encodings

This section lists the character set encodings that are supported by SQR for this PeopleTools release. The character set encodings are organized by the following character sets:

- Arabic
- Baltic
- Celtic
- Chinese (Simplified)
- Chinese (Traditional)
- Cyrillic
- Gurmukhi
- Greek
- Hebrew
- Icelandic
- Japanese
- Korean
- Latin
- Latin (Canadian French)
- Latin (Central European)
- Latin (Southeastern European)
- Malayalam
- Nordic
- Slavic
- Symbol
- Thai
- Turkish
- Unicode
- Vietnamese

Explanation of the Following Tables

The Encoding Parameter column lists the values that you can specify in the ENCODING parameter in the pssqr.ini/unx file.

ENCODING-REPORT-OUTPUT applies to all output types.

In the Output Supported column of the following tables, note the following:

Copyright © 1988, 2026, Oracle and/or its affiliates.

Field or Control	Description
PCL	*PCL* denotes PCL printing format. Values in parentheses next to PCL, for example *PCL (8V)*, are symbol sets. If PCL is not listed for an encoding parameter, the encoding is not supported output. Additional hardware support (font ROM) may be required to get correct output.
PS	*PS* denotes a PostScript printer.
CSV	CSV denotes comma-separated value format. *CSV (*)* means output is supported but Microsoft Excel may not read the output correctly. To read the output in Microsoft Excel, use a supported encoding other than those indicated with an asterisk (*).
Flat file	*Flat file* denotes that the output is generated by the WRITE command or that the text encoding is readable with the READ command.

SPF, HTML, Enhanced HTML, and PDF can output all the supported languages and encodings. For SPF, HTML, and Enhanced HTML, SQR uses UTF-8 encoding (Unicode), even if ENCODING and ENCODING-REPORT-OUTPUT is set to non-Unicode encoding in the pssqr.ini/unx file. For PDF, character support is dependent on the fonts listed in the [PDF Fonts] section of the pssqr.ini/unx file.

Data processing of Arabic/Hebrew and other languages normally written in right-to-left order is supported, but SQR for PeopleSoft does not generate reports in right-to-left page order.

Supplementary characters, such as Hong Kong characters in GB18030, are not supported in SQR PDF output.These characters are supported in HTML output. Using BI Publisher instead of SQR is recommended for supplementary character support in PDF files,or for Arabic and Hebrew support.

Arabic

The following table lists the supported encodings for the Arabic character set:

Encoding Parameter	Description	Output Support
CP720	Arabic - Transparent ASMO	PCL (8V), Flat file
CP708	ASMO708	Flat file
CP20240	IBM EBCDIC - Arabic	Flat file
CP28596	ISO 8859-6 (Arabic)	CSV, Flat file
Arabic	ISO 8859-6 (Arabic)	PCL (11N), CSV, Flat file
CP10004	Macintosh Arabic	CSV (*), Flat file
CP1256	MS Windows Arabic	PCL (9V), CSV, Flat file
CP864	MS-DOS Arabic	CSV (*), Flat file
MacArabic	Macintosh Arabic	Flat file

Baltic

The following table lists the supported encodings for the Baltic character set:

Copyright © 1988, 2026, Oracle and/or its affiliates.

Encoding Parameter	Description	Output Support
CP28594	ISO 8859–4 (Baltic)	CSV, Flat file
ISO-8859–4, Latin4	ISO 8859–4 (Baltic)	4N, CSV (*), Flat file
CP1257	MS Windows Baltic	CSV, Flat file
CP775	MS-DOS Baltic	PCL (19L), CSV (*), Flat file

Celtic

The following table lists the supported encodings for the Celtic character set:

Encoding Parameter	Description	Output Support
ISO-8859–14	ISO 8859–14 (Latin 8)	Flat file

Chinese (Simplified)

The following table lists the supported encodings for the Simplified Chinese character set:

Encoding Parameter	Description	Output Support
CCSID935	IBM EBCDIC 935	Flat file
GB18030	GB 18030-2000	Flat file
GB2312	GB 2312-80	PCL (18C), PS, , CSV, Flat file
HZ	HZ GB2312-80	Flat file
CP936	MS Windows Schinese/MS-DOS Schinese (GBK)	PCL (18C), PS. CSV, Flat file
EUC-CN	Simplified Chinese EUC	PS, CSV (*), Flat file

Chinese (Traditional)

The following table lists the supported encodings for the Traditional Chinese character set:

Encoding Parameter	Description	Output Support
Big5	Big5	PCL (18T), PS, CSV, Flat file
CCSID937	IBM EBCDIC 937	Flat file
CNS-11643-1986	CNS-11643-1986	CSV (*), Flat file
CNS-11643-1992	CNS-11643-1992	CSV (*), Flat file
GB12345	GB12345	Flat file
CP10002	Macintosh Traditional Chinese	Flat file
CP950	MS Windows Tchinese/MS-DOS Tchinese (Big5)	PCL (18T), PS, CSV, Flat file
EUC-TW	Traditional Chinese EUC	PS, CSV (*), Flat file
HKSCS	Hong Kong Supplementary Character Set	Flat file

Cyrillic

The following table lists the supported encodings for the Cyrillic character set:

Copyright © 1988, 2026, Oracle and/or its affiliates.

Encoding Parameter	Description	Output Support
CP20880	IBM EBCDIC - Cyrillic (Russian)	Flat file
CP21025	IBM EBCDIC - Cyrillic (Serbian, Bulgarian)	Flat file
CP28595	ISO 8859-5 (Cyrillic)	CSV, Flat file
ISOLatinCyrillic	ISO 8859-5 (Cyrillic)	PCL (5T), CSV, Flat file
CP10007	Macintosh Cyrillic	CSV (*), Flat file
CP1251	MS Windows Cyrillic (Slavic)	PCL (5T), CSV, Flat file
CP855	MS-DOS Cyrillic	CSV (*), Flat file
CP866	MS-DOS Russian	CSV (*), Flat file
CP20866	Russian — K018	CSV (*), Flat file
CP21866	Ukranian — K018 — RU	Flat file

Greek

The following table lists the supported encodings for the Greek character set:

Encoding Parameter	Description	Output Support
CP20423	IBM EBCDIC - Greek	Flat file
CP28597	ISO 8859-7 (Greek)	CSV, Flat file
Greek	ISO 8859-7 (Greek)	CSV, Flat file
CP10006	Macintosh Greek 1	CSV (*), Flat file
CP1253	MS Windows Greek	CSV, Flat file
CP737	MS-DOS Greek 437G	CSV (*), Flat file
CP869	MS-DOS Modern Greek	CSV (*), Flat file

Gurmukhi

The following table lists the supported encodings for the Gurmukhi character set:

Encoding Parameter	Description	Output Support
CP10010	Macintosh Romanian	CSV (*), Flat file

Hebrew

The following table lists the supported encodings for the Hebrew character set:

Encoding Parameter	Description	Output Support
CP38598	ISO 8859-8 (Hebrer Logical Ordering)	CSV, Flat file
Hebrew	ISO 8859-8 (Hebrer Logical Ordering)	CSV, Flat file
CP28598	ISO 8859-8 (Hebrer Visual Ordering)	Flat file
CP10005	Macintosh Hebrew	CSV (*), Flat file
CP1255	MS Windows Hebrew	CSV, Flat file

Copyright © 1988, 2026, Oracle and/or its affiliates.

Encoding Parameter	Description	Output Support
CP862	MS-DOS Hebrew	CSV (*), Flat file
MacHebrew	Macintosh Hebrew	Flat file

Icelandic

The following table lists the supported encodings for the Icelandic character set:

Encoding Parameter	Description	Output Support
CP10079	Macintosh Icelandic	CSV (*), Flat file
CP861	MS-DOS Icelandic	CSV (*), Flat file

Japanese

The following table lists the supported encodings for the Japanese character set:

Encoding Parameter	Description	Output Support
CP21027	Ext Alpha Lowercase	Flat file
CCSID-1027	IBM EBCDIC - Japanese	CSV (*), Flat file
CP20290	IBM EBCDIC - Japanese Kana Extension	Flat file
CCSID939, EBCDIK1027	IBM EBCDIC MBCS-HOST - Japanese (1027+ 0300)	CSV (*), Flat file
CCSID-290	IBM EBCDIK - Japanese Kana Extension	CSV (*), Flat file
CCSID930, EBCDIK290	IBM EBCDIK MBCS-HOST - Japanese (290+0300)	PDF, CSV (*), Flat file
CCSID-942	IBM MBCS-PC OS2 (1041+941) - Japanese	CSV (*), Flat file
ISO-2022-JP	ISO 2022-JP	CSV (*), Flat file
EUC-J, JEUC	Japanese EUC	PS, CSV (*), Flat file
JIS_X_0201	JIS X 0201	CSV (*), Flat file
JIS_X_0208	JIS X 0208	CSV (*), Flat file
CP10001	Macintosh Japanese	Flat file
Shift-JIS, SJIS	MS Windows Japanese/MS-DOS Japanese	PCL (19K), PS, CSV, Flat file
CP932	MS Windows Japanese/MS-DOS Japanese	PCL (19K), PS, CSV, Flat file

Korean

The following table lists the supported encodings for the Korean character set:

Encoding Parameter	Description	Output Support
ISO-2022-KR	ISO-2022-KR	CSV (*)
EUC-KR	Korean EUC	PS, CSV (*), Flat file
CP1361	Korean Johab	PS, CSV (*), Flat file

Copyright © 1988, 2026, Oracle and/or its affiliates.

Encoding Parameter	Description	Output Support
Johab	Korean Johab	PS, CSV (*), Flat file
CP10003	Macintosh Korean	Flat file
CP949	MS Windows Korean/MS-DOS Korean	PS, CSV, Flat file

Latin

The following table lists the supported encodings for the Latin character set:

Encoding Parameter	Description	Output Support
CP20277	IBM EBCDIC - Denmark/ Norway	Flat file
CP20278	IBM EBCDIC - Finland/Sweden	Flat file
CP20297	IBM EBCDIC - France	Flat file
CP20273	IBM EBCDIC - Germany	Flat file
CP20871	IBM EBCDIC - Icelandic	Flat file
CP500	IBM EBCDIC - International	CSV (*), Flat file
CP20280	IBM EBCDIC - Italy	Flat file
CP20833	IBM EBCDIC - Korean Extended	Flat file
CP20284	IBM EBCDIC - Latin America/ Spain	Flat file
CCSID1047	IBM EBCDIC - Latin1/Open System	Flat file
CP875	IBM EBCDIC - Modern Greek	CSV (*), Flat file
CP870	IBM EBCDIC - Multilingual/ ROECE (Latin2)	CSV (*), Flat file
CP20285	IBM EBCDIC - United Kingdom	Flat file
CP20269	ISO 6937 Non-Spacing Accent	Flat file
CP28591	ISO 8859-1 (Latin 1)	CSV (*), Flat file
ISO-8859-1, Latin1	ISO 8859-1 (Latin 1)	PCL (0N), PS, CSV, Flat file
ISO-8859-15	ISO 8859-15 (Latin 9)	CSV (*), Flat file
ISO-8859-2, Latin2	ISO 8859-2 (Central Europe)	PCL (2N), CSV, Flat file
CP28593	ISO 8859-3 (Latin 3)	CSV (*), Flat file
CP10082	Macintosh Croatia	CSV (*), Flat file
CP10029	Macintosh Latin2	CSV (*), Flat file
CP1252	MS Windows Latin1	PCL (19U), PS, CSV, Flat file
CP850	MS-DOS Multilingual Western Europe	PCL (12U), CSV (*), Flat file
CP860	MS-DOS Portuguese	Flat file
CP20261	T.61	Flat file
hp-roman8	HP Roman8	PCL (8U), Flat file

Copyright © 1988, 2026, Oracle and/or its affiliates.

Latin (Canadian French)

The following table lists the supported encodings for the Latin (Canadian French) character set:

Encoding Parameter	Description	Output Support
CP863	MS-DOS Canadian French	CSV (*), Flat file

Latin (Central European)

The following table lists the supported encodings for the Latin (Central European) character set:

Encoding Parameter	Description	Output Support
CP28592	ISO 8859-2 (Central Europe)	CSV, Flat file
CP1250	MS Windows Central European	PCL (9E), CSV, Flat file

Latin (Southeast European)

The following table lists the supported encodings for the Latin (Southeast European) character set:

Encoding Parameter	Description	Output Support
ISO-8859-3, Latin3	ISO 8859-3 (Latin 3)	CSV, Flat file

Latin (U.S. English)

The following table lists the supported encodings for the Latin (U.S. English) character set:

Encoding Parameter	Description	Output Support
CP20105	IA5 IRV International Alphabet No.5	Flat file
EBCDIC, CP037	IBM EBCDIC - U.S./Canada	PS, CSV (*), Flat file
CP037	IBM EBCDIC - U.S./Canada	CSV (*), Flat file
CP437	MS-DOS U.S.	CSV (*), Flat file
ASCII, ANSI	US-ASCII	PCL (0U), PS, CSV, Flat file

Latin (Western European)

The following table lists the supported encodings for the Latin (Western European) character set:

Encoding Parameter	Description	Output Support
CP10000	Macintosh Roman	CSV (*), Flat file

Malayalam

The following table lists the supported encodings for the Malayalam character set:

Copyright © 1988, 2026, Oracle and/or its affiliates.

Encoding Parameter	Description	Output Support
CP10017	Macintosh Malayalam	Flat file

Nordic

The following table lists the supported encodings for the Nordic character set:

Encoding Parameter	Description	Output Support
ISO-8859-10, Latin6	ISO 8859-10 (Latin 6)	PCL (6N), Flat file
CP865	MS-DOS Nordic	CSV (*), Flat file

Slavic

The following table lists the supported encodings for the Slavic character set:

Encoding Parameter	Description	Output Support
CP852	MS-DOS Slavic	PCL (17U), CSV (*), Flat file

Symbol

The following table lists the supported encodings for the Symbol character set:

Encoding Parameter	Description	Output Support
Adobe-Symbol-Encoding	Adobe Symbol Encoding	Flat file
CP10008	Macintosh RSymbol	Flat file

Thai

The following table lists the supported encodings for the Thai character set:

Encoding Parameter	Description	Output Support
CP20838	IBM EBCDIC - Thai	Flat file
Thai	ISO 8859-11 (Thai)	CSV, Flat file
CP874	MS Windows Thai/MS-DOS Thai	CSV, Flat file

Turkish

The following table lists the supported encodings for the Turkish character set:

Encoding Parameter	Description	Output Support
CP20905	IBM EBCDIC - Turkish	CSV (*), Flat file
CP1026	IBM EBCDIC - Turkish (Latin 5)	CSV (*), Flat file
CP28599	ISO 8859-9 (Latin 5)	CSV, Flat file
ISO-8859-9, Latin5	ISO 8859-9 (Latin 5)	PCL (5N), CSV, Flat file
CP10081	Macintosh Turkish	CSV (*), Flat file
CP1254	MS Windows Turkish	PCL (5T), CSV, Flat file

Copyright © 1988, 2026, Oracle and/or its affiliates.

Encoding Parameter	Description	Output Support
CP857	MS-DOS Turkish	PCL (9T), CSV (*), Flat file

Unicode

The following table lists the supported encodings for the Unicode character set:

Encoding Parameter	Description	Output Support
Java	Java Encoding (represents Unicode in US-ASCII)	Flat file
Big-Endian	Unicode Big-Endian Order	CSV (*), Flat file
BMP	Unicode BMP	Flat file
Little-Endian	Unicode Little-Endian Order	CSV, Flat file
UCS-2, UCS2	Unicode UCS-2 Encoding	CSV (*), Flat file
UTF-8, UTF8	Unicode UTF-8 Encoding	CSV (*), Flat file
UTF8-EBCDIC	UTF-8 EBCDIC	Flat file

Vietnamese

The following table lists the supported encodings for the Vietnamese character set:

Encoding Parameter	Description	Output Support
CP1258	MS Windows Vietnam	CSV, Flat file

Using PS/nVision Reporting for Global Implementations

In PS/nVision, you can specify 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 several instances of a report in a requested alternate language or group of languages. To that end, the features are designed to present reports to end users in their preferred languages. Thus, there is a distinction between the designer's language and the user's language.

> **Note**
>
> Your Microsoft Windows locale must be able to support the sign-in language.

Most of the strings in PS/nVision output are fetched from the Strings table, and they are delivered with the report instance. However, many labels used in macros and dialog boxes have to be translated directly in the file, as they cannot be fetched at runtime. When a PS/nVision report happens to contain labels in macros or dialog boxes, the layout needs to be translated in Microsoft Excel, and a separate copy of the layout maintained for each language. In this case, each language has its own version of the file, and the same rules for locating the appropriate language of the PS/nVision layout spreadsheet files. Each PS/nVision layout and drill-down directory can contain a subdirectory for each language. PS/nVision searches the

Copyright © 1988, 2026, Oracle and/or its affiliates.

appropriate directory for the user's current language, and performs the appropriate fallback if a translation is not found.

Language-sensitive features of PS/nVision can be separated into two distinct parts:

Field or Control	Description
Design-Time Features	These are features related to designing PS/nVision layouts.
Runtime Features	These are features activated when a user makes a PS/nVision report request.

Related Topics

- PS/nVision

PS/nVision Design-Time Globalization Features

The following design-time features of PS/nVision adapt to the current user's language, enabling you to build a layout that can produce reports in multiple languages:

Field or Control	Description
Column Headings	When the user selects a column from a list, the list appears 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 appears in the user's language. Depending on the structure of your tree, this may require adding related language records to one or more prompting views.

Copyright © 1988, 2026, Oracle and/or its affiliates.

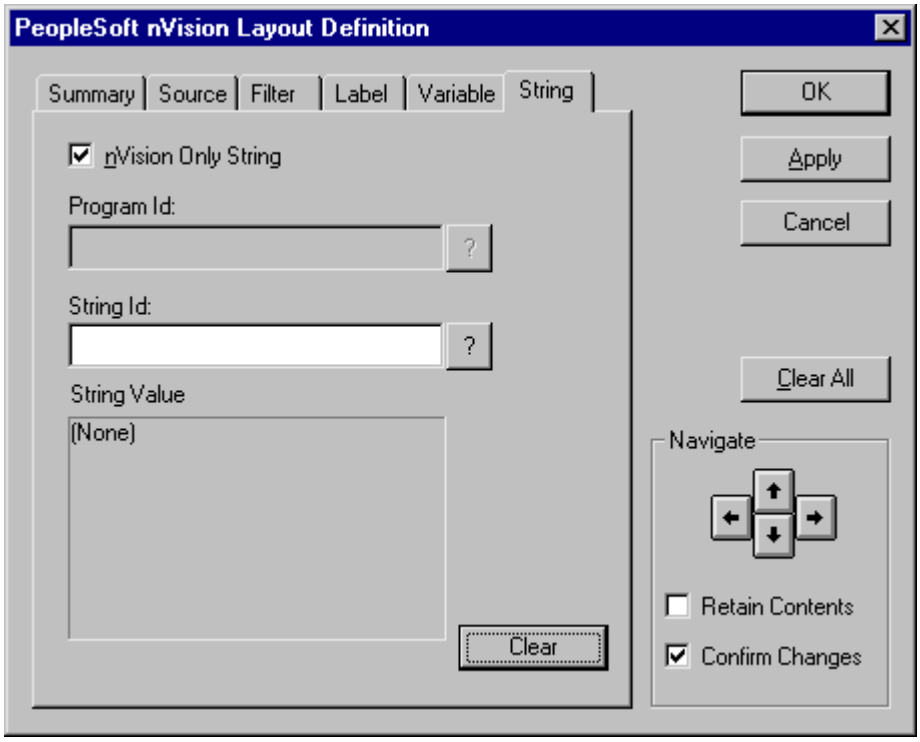
Field or Control	Description
String Variables	Layouts typically contain a considerable amount of constant text, such as the column headings Last Year to Date and Current Budget. PS/nVision provides the option to build multilingual layouts where these text strings are replaced by specially formatted strings 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 used on similar reports. You can look up the *name* in the Strings table using the Strings Table utility. **Note:** This approach is similar to the one used by PeopleSoft application SQR programs. Names are case sensitive.

Setting Up String Criteria

Use the PeopleSoft nVision Layout Definition dialog box: String tab to set up string criteria for the current cell selection:

This example illustrates the fields and controls on the PeopleSoft nVision Layout Definition dialog box: String tab.

Copyright © 1988, 2026, Oracle and/or its affiliates.



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.

   Use this tab to insert a string from the Strings table into a cell in your layout.
3. Clear the **nVision Only String** 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 select from all available strings, clear the nVision Only String box.
4. If the **nVision Only String** box is cleared, select a **Program ID**.

   > **Note**
   >
   > If the nVision Only String check box is selected, the program ID is *NVISION.*
5. In the **String ID** field, select the ID of the string you want to insert.

   Select any of the strings assigned to the program ID that you specified.
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** box. This preserves all the dialog box information when you navigate to a new cell selection. Then repeat steps 3 through 5 to define additional string criteria.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Related Topics

- Using the Strings Table for Language-Sensitive Text in Reports
  The PeopleTools Strings table (STRINGS_TBL) stores text strings used for language-sensitive labels and other text for PS/nVision and SQR reports. This avoids hard-coding labels into the report files themselves. The use of strings rather than hard-coded text in reports enables translators to translate the report layout in the database without editing 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 and report logic.

PS/nVision Run-Time Language Features

The features described below are activated when a user makes a PS/nVision report request.

Path Search

As PS/nVision looks for a file (such as a Microsoft Excel spreadsheet, layout, or template), it steps through the directories listed in the appropriate paths. The paths are defined in different ways for the PeopleTools development environment and the Web. For the PeopleTools development environment, the paths are defined in PeopleSoft Configuration Manager. In the web, the paths are defined in the [nVision] section of the PeopleSoft Process Scheduler configuration file (psprcs.cfg).

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.

For example, when searching for NVSUSER files for a user with France as the preferred language and the defined path for macros is *PS_HOME*\EXCEL, PS/nVision searches for *PS_HOME*\EXCEL\FRA directory first. If the files are not found, PS/nVision uses the files in the *PS_HOME*\EXCEL.

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 formatting. So, if a French user runs a PS/nVision report, PeopleTools first looks in the FRA directory under the PS/nVision directory that is defined for an installation. If the required report layout isn't found in the FRA directory, PeopleTools uses the generic report layout in the base PS/nVision directory.

Request Language

The PS/nVision Report Request page includes a **Language Template** text box on the Advanced Options page. In the **Foreign Language Translation** field, enter an alternate language code to automatically generate a translated report. If you are applying a scope to the report request, enter a string containing one or more PS/nVision variables.

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}%
```


Copyright © 1988, 2026, Oracle and/or its affiliates.

Note that the field names required vary, depending on the type of scope you're using. However, 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 uses 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*, *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 can be found on the same node or detail value table associated with the *scopefield*. Typically, the field name used is *LANGUAGE_CD.*

The following example shows a common implementation of this variable (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*, then 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* change, 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.

Labels

When retrieving node or detail row and column labels, PS/nVision uses the appropriate alternate language record, if one has been defined through 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.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Related Topics

- PS/nVision: Creating Requests
- PS/nVision: Understanding Report Scopes

International Versions of Microsoft Excel

Microsoft Excel is available in numerous languages. PeopleTools is designed to work with all international editions of the supported release of Microsoft Excel.

> **Note**
>
> Downloaded data such as numbers and dates are formatted in the Microsoft Excel spreadsheet according to the regional settings on the user's machine.

BI Publisher

Oracle's Business Intelligence (BI) Publisher supports translation of report templates. You can create template translation files for the template's "translatable units" only when the report template or sub-template is in the RTF format. When the translation exercise is complete, the translation file is uploaded and integrated into the BI Publisher translation system.

More information on template translation can be found in the BI Publisher PeopleBook.

See BI Publisher for PeopleSoft: Maintaining Template Translations.

Copyright © 1988, 2026, Oracle and/or its affiliates.

10

Developing Global Applications

Developing Easily Translatable Applications

When designing application objects, especially pages, follow these guidelines to ensure easy translation:

- Avoid using page-based text (static text)in page element labels and in other labels that can be derived from field labels (such as query heading names and string definitions in the Strings table [STRINGS_TBL]).

  Overriding field labels on the page with page-based text associates a new label with a field that is specific to the current page. The new label cannot be shared with other pages in the system, so if the new label is used multiple times, it requires translation for each occurrence. Instead, associate the new label with the field that it is labeling, and choose your new label for the occurrence of the field on your page. That way, the label can be used on other pages without having to be translated again.

- Starting with PeopleTools 8.53, HTML objects and free-form style sheets no longer have related language records. If you plant to include translatable text in these objects, use dynamic text instead of static text.

- Design pages with enough room to accommodate object labels that become longer when translated.

  A good guideline if developing in English is to allow at least 30 percent plus two characters of extra space between a label and its field. If you enable translatability options in Application Designer, a dotted line translation buffer will visually show the amount of space to allow, and will warn you if this buffer overlaps another object on the page.

  Field labels for edit boxes should have alignment = right and position = left. This puts the labels at a consistent distance to the left of the field, and allows the text to expand out to the translation buffer on the left. Newer PeopleSoft pages use this alignment and also have the colon display set to off, for a cleaner look.

  Buttons should be visually checked to ensure they have enough expansion space for translations.

  If static text is avoided and enough space is left in page elements, you will be able to eliminate per-language page realignment in the translation process. The page text values will match the session language dynamically, and the translations will fit without readjustment.

- Translate field labels first. Translated field labels appear throughout the system in page element labels, search dialog box field labels, record field names, Strings table definitions, query heading labels, and so on. Translating these labels before translating other objects helps you to build a glossary of terms for your application and give you a head start in translating each page, as the majority of text on a page is derived from field labels.

  Always use dynamic text rather than hard coded strings in HTML objects. As of 8.52, these objects are read from and written to the base table only, similar to PeopleCode objects.

- Identify strings that should not be translated. These patterns are identified as **DNT** (do not translate) by Translation Factory:

Copyright © 1988, 2026, Oracle and/or its affiliates.

  - All capital and > 4 characters. Example: CONTINUE
  - Words with symbols (_ or $ or a number). Example: Total_amount, address1
  - Words without spacing. Example: referencedocument, ItemNumber
  - Tokens (words beginning with %). Example: %1 or %User
- Avoid polysemic words in labels.
- Use capital letters without spaces for key values.
- Eliminate leading space in field values.
- Restrict long fields to < 4K characters.
- Keep spacing and a separator such as, | or line break for hyperlink. This is to ensure availability of buffer space during translation.

Selective Translation

If you add custom functionality or data tables to a PeopleSoft application, you might want to translate the new PeopleTools objects and data fields for display in multiple languages. PeopleTools is designed to permit selective translation of the elements in an application database.

Because the system displays base language versions of any language-sensitive elements for which no translation is provided, you can translate only those pages, menus, messages, and so on that are required by users who are working in non-base languages. Often, you can easily determine if the page will be accessed only by administrative staff who share a common language with developers, or if it will be deployed to end users widely and will therefore require translation.

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 sees the object in the base language of the system.

Designing Global-Ready Pages

You can simplify a global development project and make a global system easier to maintain by making base language pages as translation-ready as possible. Eliminate textual elements that are maintained on the page definition and instead derive 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 takes effect across all pages where that string is referenced.

To design 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 in Application Designer, use the RFT Short or RFT Long setting whenever possible. This setting causes the control labels, as well as the button 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, such as page-based floating text labels or objects with value of Static Text, on the page.
- Associate group boxes with record fields.

  Use the Page Field Properties dialog box in Application Designer to 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.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Adding and removing page objects and PeopleCode on a page takes effect across all languages; however, the layout of a page is language-specific. After you start translating pages, any alignment or page layout changes you make to the base language page must be reapplied to each translation of the page.You can minimize this impact by designing pages with no static text, and enough translation buffer space so that realignment is never needed.

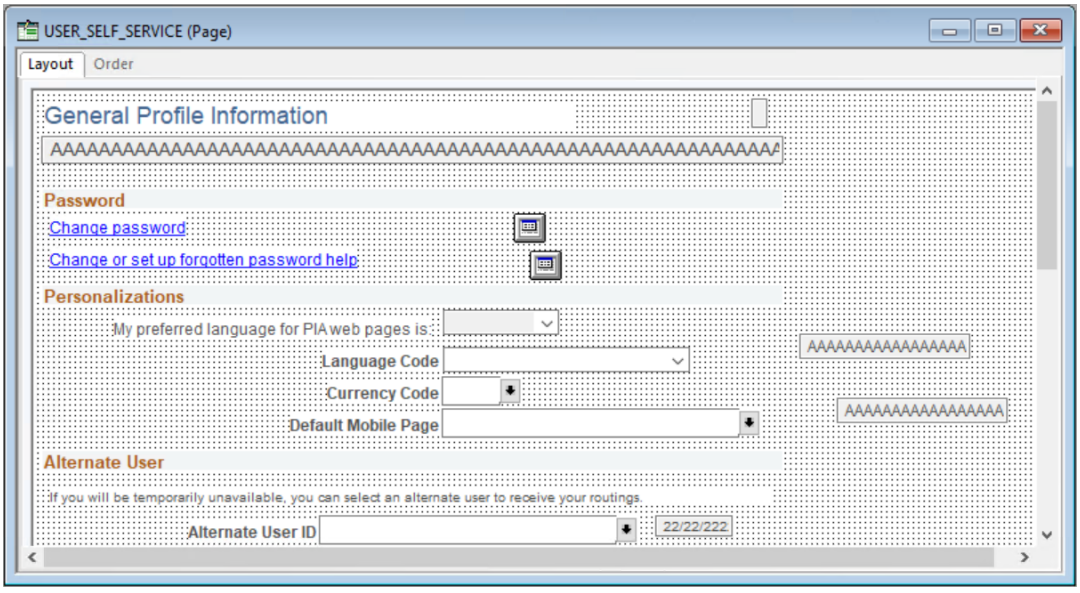
- Size and arrange page controls so that there is enough space to accommodate data in non-base languages.

  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, you see a translation buffer indicator. This translation buffer indicator is 30 percent plus two characters 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. Ensure that there is enough space so that the translation buffer indicator doesn't overlap the field or other page controls. This example shows both the label text box and the translation buffer indicator for the Description text label for a page within Application Designer:

  Edit box labels should generally be right aligned and positioned left, so that the right edge of the label lines up at a constant distance from the field and translation expansion will increase to the left without affecting the field positioning.

  Following is an example of My System Profile page, USER_SELF_SERVICE. This example has the Currency Code label selected and shows the translation buffer (dotted line) to indicate the amount of space recommended to leave for translations.

  This example illustrates the fields and controls on the Field label and translation buffer indicator.



  The Application Designer PeopleBook describes how to position page control labels to prevent translation buffer overlap.

- Validate Labels for Translation

  See Application Designer Developer's Guide: Positioning Page Control Labels.

Copyright © 1988, 2026, Oracle and/or its affiliates.

In Application Designer, select Tools, Options on the General tab, select Enable Translatability Options. When you save each page, a warning will be issued if there is an overlap, that could cause translations to wrap when the page is displayed in a browser.

The Translatability Options feature also warns if UI intensive elements do not allow enough character storage expansion buffer in the database field. The affected elements are Field Label Short values, which have an English maximum of 10, and Xlat Short values, which have an English maximum of 6,when this feature is enabled. This is to ensure there is enough room for translations expansion in the database. Using long values instead of short values will allow more expansion space for translations in the database. Verifying that the translation buffer (dotted line) does not overlap other objects will help prevent layout issues such as wrapping or truncation when the page is viewed with translations. Setting the colon display to off will result in a cleaner looking page. Together, these standards promote better readability and translatability and conform to the recommended look and feel for PeopleSoft applications.

Copyright © 1988, 2026, Oracle and/or its affiliates.

11

Using Related Language Tables

Understanding Related Language Tables

This section discusses related language tables.

Related Language Tables Overview

PeopleTools can store multiple translations of application data and PeopleTools objects in a single database. Each PeopleSoft database has a single *base language*. The base language of a new PeopleSoft database upon installation is always English, but can be changed by an administrator. The base language should be the language most commonly used by application users, and is the language in which data is stored in the core PeopleSoft tables known as *base language tables*.

All PeopleTools objects, such as pages, fields and queries, can be maintained in multiple languages. Descriptions of application data elements, such as departments, locations and job codes, can also be maintained in multiple languages.

The key to maintaining this data in multiple languages is the use of special tables known as related language tables. A *related language table* stores descriptions and other language-sensitive elements in all languages other than the base language of the database. In this way, while any table in the database can store data in the base language of that database, only tables that have related language tables can maintain the same data in multiple languages simultaneously. For example, it is unlikely that you will want to maintain the descriptions of your general ledger journal lines in multiple languages − the sheer volume of the journal lines in most systems would preclude any effort to maintain translations of their descriptions. The cost of hiring a translator to translate each journal line would be prohibitive, and in most cases only the person entering the journal line, and possibly their supervisor, would be likely to want to view it again. However for frequently used values, such as a chart of accounts, many users across your entire organization would often need to refer to this data, and so you would most likely want to maintain the descriptions of each ChartField entry in each language spoken by your users.

In this case, you would not need a related language table for your Journal Lines table, as you would be maintaining Journal Line descriptions in a single language only, and this would be in the base table, but you would need a related language table for each of your ChartField tables.

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 following:

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

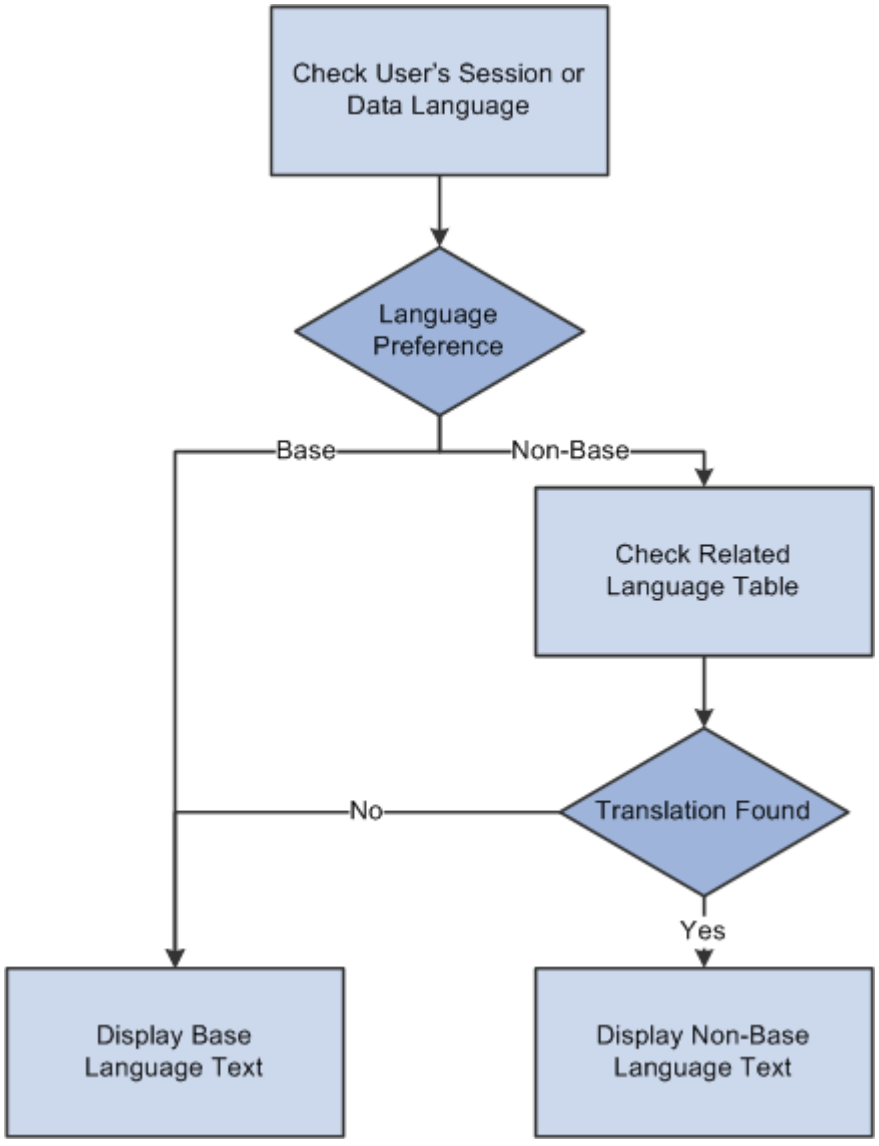
The *language preference* applies to both the session language and the data language and can be different for each.

Copyright © 1988, 2026, Oracle and/or its affiliates.

> **Note**
>
> In the case of Application Designer, the language preference is determined by the PeopleSoft Configuration Manager language setting.

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.

This diagram illustrates the flow of execution of translated language text



Synchronization of related language tables with their base tables is automatically maintained by PeopleTools. When you translate a language-sensitive field, the system adds a new row to

Copyright © 1988, 2026, Oracle and/or its affiliates.

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.

> **Note**
>
> PeopleTools automatically handles the storage and display of data in related language tables and their synchronization with corresponding entries in the base language tables based on each user's language preference. However, PeopleTools cannot automatically translate data! The PeopleSoft system provides translations of PeopleSoft Pure Internet Architecture components and user interfaces for most application products into several languages. Also, translations of several key application data tables, such as country and currency codes, are provided by the PeopleSoft system. However, you will need to plan a way to translate application data entered by your users that needs to be maintained in multiple languages. Some users who are familiar with several languages may want to translate the data elements they enter themselves using the Multilingual Data Entry Support feature. In other cases, you may want to bring in translators periodically to translate key elements into each language used in your organization. PeopleTools provides an architecture to store these translations, but does not perform "machine translation" or provide other automated linguistic translation technologies.

Related Topics

- Understanding International Preferences
  The international preference settings control the language in which users view and use the PeopleSoft system, as well as locale formatting conventions for dates, times, and numbers. The PeopleSoft system treats language preferences and local formatting preferences separately.
- Editing Data in Multiple Languages
  This section provides an overview of multi-language entry and discusses how to enter data in multiple languages.

Related Language Table Structure

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

Each table in your database that is not a related language table is, by definition, a base language table. The Message Catalog table, PSMSGCATDEFN, is one example of a base table. Note that it has a DESCRLONG field—which is typically language-sensitive—as well as MESSAGE_TEXT, which is also language-sensitive.

This diagram illustrates related language table structure

Copyright © 1988, 2026, Oracle and/or its affiliates.

PSMSGCATDEFN (Record)

Record Fields | Record Type

Num	Field Name	Type	Len	Format	Short Name	Long Name
1	**MESSAGE_SET_NBR**	Nbr	5		Set	Message Set Number
2	**MESSAGE_NBR**	Nbr	5		Msg	Message Number
3	**MESSAGE_TEXT**	Char	100	Mixed	Message	Message Text
4	MSG_SEVERITY	Char	1	Upper	Severity	Message Severity
5	LAST_UPDATE_DTTM	DtTm	26		Last Updt	Last Update Timestamp
6	DESCRLONG	Long	0		Descr	Description

PeopleSoft has therefore created a new record definition, PSMSGCATLANG, to act as the related language table for PSMSGCATDEFN. Related language records are defined in Application Designer just like any other record definitions. However, as they often contain many of the same fields as their base language table, it is often quicker to create the related language table by *cloning* the base table. To clone a base table, open the base language table and the use the Save As option on the File menu and enter a name for the new record. Once cloned, any fields not required in the related language record can be deleted from the new record.

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**, **DESCRSHORT** or DESCRLONG; or **LONGNAME** or **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, PSMSGCATLANG, meets all of these requirements. Note that both language-sensitive fields, DESCRLONG and MESSAGE_TEXT, appear in PSMSGCATLANG.

This diagram illustrates related language table structure displaying DESCRLONG and MESSAGE_TEXT

PSMSGCATLANG (Record)

Record Fields | Record Type

Num	Field Name	Type	Len	Format	Short Name	Long Name
1	MESSAGE_SET_NBR	Nbr	5		Set	Message Set Number
2	MESSAGE_NBR	Nbr	5		Msg	Message Number
3	LANGUAGE_CD	Char	3	Upper	Lang Cd	Language Code
4	MESSAGE_TEXT	Char	100	Mixed	Message	Message Text
5	DESCRLONG	Long	0		Descr	Description

The base table must be explicitly associated with its related language table.

To associate a base table with its related language table:

1. Open the Record Properties dialog box for the base table record.

Copyright © 1988, 2026, Oracle and/or its affiliates.

2. Click the Use tab.
3. Under Record Relationships, in the **Related Language Record** field enter the name of the related language table.

> **Note**
>
> There is a one-to-one relationship between the base record and the related language record. Also, the record type of the base record and the related language record should be the same.

4. Click **OK.**

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 PSMSGCATDEFN table.

MESSAGE_SET_NBR	MESSAGE_NBR	MESSAGE_TEXT	MSG_SEVERITY	DESCRLONG
1	1	Distributed Object Manager: Help Name=%1 Language=%2	M	A Help Text object is being retrieved from the database. This message is for your information only. It is not an error or a warning.

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

MESSAGE_SET_NBR	MESSAGE_NBR	LANGUAGE_CD	MESSAGE_TEXT	DESCRLONG
1	1	ESP	Gestor de objetos distribuidos: Nombre de ayuda=%1 Idioma=%2	Se está recuperando un objeto texto de ayuda de la base de datos. Este mensaje sólo es informativo. No es un error ni un aviso.

Copyright © 1988, 2026, Oracle and/or its affiliates.

MESSAGE_SET_NBR	MESSAGE_NBR	LANGUAGE_CD	MESSAGE_TEXT	DESCRLONG
1	1	FRA	Gestionnaire d'objets répartis: Aide=%1 Langue=%2	Un objet de type Aide en ligne est extrait de la base de données. Ce message est affiché seulement à titre d'information. Ce n'est ni un message d'erreur, ni un message d'avertissement.

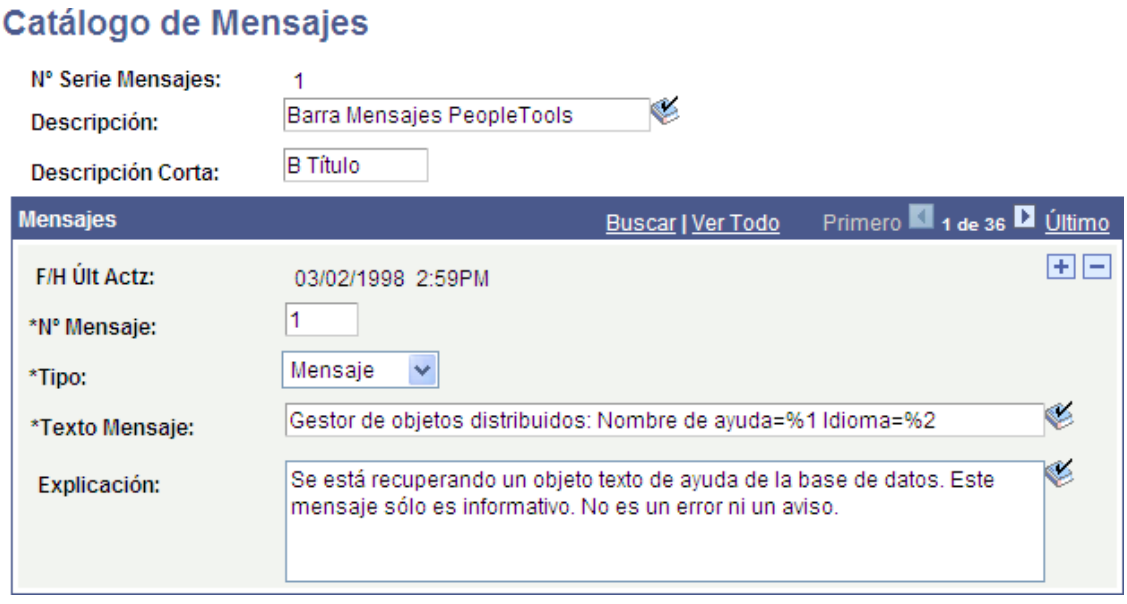
How Related Language Tables Are Used

Related language tables are the primary means of storing language translations in the PeopleSoft system. They store translations of descriptive text for 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, and, if the user's language preference is 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 Message Catalog page appears when the user's language preference is Spanish (ESP). The Texto Mensaje (MESSAGE_TEXT) field and the Explicación (DESCRLONG) field present translations from the PSMSGCATLANG related language table.

This is an example of the Message Catalog page in Spanish

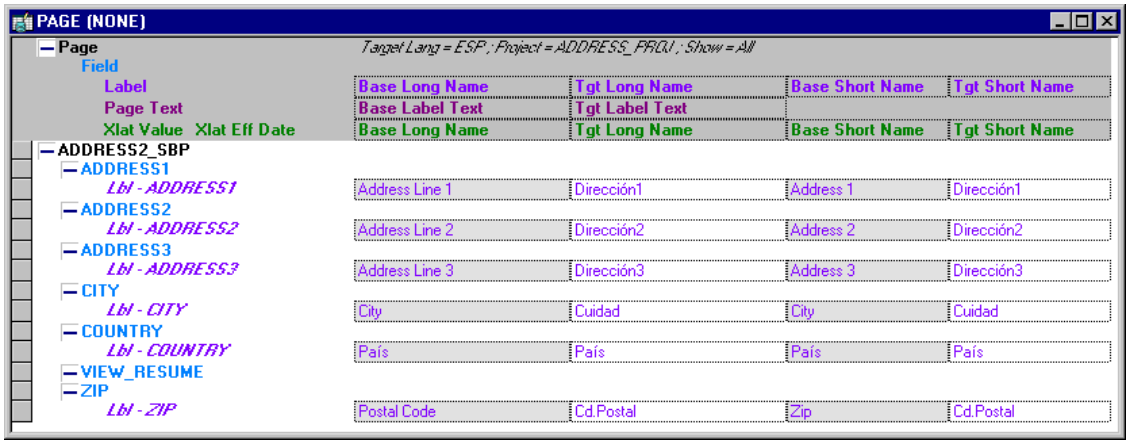


Many PeopleTools object definitions, including fields, records, and components, also have associated related language tables. This makes maintenance of multiple language

Copyright © 1988, 2026, Oracle and/or its affiliates.

PeopleTools objects, and therefore the system user interface, 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.

This is an example of the fields on the ADDRESS2_SBP subpage in Spanish



Because the fields are translated, the translated versions of field labels appear 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, whenever the base labels change, you must also update the translations of the object names.

Related Topics

- Understanding Data Editing in Related Language Tables and Base Tables
  When a user changes data on a language-sensitive page, the system responds by changing the contents of the base table, the related language table, or both, depending on the language of the user and which field was modified. The following table explains at the database table level the actions that are performed when you work with language-sensitive tables:
- Understanding Application Definition Translation
  If you are running PeopleSoft applications with a multiple language user interface, you should translate any customizations you make to the product using PeopleTools into each of the languages that you are using. PeopleTools provides a full suite of tools to help you translate the application user interface; these are the same tools that PeopleSoft uses internally to provide the translations we ship on the global multi-language CDs.

Installing Oracle-Provided Translations

This section discusses translation installation.

Understanding Translation Installation

Oracle provides translations of the user interface and key application data for PeopleSoft application products.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Typically, you would install these translations when you first install your database. However, if you did not install the Oracle-provided translations during your initial install, this section describes the steps to follow to add translations to an already-existing database.

The database translations are delivered in a Data Mover file with the format xxyyya.db, where xx is the product code, and yyy is the language code, for instance, hcfraa.db. The location of the .db file will typically be in PS_APP_HOME\data, and can also be found in pi_home\xxx\ComPlat\DBFILE\bse\data, where xxx is the language code.

> **Note**
>
> The size impact on the database of installing translations is difficult to predict exactly, but a general rule of thumb is approximately 5% per language.

Adding a New Language

This section explains how to add a language which was not initially installed.

Start with a production or copy of production database with the latest PUM applied, leaving it at the PeopleTools patch level that was delivered in the PUM Image. A standalone DEMO database can also be used if this is for demo rather than production purposes.

1. In Data Mover, create and run a script such as *hcfra.dms* to load the .db file. Here is an example for loading French, for HR92:

   ```
   SET LOG hcfra.log;
   SET INPUT hcfraa.db;
   SET UPDATE_DUPS;
   SET ENABLED_DATATYPE 9.2;
   SET UNICODE ON;
   IMPORT *;
   ```

2. After the import is complete, run the following SQL statement, replacing XXX with the language code.

   ```
   UPDATE PSLANGUAGES SET INSTALLED=1 WHERE LANGUAGE_CD='XXX';
   ```

3. Run the VERSION Application Engine program to update your PeopleTools version numbers.
4. Run the PTIACLNLNGCA Application Engine program.

   ```
   psae.exe -CT <DBTYPE> -CD <DBNAME> -CO <USERID> -CP <USERPWD> -R
   <RUNCONTROLID> -AI PTIACLNLNGCA -FP <TEMPDIR>
   ```

   See the product documentation for PeopleSoft 9.2 Application Installation for your database platform, Cleaning Up Orphaned Language Data.
5. Run the PSXPCLEAN Application Engine program to remove XML Publisher orphans.

   ```
   psae.exe -CT <DBTYPE> -CD <DBNAME> -CO <USERID> -CP <USERPWD> -R
   <RUNCONTROLID> -AI PSXPCLEAN
   ```

6. Run the database audits.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Refer to the sections about creating and checking a database in your installation guide for more information.

7. If you have made customizations, evaluate them to see if they need to be applied or if they need to be translated into the new language.
8. If you will be using nVision or two tier clients like Configuration Manager in the newly added language, copy over non-database files specific to the language (nVision files in excel\xxx directory, *xxx.dll files from the PS_HOME client bin directory).

Once the new language translations are added to the production or copy of production database, and PSLANGUAGES.INSTALLED=1, future updates through PUM and Change Assistant will detect and update the newly added language.

Swapping the Base Language

This section provides an overview of base language swapping, lists prerequisites, and discusses steps related to it.

Understanding Base Language Swapping

As delivered, the PeopleSoft database has a base language of English. The system has English data in its base language tables and, if you have installed the PeopleSoft-provided translations, non-English data in the related language tables. To choose a language other than English as your base language, you must move all English data to the related language tables (with an appropriate language code) and 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 must go through the same process.

There is a slight performance increase for the base language over the non-base languages, but there is extra administrative over head required when maintaining a swapped base environment. Take both of these into account if you are deciding whether to swap the base language.

PeopleSoft Data Mover provides the following command to automate this process:

```
SWAP_BASE_LANGUAGE target_language;
```

To swap the base language of your database, you must identify the language to swap in as the new base language. PeopleSoft 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.

You can check the existing base language by looking in the PeopleTools Options page. To navigate to the PeopleTools Options page, select **PeopleTools**, and then **Utilities**, and then **Administration**, and then **PeopleTools Options**. You can't change the base language setting in this page; it is for display purposes only. To change the base language, use the `SWAP_BASE_LANGUAGE` PeopleSoft Data Mover command.

Prerequisites

Before swapping the base language of your database, you must have:

- A clean audit report to ensure that there are no data integrity problems.

Copyright © 1988, 2026, Oracle and/or its affiliates.

- Sufficient space and resources to run the swapping process.

Running the Swap Audit Report

The swap audit report, SWPAUDIT.SQR, identifies data integrity problems that will cause errors during the swap process. This SQR program does not have a run control and, therefore, cannot be run with PeopleSoft Process Scheduler. For information on how to run SQR programs manually in your environment, refer to the section on creating a database in *PeopleSoft 9.2 Application Installation* for your database platform.

The swap audit report might list one or more of the following errors. Refer to the following instructions to correct these errors. Rerun the audit until it reports no errors before continuing with the swap process.

If the key structure is not valid between the base and RLR table, subsequent SWAP checks may fail. So, you may need to correct the key structure before rerunning the SQR to check the other SWAP checks.

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

Audit	Error Description	Resolution
SWAP-1	These are 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.	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 and remove its association with the related language record in the Object Properties dialog box.
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.	Make **LANGUAGE_CD** into a key field on each specified related language record.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Audit	Error Description	Resolution
SWAP-3	The following Related Language View(s) have the wrong structure defined: A related language record must share all of the same keys as its base language record, plus an additional key field (LANGUAGE_CD). Also, any non-key fields must also be in the base language record.	Correct the field structure for the specified related language views to ensure that they conform to this requirement. Then, recreate the views in the database.
SWAP-4	The following Related Language Table(s) have the wrong key structure defined: A related language record must share all of the same keys as its base language record, plus an additional key (**LANGUAGE_CD**).	Correct the key structure for the specified related language tables to ensure that they conform to this requirement. Then, alter the tables in the database to match the new key structure.
SWAP-5	The following Related Language Table(s) have the wrong structure defined: A related language record must share all of the same keys as its base language record, plus an additional key field (LANGUAGE_CD). Also, any non-key fields must also be in the base language record.	Correct the field structure for the specified related language tables to ensure that they conform to this requirement. Then, alter the tables in the database to match the new field structure.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Audit	Error Description	Resolution

SWAP-6	The following Related Language Table(s) have the Orphan Row defined:
	For each row on the related language record there must be a single row on the base table with matching keys. An orphan row is a row of data on the related language record that does not have a corresponding parent row on the base table. You must delete orphan rows from related language tables.
		Run the PTIACLEANLNG application engine program to clean orphan rows or use your platform's query tool to select against the related language table using the fields listed in the report. For every row in the report, there is an orphan row in a related language table. Perform a SELECT command first to ensure that you are getting the same row count as the report, then use the DELETE command to delete the selected rows.
		The following provides example code for an Oracle database. In this example, *xxx* is equal to the language code to which you are swapping.
		
		`SELECT * FROM PSMSGCATLANG WHERE NOT EXISTS (SELECT 'X' FROM PSMSGCATDEFN A WHERE PSMSGCATLANG.MESSAGE_SET_NBR = A.MESSAGE_SET_NBR AND PSMSGCATLANG.MESSAGE_NBR = A.MESSAGE_NBR) AND LANGUAGE_CD = 'xxx'`
		`DELETE FROM PSMSGCATLANG WHERE NOT EXISTS (SELECT 'X' FROM PSMSGCATDEFN A WHERE PSMSGCATLANG.MESSAGE_SET_NBR = A.MESSAGE_SET_NBR AND PSMSGCATLANG.MESSAGE_NBR = A.MESSAGE_NBR) AND LANGUAGE_CD = 'xxx'`
		`SELECT * FROM Related_Language_Record A WHERE NOT EXISTS (SELECT 'X' FROM Base_Record B WHERE A.Field_Name =`

Copyright © 1988, 2026, Oracle and/or its affiliates.

Audit	Error Description	Resolution
		B*Field_Name ...for each field name listed*) AND LANGUAGE_CD = '*xxx*'
SWAP-7	The following base records do not have a Unique Index: A base record that has a corresponding related language record must have a unique index—that is, a key field.	Open each base record listed in the report and select a field that you want to make a key field. Then alter the tables in the database to match the new key structure.
SWAP-8	The following base records and the Related Language Records have different record types: There is a mismatch between the record type of the base record definition and its related language record definition.	Create a new related language record definition and assign it (or reassign an existing related language definition) to the base record, making sure that both the base record and related language record definition are the same record type.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Audit	Error Description	Resolution

SWAP-9	The following translated BI Publisher template files are orphans:
	An orphaned translated BI Publisher template file is a file that does not have a corresponding base template file. For each translated BI Publisher template file, there must be a base BI Publisher template file with matching keys. You must delete the row for each orphaned translated BI Publisher template file.
		Use your platform's query tool to select against the tables storing translated BI Publisher template file using the fields listed in the report. Perform a SELECT command first to ensure that you are getting the same row count as the report, then use the DELETE command to delete the selected rows.
		The following provides example code for an Oracle database. In this example, *xxx* is equal to the language code to which you are swapping.
		`SELECT * FROM PSXPTMPLTRINFO A WHERE NOT EXISTS (SELECT 1 FROM PSXPTMPLFILEDEF B WHERE A.TMPLDEFN_ID = B.TMPLDEFN_ID AND A.EFFDT = B.EFFDT) AND A.TMPLLANGCD = 'xxx';`
		You must delete the orphan rows from PSFILEDEFN and PSFILEDATA before you delete the orphan rows from PSXPTMPLTRINFO.
		`DELETE FROM PSFILEDEFN WHERE FILEID IN (SELECT XLIFF_FILEID FROM PSXPTMPLTRINFO A WHERE NOT EXISTS (SELECT 1 FROM PSXPTMPLFILEDEF B WHERE A.TMPLDEFN_ID = B.TMPLDEFN_ID AND A.EFFDT = B.EFFDT) AND TMPLLANGCD='xxx');`
		`DELETE FROM PSFILEDATA WHERE FILEID IN (SELECT XLIFF_FILEID FROM PSXPTMPLTRINFO A WHERE NOT EXISTS (SELECT 1 FROM`

Copyright © 1988, 2026, Oracle and/or its affiliates.

Audit	Error Description	Resolution
		PSXPTMPLFILEDEF B WHERE A.TMPLDEFN_ID = B.TMPLDEFN_ID AND A.EFFDT = B.EFFDT) AND TMPLLANGCD='*xxx*'); DELETE FROM PSXPTMPLTRINFO WHERE NOT EXISTS (SELECT 1 FROM PSXPTMPLFILEDEF B WHERE PSXPTMPLTRINFO.TMPLDEFN_ ID = B.TMPLDEFN_ID AND PSXPTMPLTRINFO.EFFDT = B.EFFDT) AND TMPLLANGCD='*xxx*';

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 75 MB log or rollback segment space before running the swap process.

Running the SWAP_BASE_LANGUAGE Data Mover Command

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

```
SWAP_BASE_LANGUAGE target_language;
```

PeopleSoft 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, environmental issues, such as lack of database space, log space, or rollback segment space, causes any errors during this stage of the process. If a failure occurs, note the database-specific error message issued and take the appropriate action according to your database platform documentation.

Copyright © 1988, 2026, Oracle and/or its affiliates.

PeopleSoft Data Mover stops when errors are encountered. 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, rerun the SWAP_BASE_LANGUAGE command. PeopleSoft 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 PeopleSoft Data Mover commands:

```
SET BASE_LANGUAGE target_language;
SWAP_BASE_LANGUAGE recname;
```

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

If you want PeopleSoft Data Mover to not stop when errors are encountered and continue the swap process, set the following:

```
SET IGNORE_ERRORS;
SWAP_BASE_LANGUAGE LANGUAGE_CD;
```

If you chose to ignore errors during the swap, after the swap process is complete carefully review the log file to ensure that no errors occurred. If errors occur, fix the errors and then re-swap each table using the SWAP_BASE_LANGUAGE command in individual table mode for each table that failed.

Managing and Patching Swapped Language Database

After you have upgraded to a new release of PeopleTools, you can apply patches to a swapped database using the usual upgrade tools such as the Change Assistant, Application Designer Project Copy, or Data Mover. These tools will make the transformation between the non-swapped source tables and the swapped target tables. If the patch involves direct SQL statements, the SQL statements should account for swapped and non-swapped environments. If you are writing or running your own custom SQL or querying tables directly, keep in mind that if a table has a related language table associated with it, after swapping the base language you may need to adjust your SQL to work with the correct table.

To apply patches for multi language files, the steps are listed within each upgrade or install guide for your database platform.

See the product documentation for *PeopleSoft 9.2 Application Installation* for your database platform, “Working with Multilingual Databases”.

The following table compares the two patching methods, for applying patches to applications:

Copyright © 1988, 2026, Oracle and/or its affiliates.

Using the Classic method for applying patches	Using the PeopleSoft Upgrade Manager (PUM) for applying patches
Used for Applications 9.1 and earlier.	Used for Applications 9.2 and later.
Change Assistant loads patches and bundles in a predefined upgrade package. Change Assistant will call Data Mover or Project Copy utilities.	PeopleSoft Update Manager generates a custom upgrade package, then uses Change Assistant to call Data Mover or Project Copy utilities.
Translations ship separately from English language, sometimes at a later release date than English. The translations are applied after applying English.	Translations ship in the patch image along with English language, and each language is loaded if the target database has it enabled. More information on PUM is at PeopleSoft Update Manager (PUM) Home Page (Doc ID 1464619.1).

The following table compares the two patching methods, for applying patches to PeopleTools:

Using the Manual method for applying patches	Using the Change Assistant method for applying patches
Used for Tools 8.53 and earlier.	Used for Tools 8.54 and later.
Manually calls Data Mover or Project Copy utilities.	Change Assistant calls Data Mover or Project copy utilities.
Translations ship in the project and data directories, along with English.	Translations ship in the project and data directories, along with English, and each language is loaded if the target database has it enabled.

Also see,

How to Download PeopleSoft Bundles and Fixes from My Oracle Support (Doc ID 1432368.1)

Creating Related Language Tables

PeopleSoft applications provide related language tables for most edit tables that you are likely to translate, so in most cases, you won't have to create your own related language tables. However, if during a customization you create a new table that requires translation, you must create your own related language table.

A one-to-one relationship is recommended between the base record and the related language record. In addition, both base and related language records are recommended to share the same record type. For example, if the base record is a SQL table, then the related language record should be defined as a SQL table.

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

> **Note**
>
> If a field is defined as a Long (DESCRLONG), then you must ensure that the table is in a tablespace dedicated to tables containing LONG/IMAGE columns (for example, PSIMAGE tablespace for PeopleTools-owned tables).

Copyright © 1988, 2026, Oracle and/or its affiliates.

> **Note**
>
> Records with duplicate order keys should not have related language tables associated to them. If the record requires a related language record, you must remove the order keys. Do not create a related language record with a duplicate order key.

To create a related language table:

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

   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.

   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.*

   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.

3. Build the related language table.

   An easy way to build the related language table is to copy the base record definition using the Save As option on the File menu. 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.

   > **Note**
   >
   > 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. For this reason, you should not maintain PeopleCode programs on related language records.
   >
   > It is also recommended that you remove any search key and alternate search key attributes from the related language table, as these do not need to be translated and can hinder performance.

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

   Open the Record Properties dialog box for the base table. On the Use tab, select the related language record in the **Related Language Record** field.

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

Copyright © 1988, 2026, Oracle and/or its affiliates.

5. Save the record.

Creating Related Language Views

This section discusses related language views.

Understanding Related Language Views

Just as records with language-sensitive fields require related language records, views with language-sensitive fields require related language views.

Related language views work the same way as related language records and must adhere to the same rules:

- Any time you create a view over a table that has a related language record, you typically also need a related language view.
- The related language view consists of, at a minimum, 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, the PeopleSoft system 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, such as prepending L or _LANG to the _VW suffix of the view name.

> **Note**
>
> The related language record type must match the base record type—that is, either both are SQL tables or both are SQL views.

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.

> **Note**
>
> In the following examples, the changes to the SELECT statement for the related language view have been highlighted in bold.

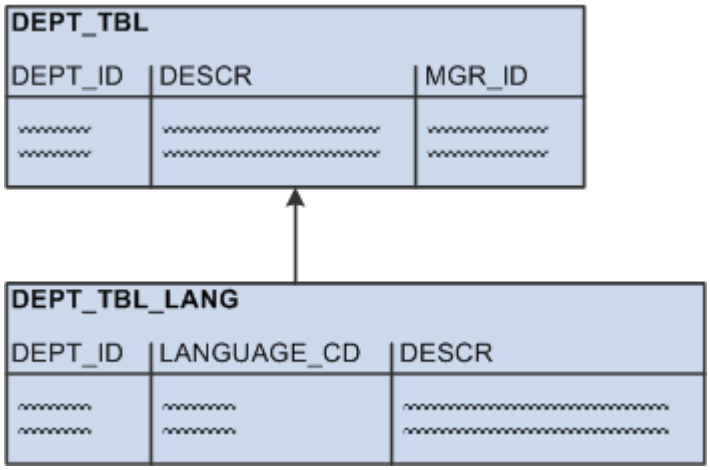
Creating Views for One Base Table and One Related Language Table

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

Copyright © 1988, 2026, Oracle and/or its affiliates.

The following diagram shows DEPT_TBL with one key column (DEPT_ID), one language-sensitive column (DESCR), and one column that is not language sensitive (MGR_ID). DEPT_TBL has a one-to-many relationship with DEPT_TBL_LANG, a table that includes DEPT_ID, LANGUAGE_CD, and DESCR:

This diagram illustrates views of one base table, one related language table



As an example, DEPT_TBL contains the following data:

DEPT_ID	DESCR	MGR_ID
110	Finance	01732
120	Engineering	22056
131	Sales - Belgium	08630
132	Sales - Germany	08630
133	Sales - UK	12972
134	Sales - Japan	28553

The related language table, DEPT_TBL_LANG, contains the following data:

DEPT_ID	LANGUAGE_CD	DESCR
110	FRA	Finances
110	GER	Finanzierung
120	FRA	Technologie
120	GER	Technik
131	FRA	Ventes - Belgique
132	GER	Verkäufe - Deutschland
133	GER	Verkäufe - GB

The following is the SELECT statement for the base view:

```
SELECT A.DEPT_ID, A.DESCR FROM DEPT_TBL A
```


Copyright © 1988, 2026, Oracle and/or its affiliates.

The following is the SELECT statement for the related language view:

```
SELECT B.DEPT_ID, B.DESCR, B.LANGUAGE_CD FROM DEPT_TBL_LANG B
```

Because this base view is 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_CD column.

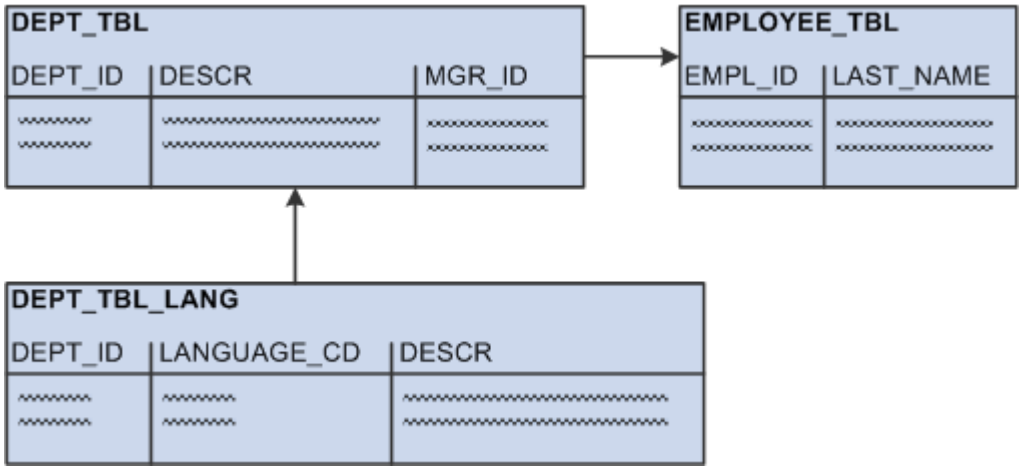
If the base view also selected non-key, non-language-sensitive columns from the base table, those columns would need to be selected from the base table in the related language view because *non-key,* non-language-sensitive fields *don't exist* in related language records.

Creating Views for Two Base Tables and 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.

The following diagram shows the addition of EMPLOYEE_TBL to the previous diagram. DEPT_TBL has a many-to-one relationship with EMPLOYEE_TBL, which has one key column (EMPL_ID) and one non-language-sensitive column (LAST_NAME). The same one-to-many relationship as depicted previously exists between DEPT_TBL and the related language record, DEPT_TBL_LANG:

This diagram illustrates views of two base tables, one related language table



As an example, DEPT_TBL contains the same data as listed previously:

DEPT_ID	DESCR	MGR_ID
110	Finance	01732
120	Engineering	22056
131	Sales - Belgium	08630
132	Sales - Germany	08630
133	Sales - UK	12972

Copyright © 1988, 2026, Oracle and/or its affiliates.

DEPT_ID	DESCR	MGR_ID
134	Sales - Japan	28553

The related language table, DEPT_TBL_LANG, also contains the same data:

DEPT_ID	LANGUAGE_CD	DESCR
110	FRA	Finances
110	GER	Finanzierung
120	FRA	Technologie
120	GER	Technik
131	FRA	Ventes - Belgique
132	GER	Verkäufe - Deutschland
133	GER	Verkäufe - GB

Finally, EMPLOYEE_TBL contains the following data:

EMPL_ID	LAST_NAME
01732	Jones
08630	Gräff
12972	Smythe
17145	De Bruecker
22056	Agarwal
28553	Katsuhiro

The following SELECT statement for the base view selects the department ID, description, and the last name of the manager for each department. *The department description is language-sensitive, but the employee last name is not.*

```
SELECT A.DEPT_ID, A.DESCR, B.LASTNAME
FROM
  DEPT_TBL A,
  EMPLOYEE_TBL B
WHERE A.MANAGER_ID = B.EMPL_ID
```

The following SELECT statement is for the related language view:

```
SELECT A.DEPT_ID, C.DESCR, B.LASTNAME, C.LANGUAGE_CD
FROM
  DEPT_TBL A,
  EMPLOYEE_TBL B,
  DEPT_TBL_LANG C
WHERE A. MGR_ID = B.EMPL_ID
  AND A.DEPT_ID = C.DEPT_ID
```

As an example of a view used in PeopleTools, the following SELECT statement builds the PeopleTools view (PTLT_FEATFLTRVW). The base view selects one key column (PTLT_FEATURE_CODE) and one language-sensitive column (PTLT_FEATURE) from the

Copyright © 1988, 2026, Oracle and/or its affiliates.

base table (PTLT_FEATURE) while joining additional data (PTLT_PROJ_NAME) from another base table (PTLT_PROJ_DEFN):

```
SELECT DISTINCT
  A.PTLT_PROJ_NAME,
  D.PTLT_FEATURE_CODE,
  D.PTLT_FEATURE
FROM
  PTLT_PROJ_DEFN A,
  PTLT_PROJ_TASK B,
  PTLT_ASSGN_TASK C,
  PTLT_FEATURE D
WHERE A.PTLT_PROJ_NAME = B.PTLT_PROJ_NAME
  AND B.PTLT_TASK_CODE = C.PTLT_TASK_CODE
  AND C.PTLT_FEATURE_CODE = D.PTLT_FEATURE_CODE
```

The following SELECT statement is for the related language view (PTLT_FEATFL_LVW). In this view, the key field, language-sensitive data, and the language code are selected from the related language table (PTLT_FEAT_LANG):

```
SELECT DISTINCT
  A.PTLT_PROJ_NAME,
  D.PTLT_FEATURE_CODE,
  D.PTLT_FEATURE,
  D.LANGUAGE_CD
FROM
  PTLT_PROJ_DEFN A,
  PTLT_PROJ_TASK B,
  PTLT_ASSGN_TASK C,PTLT_FEAT_LANG D
WHERE A.PTLT_PROJ_NAME = B.PTLT_PROJ_NAME
  AND B.PTLT_TASK_CODE = C.PTLT_TASK_CODE
  AND C.PTLT_FEATURE_CODE = D.PTLT_FEATURE_CODE
```

This related language view differs from the base view in only two ways:

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

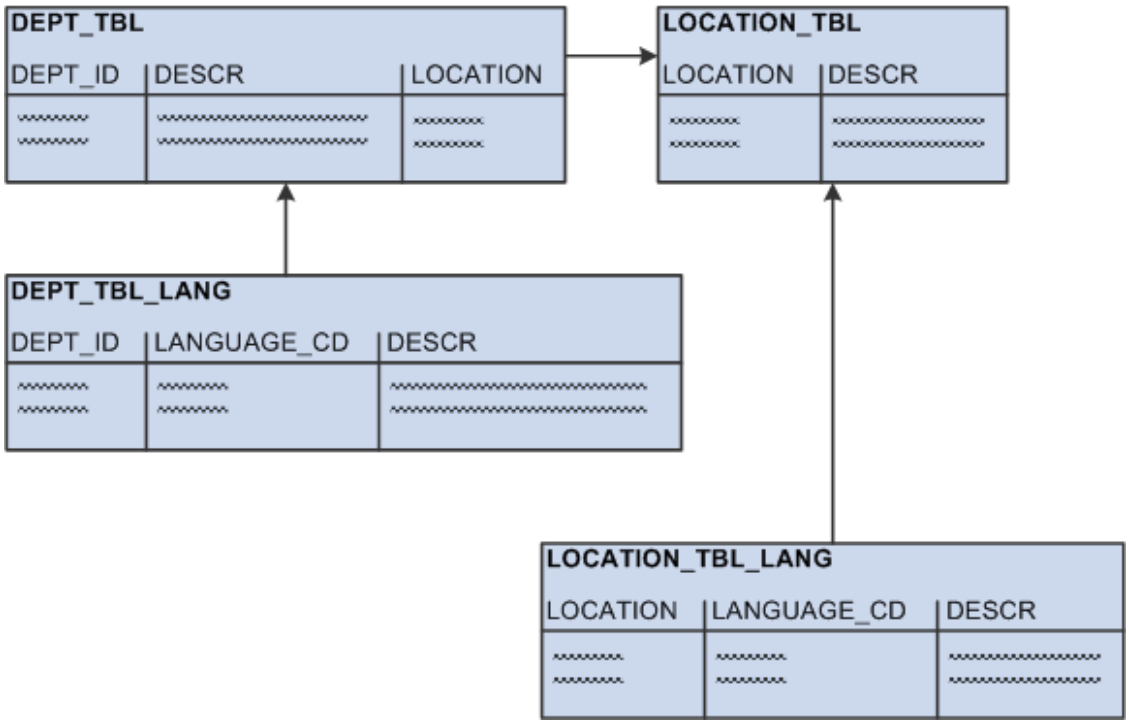
Creating Views for Two Base Tables and Two Related Language Tables

When you join two tables that both have related language records, the SELECT statement becomes complex.

The following diagram shows a new many-to-one relationship of DEPT_TBL to LOCATION_TBL. Unlike the previous examples, DEPT_TBL includes a LOCATION column rather than a MGR_ID column. LOCATION_TBL includes a key column (LOCATION) and a language-sensitive column (DESCR). The same relationship as depicted previously exists between the related language record, DEPT_TBL_LANG, and DEPT_TBL. In addition, there is a one-to-many relationship between LOCATION_TBL and its related language record, LOCATION_TBL_LANG, which includes LOCATION, LANGUAGE_CD, and DESCR:

This diagram illustrates views of two base tables, two related language tables

Copyright © 1988, 2026, Oracle and/or its affiliates.



Unlike the previous examples, DEPT_TBL includes a LOCATION column with data rather than a MGR_ID column:

DEPT_ID	DESCR	LOCATION
110	Finance	RWS
120	Engineering	RWS
131	Sales - Belgium	BRU
132	Sales - Germany	MUN
133	Sales - UK	LON
134	Sales - Japan	OSA

The related language table, DEPT_TBL_LANG, contains the same data as previously:

DEPT_ID	LANGUAGE_CD	DESCR
110	FRA	Finances
110	GER	Finanzierung
120	FRA	Technologie
120	GER	Technik
131	FRA	Ventes - Belgique
132	GER	Verkäufe - Deutschland
133	GER	Verkäufe - GB

LOCATION_TBL contains location codes and descriptions:

Copyright © 1988, 2026, Oracle and/or its affiliates.

LOCATION	DESCR
RWS	Redwood Shores
BRU	Brussels
MUN	Munich
LON	London
OSA	Osaka

Finally, LOCATION_TBL_LANG contains translations of the language-sensitive field, DESCR:

LOCATION	LANGUAGE_CD	DESCR
BRU	FRA	Bruxelles
BRU	GER	Brüssel
LON	FRA	Londres
MUN	GER	München

In this example, the base view joins DEPT_TBL with LOCATION_TBL to retrieve the location of the department. The following SELECT statement is for the base view:

```
SELECT A.DEPT_ID, A.DESCR, B.LOCATION, B.DESCR
FROM
  DEPT_TBL A,
  LOCATION_TBL B
WHERE A.LOCATION = B.LOCATION
```

However, in this example, creating the related language view is difficult because 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 LOCATION_TBL_LANG.

The related language view needs to take into account the following scenarios:

- A translated department may reference a translated location—for example, the German for department 132 (Verkäufe – Deutschland) references the German for location MUN (München).
- A translated department may reference an untranslated location—for example, the French for department 120 (Technologie) does not have a translated location and therefore references the untranslated location for RWS (Redwood Shores).

  In this scenario, there is a row in DEPT_TBL_LANG, but there is no row in LOCATION_TBL_LANG. If you were to join DEPT_TBL_LANG with LOCATION_TBL_LANG in the view, no translation would be retrieved by the related language view because the location doesn't have a translation.
- A translated location may reference an untranslated department —for example, the German for location BRU (Brüssel) does not have a translated department and therefore references the untranslated department for 131 (Sales - Belgium).

  In this scenario, there is a row in LOCATION_TBL_LANG, but there is no row in DEPT_TBL_LANG. If you were to join LOCATION_TBL_LANG with DEPT_TBL_LANG in the view, no translation would be retrieved by the related language view because the department doesn't have a translation.

Copyright © 1988, 2026, Oracle and/or its affiliates.

- An untranslated department may reference an untranslated location—for example, department 134 (Sales - Japan) and the location OSA (Osaka) are not translated into either French or German. The related language view should return no rows.

To address these scenarios, the SQL statement for the related language view must include logic for the first three scenarios—those where translations exist. Following is the SELECT statement for the related language view:

```
SELECT C.DEPT_ID, C.DESCR, B.LOCATION, B.DESCR, C.LANGUAGE_CD
FROM
  DEPT_TBL A,
  LOCATION_TBL_LANG B,
  DEPT_TBL_LANG C
WHERE A.DEPT_ID = C.DEPT_ID
  AND A.LOCATION = B.LOCATION
  AND C.LANGUAGE_CD = B.LANGUAGE_CD
UNION
SELECT F.DEPT_ID, F.DESCR, E.LOCATION, E.DESCR, F.LANGUAGE_CD
FROM
  DEPT_TBL D,
  LOCATION_TBL E,
  DEPT_TBL_LANG F
WHERE
  F.DEPT_ID = D.DEPT_ID
  AND D.LOCATION = E.LOCATION
  AND NOT EXISTS (SELECT 'X'
    FROM LOCATION_TBL_LANG H
    WHERE H.LOCATION = E.LOCATION
    AND H.LANGUAGE_CD = F.LANGUAGE_CD)
UNION
SELECT I.DEPT_ID, I.DESCR, J.LOCATION, J.DESCR, J.LANGUAGE_CD
FROM
  DEPT_TBL I,
  LOCATION_TBL_LANG J
WHERE
  I.LOCATION = J.LOCATION
  AND NOT EXISTS (SELECT 'X'
    FROM DEPT_TBL_LANG K
    WHERE K.DEPT_ID = I.DEPT_ID
```

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 previously described. Let’s examine this view, statement by statement.

SELECT Statement One

The following SELECT statement addresses scenario one. It retrieves the rows for which translations for both the department and the location exist.

```
SELECT C.DEPT_ID, C.DESCR, B.LOCATION, B.DESCR, C.LANGUAGE_CD
FROM
  DEPT_TBL A,
  LOCATION_TBL_LANG B,
  DEPT_TBL_LANG C
WHERE A.DEPT_ID = C.DEPT_ID
```


Copyright © 1988, 2026, Oracle and/or its affiliates.

```
  AND A.LOCATION = B.LOCATION
  AND C.LANGUAGE_CD = B.LANGUAGE_CD
```

SELECT Statement Two

The following SELECT statement addresses scenario two. It retrieves the rows for which the department translation exists, but the location translation does not exist.

The sub-SELECT statement is required in order to prevent this statement from retrieving records that are returned by statement one.

```
SELECT F.DEPT_ID, F.DESCR, E.LOCATION, E.DESCR, F.LANGUAGE_CD
FROM
  DEPT_TBL D,
  LOCATION_TBL E,
  DEPT_TBL_LANG F
WHERE
  F.DEPT_ID = D.DEPT_ID
  AND D.LOCATION = E.LOCATION
  AND NOT EXISTS (SELECT 'X'
    FROM LOCATION_TBL_LANG H
    WHERE H.LOCATION = E.LOCATION
    AND H.LANGUAGE_CD = F.LANGUAGE_CD)
```

SELECT Statement Three

The following SELECT statement addresses scenario three. It retrieves the rows for which translations exist for the location but not the department. Again, the sub-SELECT statement is needed to avoid returning rows that are returned by statement one.

```
SELECT I.DEPT_ID, I.DESCR, J.LOCATION, J.DESCR, J.LANGUAGE_CD
FROM
  DEPT_TBL I,
  LOCATION_TBL_LANG J
WHERE
  I.LOCATION = J.LOCATION
  AND NOT EXISTS (SELECT 'X'
    FROM DEPT_TBL_LANG K
    WHERE K.DEPT_ID = I.DEPT_ID
    AND K.LANGUAGE_CD = J.LANGUAGE_CD)
```


Copyright © 1988, 2026, Oracle and/or its affiliates.

12

Working With Language-Sensitive Application Data

Understanding Data Editing in Related Language Tables and Base Tables

When a user changes data on a language-sensitive page, the system responds by changing the contents of the base table, the related language table, or both, depending on the language of the user and which field was modified. The following table explains at the database table level the actions that are performed when you work with language-sensitive tables:

Action	Sign-in 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 sign-in language.
Edit a language-sensitive field.	Base	Changes a field in the base table without 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 sign-in language. Doesn't affect 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 sign-in language, to the related language table. This could introduce non-base language descriptions into the base language table.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Action	Sign-in Language	Result
Insert a row (effective-dated).	Base	Adds a new effective-dated row to both the base table and related language 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.

Related Topics

- Understanding Application Definition Translation
  If you are running PeopleSoft applications with a multiple language user interface, you should translate any customizations you make to the product using PeopleTools into each of the languages that you are using. PeopleTools provides a full suite of tools to help you translate the application user interface; these are the same tools that PeopleSoft uses internally to provide the translations we ship on the global multi-language CDs.
- Using 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.
- Understanding Related Language Tables
  This section discusses related language tables.

Editing Data in Multiple Languages

This section provides an overview of multi-language entry and discusses how to enter data in multiple languages.

Understanding Multi-Language Entry

Use one of these two methods to maintain translations of application data:

- Log into the system in a language other than the database's base language and update a field that is language-sensitive.
- Use the multi-language entry option.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Using the multi-language entry option provides significant benefits over simply changing the session language and overwriting existing base language descriptions with their translations. It allows you to:

- Enter or edit data in multiple languages during the same session without changing the sign-in language.
- 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.
- Easily recognize which fields on a page are language-sensitive and which are language-neutral.

Multi-language entry makes it 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.

Multi-language entry is active if all of the following conditions are true:

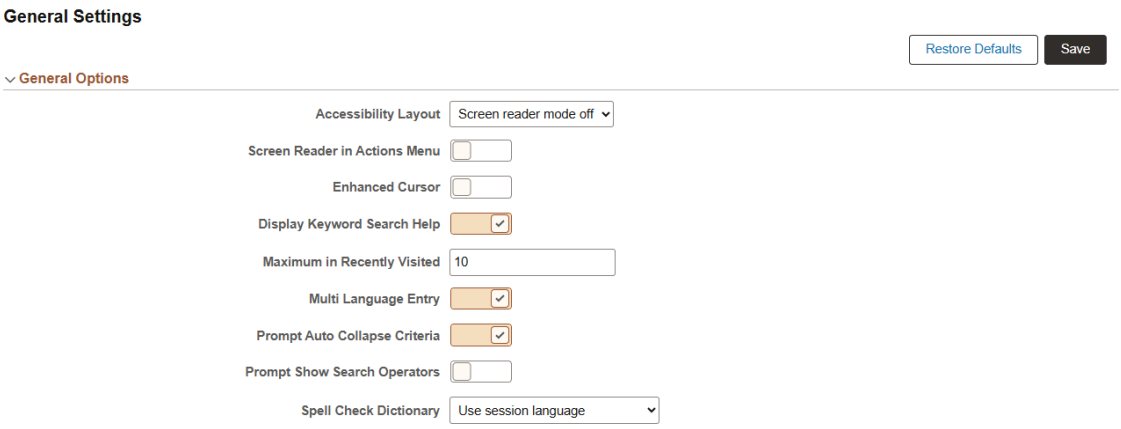
- The multi-language entry option is enabled for the user operator.
- The current page contains a record that has a related language record.
- The component is not a fluid component.

Enabling Multi-Language Entry

Multi-language entry makes it more intuitive for a multilingual user to maintain data in several languages before saving a page.

The following example shows the Option Category: General Options page that you use to enable multi-language entry.

This example illustrates the fields and controls on the Personalizations: General Options page with multi-language entry enabled.



To enable multi-language entry:

1. Select **My Personalizations** on the navigation menu from your homepage.

   The Personalizations page appears.

2. Click the **Personalize Options** button on the General Options line.

   The Option Category General Options page appears.

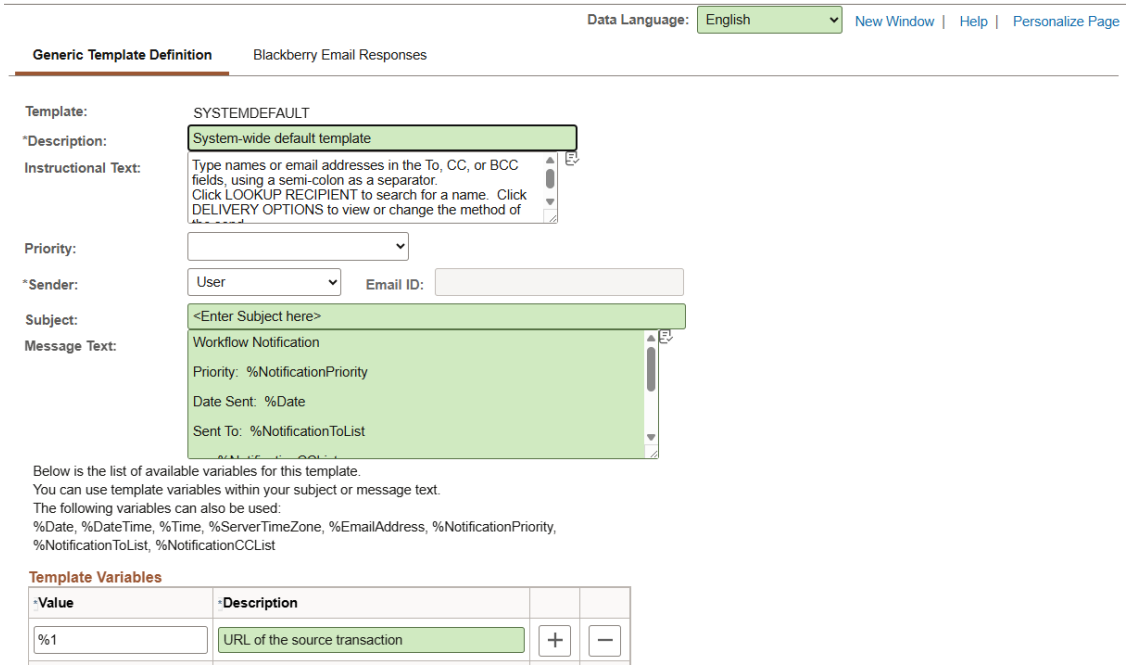
Copyright © 1988, 2026, Oracle and/or its affiliates.

3. Change the value for the **Multi Language Entry** option to *Yes*.
4. Click **OK**.

Entering Data in Multiple Languages

After you have enabled multi-language entry in **My Personalizations,** the display of pages changes in two important ways:

This example illustrates the fields and controls on a page with multi-language entry enabled.



1. A **Data Language** field appears at the top of each page.

   The **Data Language** field indicates in which language the language-sensitive fields on the current page are being maintained. You can switch the display of the page between different data languages and maintain translations in each by changing the selected language in the drop-down menu.

2. Any fields that are language-sensitive (that is, they are translatable as they exist both in the base language and related language records and are not keys) appear with a pale green shaded background.

   This shading makes it easy to determine if the changes you make to the data will take effect in the current language only, or take effect across all languages in the database.

   For example, in the previous graphic, the **Subject** field is language-sensitive. A different translation of the subject can be entered for each language enabled in the database, but the **Email ID** field is not, so any changes to the **Email ID** will be visible to all system users.

To translate this data into another language, simply pick a new language in the **Data Language** field. If translations exist for the current row of data in the new language, those translations will appear. If not, the base language values will still appear, and you can translate these into to the new language selected.

Copyright © 1988, 2026, Oracle and/or its affiliates.

> **Note**
>
> Changing the data language of a page using multi-language entry only changes the language of application data; it does not change the language of the user interface of the page such as field labels and help text. The language of the user interface is controlled only by your sign-in language preference.

As changing the data language does not change the labels and other user interface elements of a page, it is also useful to determine terminology in different languages. For example, if Department descriptions in your Department table have been translated into several different languages, you can view the description of a department in each language simply by navigating to the Department table page and changing the data language. Because this only changes the language of the application data and not the user interface, you can “look up” terminology in other languages without needing to be able to navigate or use a page that has been translated into that language.

To enter data in multiple languages:

1. Enable the multi-language entry option.

   See Enabling Multi-Language Entry.

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

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

3. In the **Data Languages** field, select a target language.

   The language that appears on the page remains the same; only the language-sensitive data is reloaded in the selected language.

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

5. When finished editing the multilingual fields, click the Save button.

The system updates the base and related-language tables for the page.

Copyright © 1988, 2026, Oracle and/or its affiliates.

13

Implementing Bidirectional Language Support

Understanding Bidirectional Language Support

The PeopleSoft system provides bidirectional support that enables you to use right-to-left oriented languages, such as Arabic and Hebrew.

This section discusses features, translations and platforms.

> **Note**
>
> Bidirectional languages such as Arabic and Hebrew are not certified with the PeopleSoft Fluid User Interface.

Features

These features are included for bidirectional support in the PeopleSoft Pure Internet Architecture:

- Right-aligned menu navigation pane.
- Right-aligned scroll bars.
- Right-aligned field labels.
- Right-to-left data entry.
- Right-aligned display images.
- Hijri (Umm al-Qura) calendar support.
- Ability to designate images to display horizontally flipped.

Translations Provided

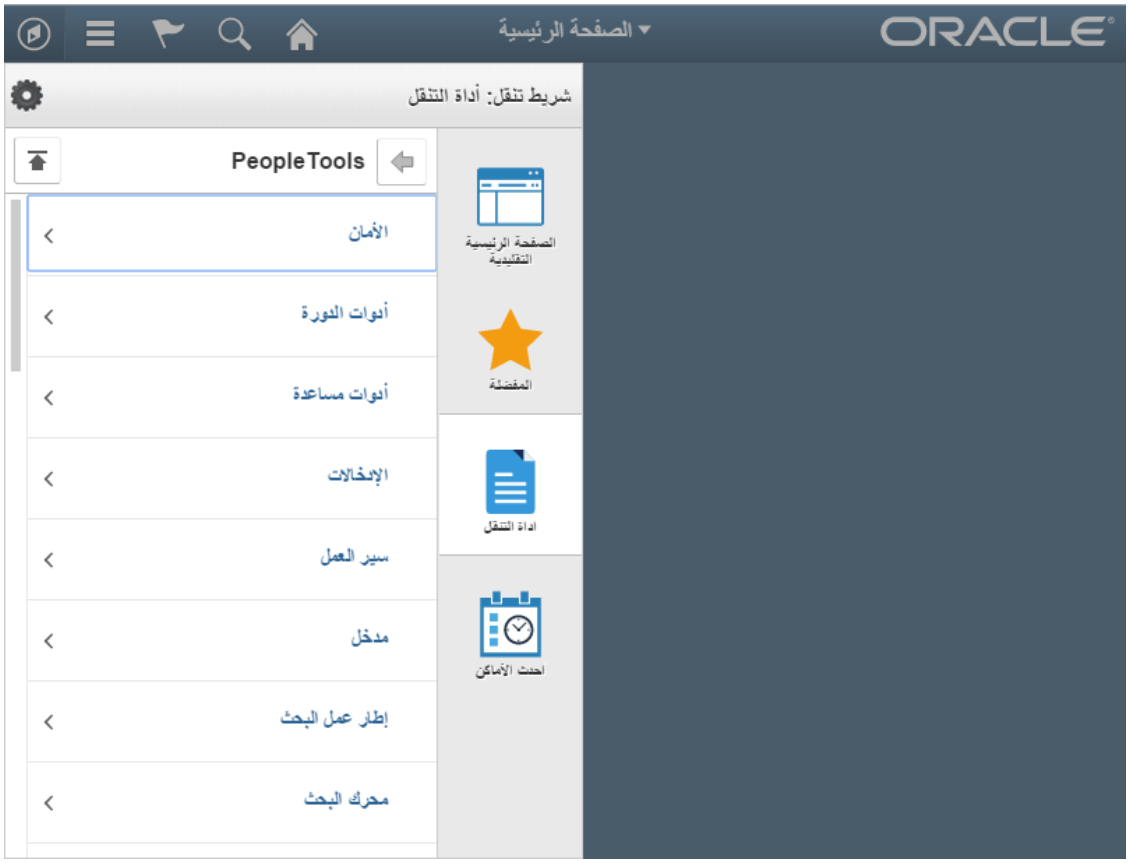
PeopleTools is delivered with Arabic and is supported on the following Unicode databases: Oracle, Microsoft SQL Server, and DB2 z/OS.

Navigating in Bidirectional PeopleSoft Pure Internet Architecture Pages

The bidirectional PeopleSoft Pure Internet Architecture pages are enabled when you select Arabic as the sign-in language. After signing in to the PeopleSoft Pure Internet Architecture, the bidirectional PeopleSoft fluid homepage appears in Arabic. Once signed on as an Arabic or other bidirectional user, text, fields, scroll bars, and images are right-aligned.

This example illustrates bidirectional PeopleSoft fluid homepage.

Copyright © 1988, 2026, Oracle and/or its affiliates.



You can also access the classic user interface from the NavBar button.

Customizing PeopleSoft Pure Internet Architecture Pages for Bidirectionality

This section discusses how to customize the PeopleSoft Pure Internet Architecture pages for bidirectionality.

> **Note**
>
> You can set bidirectionality explicitly for an object or inherit it from parent objects.

Working with HTML Areas and Style Sheets

This table lists and describes the system and meta-HTML variables that are used to develop bidirectional PeopleSoft Pure Internet Architecture pages:

Directionality Variables	Description
%AlignStart	Use this system or meta-HTML variable instead of hard-coded `left` to align page elements.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Directionality Variables	Description
%AlignEnd	Use this system or meta-HTML variable instead of hard-coded `right` to align text.
%AlignAmount	Use this system or meta-HTML variable instead of hard-coded `right` for numeric amounts
%Direction	Use this system or meta-HTML variable instead of hard-coded `ltr` or `rtl` to set the directionality of elements, such as the `<html>` tag of an HTML document.

Anywhere you normally use `left`, `right`, `ltr`, or `rtl`, use the appropriate directionality variable. For instance, instead of

```
<td align='right'>
```

use

```
<td align='%AlignEnd'>
```

This also applies to `left` and `right` keywords within HTML substrings. For instance:

```
%AlignStartmargin=10
```

> **Note**
>
> HTML elements inherit the directionality of the parent element, so setting `dir='%Direction'` on the HTML tag will take effect for all the elements within the HTML page, unless overridden. If no directionality is set on the HTML element, the default value is `dir='ltr'`.

Working with Images

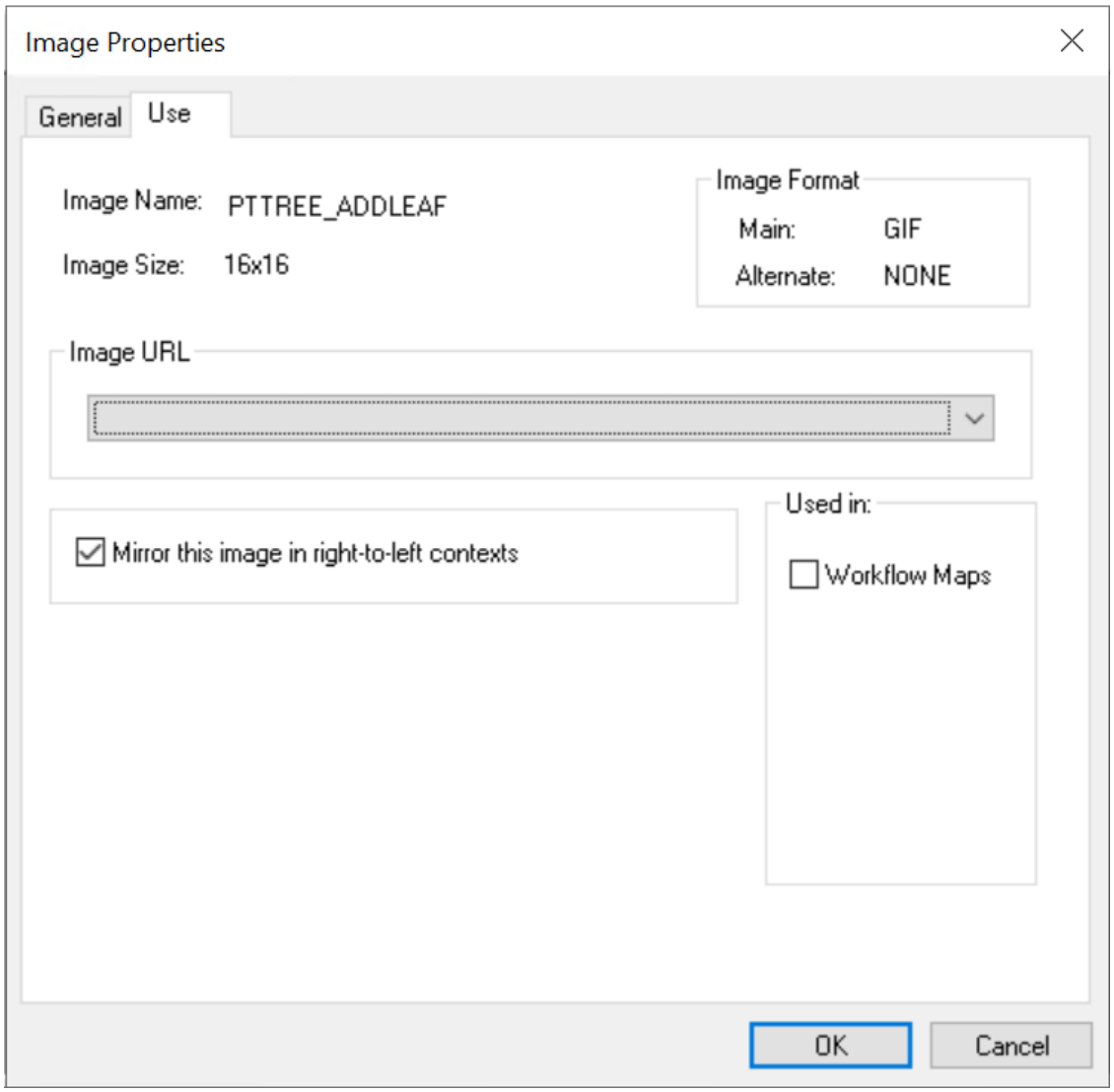
When adding images to the image catalog, you must identify the images you want to flip horizontally when loaded.

The Use tab on the Image Properties dialog box features a Mirror this image in right-to-left contexts check box that enables you to specify that an image be flipped horizontally when loaded onto a page in a right-to-left context.

> **Note**
>
> The check box can only be changed when the user is logged into Application Designer as a base language user. If an image is marked to be mirrored, it cannot be translatable. Any translations for an image marked to be mirrored will be discarded.

This example illustrates the fields and controls on the Image Properties - Use tab.

Copyright © 1988, 2026, Oracle and/or its affiliates.



As a general rule, set this option for any image that does not contain text and has a directional meaning, such as arrow icons that indicate next and previous actions.

Working with Bidirectional Languages and Fluid

If you follow the guidelines for HTML areas and style sheets to create pages, the pages will also render correctly for bidirectional languages. See Customizing PeopleSoft Pure Internet Architecture Pages for Bidirectionality.

But for few cases in the FreeForm Style Sheet used for Fluid pages you need to follow some special guidelines related to margin, padding, border, translate, skew, rotate, matrix, radius, shadow, and clip.

These can be supported by using selectors beginning with `:root.psc_dir-rtl`.

Copyright © 1988, 2026, Oracle and/or its affiliates.

For example, to set the box-shadow in the CSS selectors `.ps-button:not([disabled]):active and .ps-button:not([disabled]):hover:active`, the *box-shadow:inset* parameters should have a different order between *ltr* and *rtl* sessions:

```
:root.psc_dir-rtl .ps-button:not([disabled]):active,
:root.psc_dir-rtl .ps-button:not([disabled]):hover:active
{
 %BP(box-shadow:inset -1px 2px 2px #fff); /* BYPASSRTL scan */
}
```

You can automate some replacement patterns, as shown by the Perl regular expressions below. This is an example from Perl script which you can adapt to check and transform a set of files (Freeform style sheets in Application Designer, exported to CSS files), to have *rtl* support. In this script example, **BYPASSRTL** is a keyword used to prevent certain cases of automatic processing from occurring twice.

```
# cssrtl.pl

print ("CssRtl 1.9\n");

$workdir = $ARGV[0];$cssdir = "$workdir\\cssrtl\\css";$warndir = "$workdir\
\cssrtl\\warn";$rtldir = "$workdir\\cssrtl\\rtl";
opendir(DH, $cssdir);my @files = grep {/\.css$/} readdir DH;closedir(DH);
if (! -d $warndir) {     mkdir $warndir;}
if (! -d $rtldir) {     mkdir $rtldir;}
die "no files found in $cssdir\n" if (! @files);
foreach my $file (@files){       print "$file\n";       open(INFILE,
" $cssdir\\$file");        open(OUTFILE, "> $rtldir\\$file");
open(WARNFILE, "> $warndir\\$file");
$line_cnt = 0;
while(<INFILE>)  {     $line_cnt ++;     if(!/BYPASSRTL/ && /left|right/) { #
only process if comment not present to BYPASSRTL processing and contains left/
right
            #Pre-process Exclusion
               #Handle CSS Selectors
               s/\.([^\.\,\s\{]*)left/.\1%%L%%/g;
            s/\.([^\.\,\s\{]*)right/.\1%%R%%/g;
            #Handle image names
            s/(%image\s*\(\s*[^\(\)\s]*)left/\1%%L%%/g;
            s/(%image\s*\(\s*[^\(\)\s]*)right/\1%%R%%/g;
            #Handle quoted strings
             s/("[^.\"]*?)left([^.\"]*?")/\1%%L%%\2/g;
            s/("[^.\"]*?)right([^.\"]*?")/\1%%R%%\2/g;
            s/('[^.\']*?)left([^.\']*?')/\1%%L%%\2/g;
            s/('[^.\']*?)right([^.\']*?')/\1%%R%%\2/g;

            # Do replacement
             s/\bleft\b/%AlignStart/g;
             s/\bright\b/%AlignEnd/g;

         # Revert exclusion
              s/%%L%%/left/g;
              s/%%R%%/right/g;
 }
if (!/BYPASSRTL/ && /(margin|padding|border-color|border-width|border-style)
```


Copyright © 1988, 2026, Oracle and/or its affiliates.

```
\s*:\s*([^\s;]+)\s+([^\s;]+)\s+([^\s;]+)\s+([^\s;]+)\s*;/) {        if ($3
ne $5) {        if ($5 ne '') {
       print WARNFILE "line ($line_cnt): $_";
       print WARNFILE "RTL: add BYPASSRTL at the end of line $line_cnt, and
append the following two lines\n";       my($res,$endval,$startval) =
($1,$3,$5);      my $res1, $res2;       if ($res =~ /border/)
{      ($res1= $res) =~ s/border-(style|width|color)/border-%AlignStart-$1/
g;      ($res2= $res) =~ s/border-(style|width|color)/border-%AlignEnd-$1/
g;    }else {              $res1 ="$res-%AlignStart";
               $res2 ="$res-%AlignEnd";       }      print WARNFILE
"$res1: $startval;\n";     print WARNFILE
"$res2: $endval;\n\n";      }    } } if(!/BYPASSRTL/ && (/translate[^YZ\(]*\
(/ || /skew[^YZ\(]*\(/ || /rotate[^\(]*\(/ || /matrix[^\(]*\(/) ) {   print
WARNFILE "line ($line_cnt): $_";  print WARNFILE "RTL: Add :root.psc_dir-rtl
<your css selector here> { } and transform inverting the sign of the X
direction parameter\n\n"; } if(!/BYPASSRTL/ && (/radius\s*:\s*([^\s;]+)\s+
([^\s;]+)\s+([^\s;]+)\s+([^\s;]+)\s*;/) ) { if ($2 ne $4)
{      my($p1,$p2,$p3,$p4) = ($1,$2,$3,$4);      $p4 =~ s/\)//g;      my
($s1,$s2) = ("$p1 $p2 $p3 $p4","$p2 $p1 $p4 $p3");   if ($s1 ne $s2)
{            print WARNFILE "line ($line_cnt): $_";            print WARNFILE
"RTL: Add :root.psc_dir-rtl <your css selector here> { } and move radius
inner numbers to outer position and vice versa eg $s1
becomes $s2\n\n";  } } } if(!/BYPASSRTL/ && (/(shadow\s*|
clip[^:]*):\s*(\S+)/) ) {  if ($2 !~ /^none/) {           print WARNFILE
"line ($line_cnt): $_";         print WARNFILE "RTL: Add :root.psc_dir-rtl
<your css selector here> { } and adjust the horizontal positioning as
appropriate\n\n";  } }  print OUTFILE
"$_"; }  close(INFILE); close(OUTFILE); close(WARNFILE);}
```

Related Topics

- Fluid User Interface Developer’s Guide

Copyright © 1988, 2026, Oracle and/or its affiliates.

14

Modifying Terminology

Understanding Terminology Management

The PeopleSoft system provides two sets of translation and terminology modification tools:

1. Translation tools for translating application objects into languages that are not supplied by the PeopleSoft system or for translating your customizations.
2. A Terminology Management tool that enables you to modify installed, PeopleSoft-delivered translations to suit local needs.

This topic discusses the Terminology Management tool. The other chapters in this part discuss the tools that are provided for translating customizations or translating objects into new languages.

See Understanding the Addition of New Language Definitions.

This section discusses terminology:

- Management.
- Searches.
- Replacement.
- Replacement undo process.
- Search statuses.

Terminology Management Overview

The PeopleTools Terminology Management tools enable you to streamline certain user interface language changes through the 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, if your Mexican users require that their Spanish user interface reflect terms specific to Mexico, you can use the Terminology Management tool to adapt the PeopleSoft-delivered Spanish translations to substitute Mexican Spanish terms where appropriate.

You can also use this tool to change PeopleSoft-provided user interface terminology to better suit your organization's internal terminology or corporate vocabulary, even if English is your only user interface language.

The Terminology Management tool only adapts an existing installed language—it cannot create a new language based on the changes you define. If you want to preserve the existing translation and use your modifications to create a new PeopleSoft language, you must first define the new language and copy the existing translations to your new language.

Searching and replacing terms is a multistep process, with several opportunities to back out the changes. The process breaks down into the following three phases, each of which is discussed in this section:

1. Search

Copyright © 1988, 2026, Oracle and/or its affiliates.

2. Replace
3. Undo replacement

Each phase consists of three separate steps: setting up the process, running the process, and reviewing the results. However, you don't have to use all three processes all the time. You can use the Search process without ever invoking the Replace or Undo processes.

Terminology Searches

The first phase in terminology management is the Search phase. This table describes the three steps of the Search phase:

Step	Description
Setting up the Search.	On the Define Translations Search page, specify which languages to search, which database objects to search, and the term for which to search. 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.	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, if you are considering replacing terminology and want to determine the potential exposure of such a change, disable the **Defining a Search with Replacement** option. This ensures that there is no risk of accidental text replacement. If Replace processing is enabled, review the results on the Review Translations Replace page. If you don't want to replace the text, you can manually mark the process as complete. If Replace processing is disabled, review the results on the Translation Search Only Result page.

Terminology Replacement

The second phase in terminology management is the Replace phase. This table describes the three steps of this phase:

Step	Description
Setting up the Replace.	If your original search criteria specified replacement text, you will see both the original text and the replacement text on the Translations Search Replace pages. At this point, nothing has actually been replaced. Accept, reject, or modify the replacement text for each search result.
Processing the Replace.	Using PeopleSoft Process Scheduler, run the Replace process.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Step	Description
Reviewing the Replace.	Go to the Review Translations Replace 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. At this point, the replacement has happened, but you can still back out the changes by continuing to the undo step. If you don't want to undo any of the replacements, manually mark the process as complete.

Terminology Replacement Undo Process

The third phase in terminology management is the optional Undo phase. This table describes the three steps of this phase:

Step	Description
Setting up the Undo.	As you review your replace results on the Review Translations Replace page, specify whether to undo any of the replacements.
Processing the Undo.	Using PeopleSoft Process Scheduler, run the Undo process.
Reviewing the Undo.	Go to the Inquiry page that displays all search results to see the final results of all your searching, replacing, and undoing. This page displays all results from the original search, regardless of whether you replaced the term or used undo for any replacement. The Search and Replace is final and the 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.

Terminology Search Statuses

As you proceed through the Search and Replace process, you will 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;* running the Undo process updates the status to *Undone.*

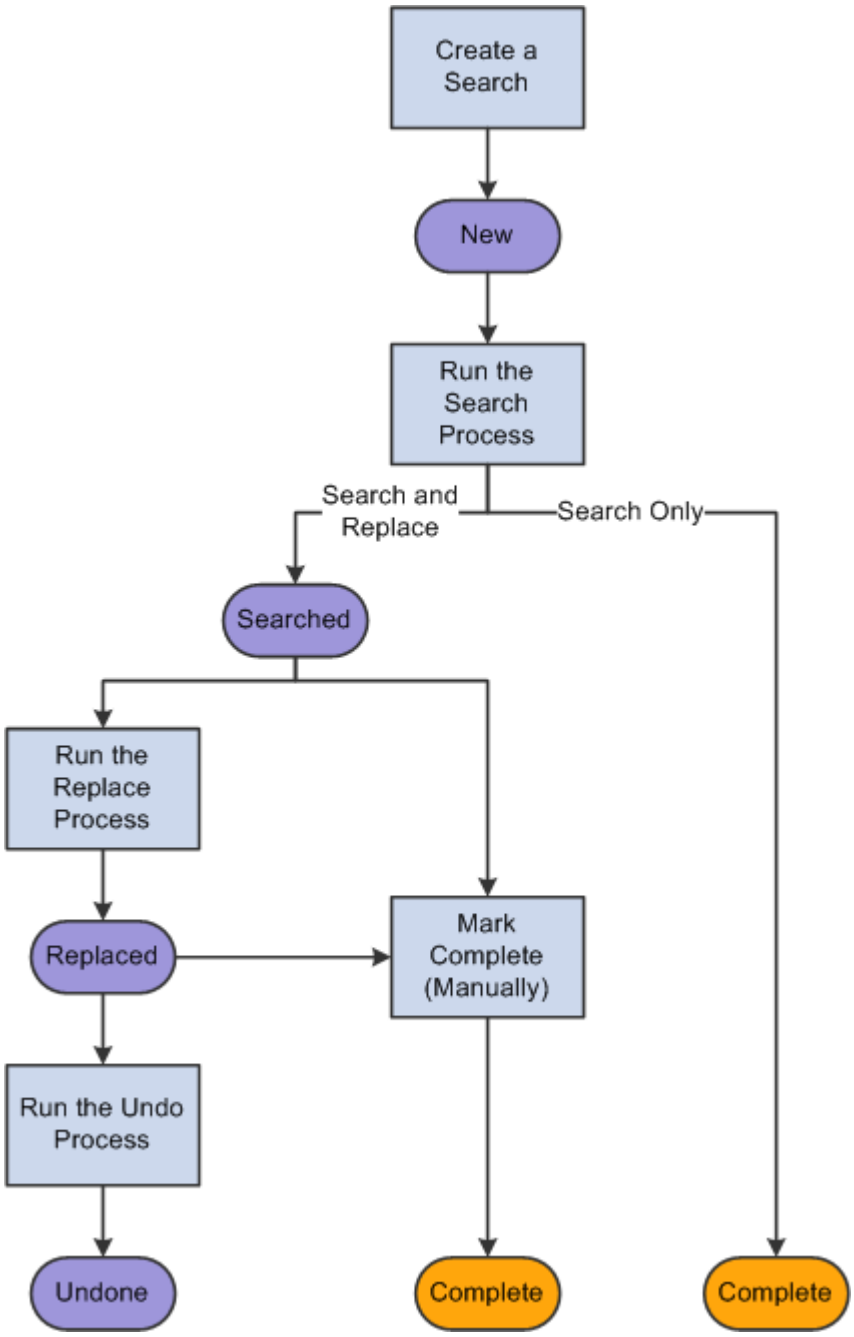
You can manually mark the status *Complete* after searching or replacing. Once the status is *Complete,* you cannot change your mind. If you decide to make further terminology changes,

Copyright © 1988, 2026, Oracle and/or its affiliates.

you must recreate the search criteria using the **Copy Search Criteria** feature on the Search Criteria Inquiry page.

The following diagram shows the relationship between these terminology search processes and statuses:

This diagram illustrates the relationship between terminology search processes and statuses



Granting Terminology Search Privileges

Use the Translations User Options page to grant search and view access to your users

Copyright © 1988, 2026, Oracle and/or its affiliates.

To set user search options:

1. Select **PeopleTools**, and then **Translations**, and then **Modify Terminology**, and then **Translations User 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.
3. Set user search permissions.

Field or Control	Description
Search Only	Select this option 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.
Search, Replace and Undo	Select this option to grant the user full access to all aspects of the PeopleSoft terminology management functionality.

4. Choose the searches that 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 access to Search, Replace, and Undo can be granted access to other users' activity.
5. Click the Save button.

Searching Terminology

This section discusses working with the search process.

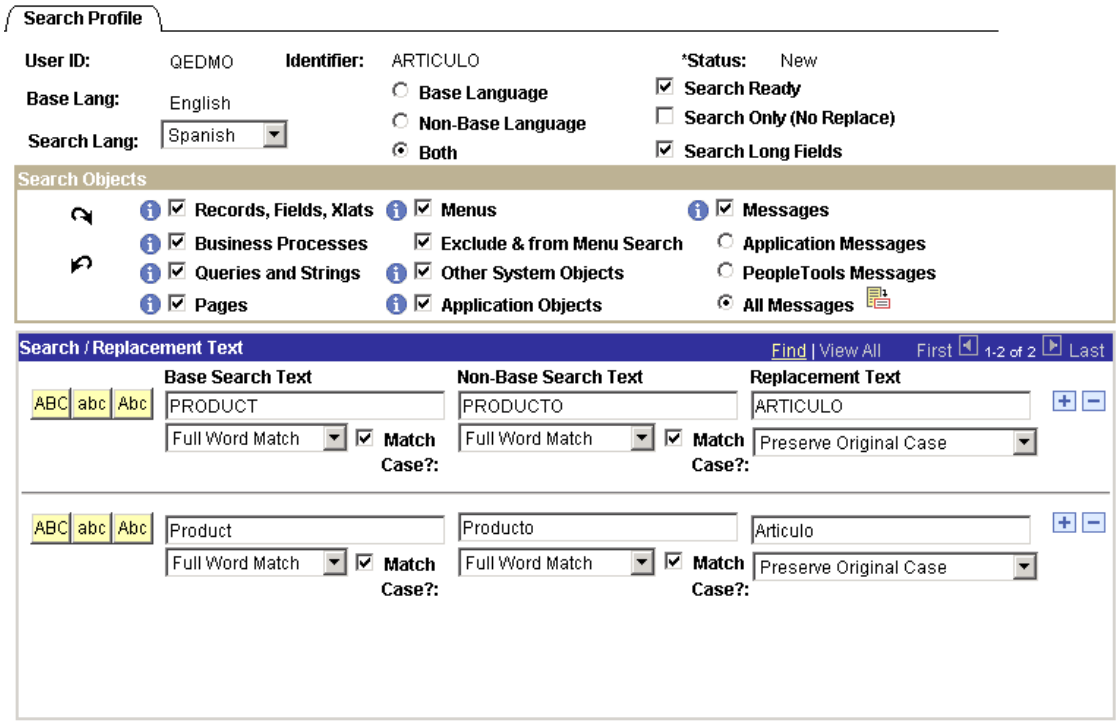
Defining Search Criteria

The first step in a Search and Replace process is to define your search criteria. To define your search, select **PeopleTools**, and then **Translations**, and then **Modify Terminology**, and then **Define Translations Search**.

> **Note**
>
> 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.

Copyright © 1988, 2026, Oracle and/or its affiliates.

This example illustrates the fields and controls on the Define Translations Search - Search Profile page.



> **Note**
>
> The Search Profile page displays searches where the status is *New* only. 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.

This section discusses how to:

- Enable Search processing.
- Choose which languages to search.
- Disable the Replace process.
- Define the search scope.
- Define search text.

Enabling Search Processing

To enable search processing, select the **Search Ready** box.

Choosing Languages to Search

To choose languages to search:

1. Select a search language in the **Search Lang** (search language) drop-down list box.

   The **Base Lang** (base language) display-only field displays the base language of the database.

Copyright © 1988, 2026, Oracle and/or its affiliates.

The search language 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, 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:

Field or Control	Description
Base Language	You can specify search text only for the base language tables. The system searches 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 ignores 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 searches the related language tables for the text that you have 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 are only those instances where the search criteria for both the base language and related language have been met.

Disabling the Replace Process

To disable the Replace process, select the **Search Only (No Replace)** check box.

Selecting this check box ensures that the criteria is 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 check box also determines whether the status is set to *Complete* or *Searched* after you search. If you select the **Search Only (No Replace)** check box, then no further action is possible after the search and the status is updated to *Complete*. Otherwise, the status is updated to *Searched*.

Defining Your Search Scope

To define your search scope:

1. Specify whether to search long text fields.

   Select the **Search Long Fields** check box to include the long description 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 must choose which types of database objects you want to search.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Use the **Search Objects** group box to specify which types of objects are searched. Each check box in this group box represents a group of searchable objects. Select the groups that you want to search.

> **Note**
>
> With certain 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:

Field or Control	Description
Records, Fields, Xlats	Includes record descriptions, field labels, and translate (xlat) values.
Business Processes	Includes the descriptions for activities and business process maps, and the labels of the icons on those maps.
Queries and Strings	Includes the descriptions for queries and strings.
Pages	Includes descriptions of pages and page text that does not come from a field definition or translate value.
Menus	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 so, select **Exclude & from Menu Search.**
Other System Objects	Includes all other PeopleTools objects, such as application messages, component interfaces, and process definitions.
Other Application Objects	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. The PeopleSoft system provides a PeopleSoft Data Mover script, TSRECPOP.dms, which creates this list of translatable application tables. If you have customized the system with new translatable tables, you must run this script to recreate the table.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Field or Control	Description
Messages	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 the PeopleSoft system, message sets 1–999 are reserved for PeopleTools messages. Messages sets 1000–19999 are used by the applications. Selecting either the **Application Messages** or **PeopleTools Messages** check boxes narrows your search to the selected numeric range. You can also select **All Messages.** Alternatively, you can narrow your search to a specific range of messages. To do this, select the **All Messages** check box, and then click the prompt button to the right to open the page where you can enter the desired message set range.

Defining the Search Text

To define the search text:

1. Enter the search and text for the base language, the search language or both.

   There are three columns for search text. Different choices are available depending on the languages you have selected to search and whether you have allowed replacing.

Field or Control	Description
Search Text - Base	This option is available if you are searching the base language or both the base and non-base languages.
Search Text - Non Base	This option is available if you are searching the non-base language or both the base language and the non-base language.
Replacement Text	This option is only available when the **Search Only (No Replace)** check box is not selected.

   For each language that you're searching, enter the text for which you want to search. The system searches only the appropriate language tables.

> **Note**
>
> 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, 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.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Use the three **abc** buttons to automatically put the text in the specified case.

Field or Control	Description
ABC	Capitalizes the text.
abc	Lowercases the text.
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:

Field or Control	Description
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 *Department,* your results won't include fields that contain *Department Code*.
Full Word Match	Returns all occurrences of the text string where the string is not embedded in another word. In this case, searching for *Department* returns occurrences of *Department Code*, but will not return occurrences of *Departments* or *Departmentalize*.
Like Text Match	Returns all occurrences of the text string

Related Topics

- The PSSQR.INI and PSSQR.UNX Files
  Most of the parameters that affect globalization of SQR for PeopleSoft are set in the configuration file. The PeopleSoft system delivers default configuration files, which are located in the *PS_HOME*\sqr directory. The files are named pssqr.ini on Microsoft Windows, and pssqr.unx on Unix/Linux. Because different configurations are needed for some of the supported languages, PeopleSoft delivers eight configuration files on each platform. These language-specific configuration files are named pssqr*language_code*.ini or pssqr*language_code*.unx. When the SQR process is started from PeopleSoft Process Scheduler, one of these configuration files is selected based on the user's language preference set in the My Personalization page. When language-specific files are not found, PeopleSoft Process Scheduler selects pssqr.ini/unx without *language_code.*
- Reviewing Search Criteria
  To see the search criteria for any search that's been defined, select **PeopleTools**, and then **Translations**, and then **Modify Terminology**, and then **Define Translations Search**.
- 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 so, 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.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Cloning Existing Search Criteria

If you want to set up multiple similar searches, expedite data entry by cloning an existing search. Cloning enables you to start with most of your criteria already in place.

To clone a search definition:

1. Select **PeopleTools**, and then **Translations**, and then **Modify Terminology**, and then **Translations Search Criteria**.

   You are prompted to identify a search.

2. Enter the identifiers for the search definition.

   Use the standard search method to access the Translations Search Criteria page.

3. Click the **Copy Search Criteria** button.

   You are prompted for a unique search identifier.

4. Enter a **Text Search Identifier** that will be the unique name for the new search.

5. Click **OK.**

Sample Search Criteria

Following are examples of possible searches:

- Replacing base language text.

  If you enter text in the **Search Text - Base** and **Replacement Text** sections, the search and replace is only on the base tables. This is a search and replace because the **Search Only (No Replace)** check box is clear. Because the search and base languages match, you can enter data in the **Search Text - Base** and **Replacement Text** sections of the page.

  Notice that all of the check boxes are selected in the **Search Objects** area. That means that the system searches all of the available search objects for the word *Department.* When the search results are returned, you can replace each instance of the word *Department* with the term *Cost Center.* The text replacement occurs in the base language tables.

- Searching base language text.

  For example, enter the word *Product* in the **Search Text - Base** section only. In this case the system searches for the word *Product* in the base language tables and shows all occurrences of *Product.* You might do this as a preliminary effort to locate the word *Product,* which might have been translated.

- Searching non-base language text.

  For example, if you enter the word *Producto* (and select the full word match) in the **Search Text - Non Base** section, 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 locate all of occurrences of a translation.

- Replacing non-base language text.

  For example if you enter the words *Producto*, *PRODUCTO*, and *producto* in the **Search Text - Non Base** section (and select the full word match for each instance), and enter the words *Articulo*, *ARTICULO*, and *articulo* in the Replacement Text section, the system searches for all instances of the word Producto and gives you the option to change each instance to the word *Articulo*.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Using the different cases enables you to catch all occurrences.

- Searching base language and non-base language text.

  Select the **Both** option so that 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 ignores where *Product* has already been translated to *Articulo* and searches only for the instances where *Product* has been translated to *Producto*. This search also enables you to change each instance of *Producto* to *Articulo*.

Running the Search Process

After you define and save your search criteria (and select the **Search Ready** check box), you are ready to run a search.

To run a search:

1. Select **PeopleTools**, and then **Translations**, and then **Modify Terminology**, and then **Run Translations 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 Run Translations Search page.
3. Verify that the appropriate search definitions will be processed.

   By default, the search includes 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 the **Search 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 that you're satisfied with the search criteria. Remember that running the search won't affect your data, but it can still be inconvenient to run the wrong search, as you can't go back and modify the search criteria.
4. Click **Run.**

   The Process Scheduler Request page appears.
5. Complete the Process Scheduler Request page.

   See Process Scheduler: Understanding Run Control IDs.
6. Click **OK.**

   The search runs. When the Search process is complete, you can navigate to the Translations Search Replace page to see the results.

Viewing and Replacing Terminology Search Results

After a search is completed, select **PeopleTools**, and then **Translations**, and then **Modify Terminology**, and then **Translations Search Replace** to review the search results.

Copyright © 1988, 2026, Oracle and/or its affiliates.

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, verify replacement text, and indicate when to run the Replace process.

This section discusses how to:

- Review search results.
- Verify replacement text.
- Enable or disable replace processing.

Reviewing Search Results

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.

Field or Control	Description
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 is 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 is replaced.
Replacement Text, Too Long, and Replace	These columns are used to finalize the replacement text.

2. Look to see which pages display the search text.

Field or Control	Description
Pages	This column indicates the number of pages on which the 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.
View Text Search Details	Click this button to display details about that occurrence of the search text. View the record and field where this instance was found, as well as the keys for each occurrence of the text in the record. Also view the text from the base language table, the corresponding translation from the related language table, and the suggested replacement text.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Field or Control	Description
View Text Found	Click this button to display 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 shows which word or words were found in this record.

3. (Optional) Review the information about the record and field where the text was found.

Field or Control	Description
Field Name	System name of the field where the text resides.
Length	Displays the length of the field. This information is useful when you plan to replace text because it tells you the maximum length of any replacement text.
Record	Name of the record where the occurrence was found.
RLang Rec (related language record)	Name of the corresponding related language table.

4. (Optional) Inspect the original search criteria.

Field or Control	Description
View Search Criteria	Click this button to display a read-only version of the search criteria. This can help you understand the search results.

Verifying Replacement Text

To verify replacement text:

1. Inspect the suggested replacement text.

Field or Control	Description
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 replaces the non-base text unless you searched only the base language, in which case it replaces the base text.

2. If necessary, modify the replacement text.

   Modify the replacement text if you're not happy with the suggested text or if the suggested text is too long for the field.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Field or Control	Description
Too Long	The system selects this 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. **Note:** When the replacement text is too long, the search text won't be replaced during the Replace process.

3. Selectively enable and disable replacement processing.

Field or Control	Description
Replace	Use this check box to replace some occurrences of the search text without replacing all of them. Select the check box 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 are 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.

Enabling or Disabling Replace Processing

To enable or disable replace processing:

1. To enable replace processing, select the **Replace Ready** check box.

   The replace process picks up only searches where this check box is selected.

2. To permanently disable replace processing for this specific search, click the **Complete** button.

   This button changes the status of the results to Complete. Once this happens, no further processing is possible. If you have edited messages on the Replace Messages page, those changes become permanent.

Copyright © 1988, 2026, Oracle and/or its affiliates.

3. Click the Save button.

Related Topics

- 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. You can use this feature at any time in the Search, Replace, and Undo process.

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. You can use this feature at any time in the Search, Replace, and Undo process.

> **Note**
>
> You must manually update message text. If you continue to the Undo step, you must also manually back out of your changes. The automatic Replace and Undo process for other database objects does not apply to message text.

To review search results and update message text:

1. Click the **Review Messages** button to load the results into the page.

   Once the results are loaded, you can see each message that contained the search text from your search.

2. (Optional) Inspect the original search criteria.

   Click the View Search Criteria button to display a read-only version of the search criteria.

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.

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, edit the text directly on this page. Saving the page updates the Message Catalog.

   > **Note**
   >
   > You must update message text manually. The automatic Replace process for other database objects does not apply to message text.

5. Click the Save button.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Running the Replace Process

After you have finalized your replacement text and settings in the Translations Search Replace page, you must 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 have one last chance to back out any changes made by the process.

See Replacing Message Text.

> **Warning**
>
> 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 have finished manually editing your messages, before you change the status.

To run the Replace process:

1. Select **PeopleTools**, and then **Translations**, and then **Modify Terminology**, and then **Run Translations 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 Run Translations Replace page.
3. Verify that the appropriate search definitions will be processed.

   By default, the Replace includes 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 Translations Search Replace page and clear the **Replace Ready** check box. If a search is missing, go back to the Translations Search Replace page and select the **Replace 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.
5. Complete the Process Scheduler Request page.

   See Process Scheduler: Submitting Process Requestsand Process Scheduler: Scheduling Process Requests.
6. Click **OK.**

   The Replace process runs. Once the replace is complete, you can navigate to the Review Translations Replace page to view the results.

Viewing and Undoing Terminology Replacement Results

This section discusses viewing and undoing terminology replacement results.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Undoing Data Replacements

After you run the Replace process, select **PeopleTools**, and then **Translations**, and then **Modify Terminology**, and then **Review Translations Replace** to review your results and decide whether to undo any of the replacements.

This section discusses how to:

- Replace replacement text.
- Enable or disable undo processing.

Reviewing Replacement Text

To review replacement text:

1. Review the results of the Replace process.

   The Review Translations Replace 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. Control which occurrences are replaced using the **Undo** check boxes.

   Select **Undo** for those rows where you want to back out of 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 are affected.

3. Click the Save button.

Enabling or Disabling Undo Processing

To enable or disable undo processing:

1. To enable replace processing, select the **Undo Ready** check box.

   The Undo process picks 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 changes 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.

   See Replacing Message Text.

3. Click the Save button.

Copyright © 1988, 2026, Oracle and/or its affiliates.

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.

To access the Undo Messages page, select **PeopleTools**, and then **Translations**, and then **Modify Terminology**, and then **Run Translations Undo**.

> **Note**
>
> Just as you have to replace message text manually, so must you back out of any changes manually. The automatic replace and Undo process for other database objects does not apply to message text.

Related Topics

- 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. You can use this feature at any time in the Search, Replace, and Undo process.

Running the Undo Process

To run the Undo process:

1. Select **PeopleTools**, and then **Translations**, and then **Modify Terminology**, and then **Run Translations 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 Run Translations Undo page.
3. Verify that the appropriate search definitions will be processed.

   By default, the undo includes 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 Review Translations Replace page and clear the **Undo Ready** check box. If a search is missing, go back to the Review Translations Replace 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.
5. Complete the Process Scheduler Request page.

   See Process Scheduler: Understanding Run Control IDs.
6. Click **OK**.

Copyright © 1988, 2026, Oracle and/or its affiliates.

The Undo process runs. Once the undo is complete, navigate to the Translations Search Results page to view the results.

Reviewing Terminology Search Information

Three read-only pages provide you with information about searches.

This section discusses how to work with the pages.

Reviewing Search Criteria

To see the search criteria for any search that's been defined, select **PeopleTools**, and then **Translations**, and then **Modify Terminology**, and then **Define Translations Search**.

This page is a display-only version of the Search Profile page.

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 to continue the Search, Replace and Undo process, you must go to the Translations Search Replace pages. The following table explains which component you see, based on the status of the search:

Status	Component	Navigation
New	Define Translations Search	PeopleTools, Translations, Modify Terminology, Define Translations Search
Searched	Translations Search Replace	PeopleTools, Translations, Modify Terminology, Translations Search Replace
Replaced/Undone	Review Translations Replace	PeopleTools, Translations, Modify Terminology, Review Translations Replace
Complete/Undone/Replaced	Translations Search Results	PeopleTools, Translations, Modify Terminology, Translations Search Results
Complete/Undone/Replaced	Translation Search Only Result	PeopleTools, Translations, Modify Terminology, Translation Search Only Result

Click the **Copy Search Criteria** button to clone this search definition.

Viewing Results for Search Only Searches

When a search is defined as Search Only (No Replace), the status is set to *Complete,* and prevents you from viewing the search results on the Translations Search 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**, and then **Translations**, and then **Modify Terminology**, and then **Translation Search Only Result**.

This page is a display-only version of the Translations Search Replace page.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Related Topics

- Viewing and Replacing Terminology Search Results
  After a search is completed, select **PeopleTools**, and then **Translations**, and then **Modify Terminology**, and then **Translations Search Replace** to review the search results.

Viewing Search Results for All Searches

To see the search results for any search, select **PeopleTools**, and then **Translations**, and then **Modify Terminology**, and then **Translations Search Results**.

This page is a display-only version of the Review Translations Replace page.

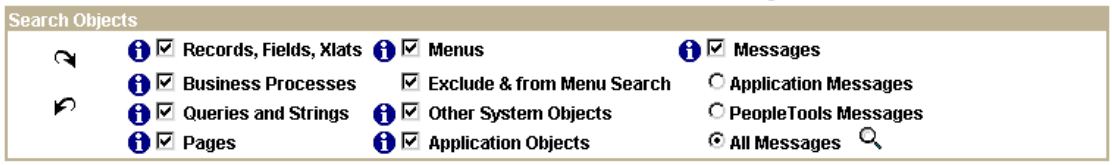
Related Topics

- Viewing and Undoing Terminology Replacement Results
  This section discusses viewing and undoing terminology replacement results.

Selecting Terminology Search Objects

When you define your search, you select the types of objects to include in your search. You do this by selecting from predefined groups of object types.

This example illustrates the fields and controls on the Groups of object types.



The PeopleSoft system 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, denoted by the “i” icon.

You can use the 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. The PeopleSoft system provides a PeopleSoft Data Mover script to do this for you.

This section discusses how to:

- View searchable objects.
- Modify searchable objects.

Viewing Searchable Objects

Use the Text Search Records page to view additional information about the searchable records.

To view the searchable fields in any searchable record:

Copyright © 1988, 2026, Oracle and/or its affiliates.

1. Select **PeopleTools**, and then **Translations**, and then **Modify Terminology**, and then **Text Search Records**.

   You are prompted to identify the record to view.

   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 to view.

   Use the standard search or add method to enter the user ID and access the Text Search Records page.

3. Examine the information about this record.

   The **Record** and **Related Language Record Name** fields identify the record you're viewing.

   The **Combined Language Table** check box is selected when the base record and the related language record are the same; you cannot change this setting.

   The **Object Type** field identifies which searchable group of objects includes this record.

> **Warning**
>
> Although you can modify the **Object Type** field on this page, it is not advisable. As delivered, records are already logically organized into appropriately named groups; changing the association causes 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 so, 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.

Reviewing 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 have customized your system and modified the delivered terminology, you may want to minimize the upgrade compare differences by doing one of the following:

Copyright © 1988, 2026, Oracle and/or its affiliates.

- 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.

> **Note**
>
> Upgrading becomes more complicated and involves extra steps when you have made terminology changes. Be sure to consider this cost when you decide whether terminology changes are necessary.

Copyright © 1988, 2026, Oracle and/or its affiliates.

15

Adding New Languages

Understanding the Addition of New Language Definitions

PeopleSoft provides translations in several 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, you can maintain data in all languages supported by the Unicode standard in a single PeopleSoft database.

As shipped, PeopleTools includes definitions primarily for the languages for which translations are provided by PeopleSoft. Definitions for some additional languages are provided in anticipation of future translations. If you plan to use additional languages in your system, you must first define these languages in PeopleTools.

There are several places where you must define new languages before PeopleTools can use and recognize them. Work with your system administrator to ensure that you complete all the necessary steps described in this section before using a new language in PeopleTools.

> **Note**
>
> You do not need to perform these steps for any language for which PeopleSoft has provided translations.

> **Note**
>
> Functionality in a customer added language may be limited by the technology stack, for example the rendering engine in reporting tools.

Adding New Language Codes to the System

This section discusses adding new language codes to the system.

See the product documentation for *PeopleSoft 9.2 Application Installation* for your database platform.

> **Note**
>
> Functionality in a customer added language may be limited by the technology stack, for example the rendering engine in reporting tools.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Determining PeopleSoft and ISO Codes for Your Language

The PeopleSoft system 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 developers. When adding a new language, you must 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, ensure that the language code contains only three uppercase US-ASCII letters—no numbers, spaces, or accented characters.

This table lists the PeopleSoft language codes that are allocated:

Code	Language
ARA	Arabic
BUL	Bulgarian
CFR	Canadian French
CRO	Croatian
CZE	Czech
DAN	Danish
DUT	Dutch
ENG	English
ESP	Spanish
FIN	Finnish
FRA	French
GER	German
GRK	Greek
HEB	Hebrew
HUN	Hungarian
JPN	Japanese
KOR	Korean
MAY	Malay
NOR	Norwegian (Bokmål)
POL	Polish
POR	Portuguese
ROM	Romanian
RUS	Russian
SER	Serbian
SLK	Slovak
SLV	Slovenian
SVE	Swedish
THA	Thai
TUR	Turkish
UKE	UK English
ZHT	Traditional Chinese

Copyright © 1988, 2026, Oracle and/or its affiliates.

Code	Language
ZHS	Simplified Chinese

As an example, we will chose *QUE* for Quechua (the QUE code is derived from the language) as our additional language.

Once you have chosen a three-letter code for your new language, you must determine which ISO locale corresponds to that language. ISO locales are comprised of 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 Quechua language, so the appropriate ISO locale is *qu*. If we were adding a specific dialect of Quechua, we would use a specific ISO locale, such as *qu_QU*. But as long as we're adding the generic Quechua language, we can use *qu* 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). You can easily create those language derivations from a related language using the PeopleTools Terminology Management tool.

Determining Appropriate Non-Unicode Character Sets

Although the majority of the PeopleSoft system runs in Unicode, some operations, such as file system operations on Unix/Linux platforms and integration to some third-party products, cannot use Unicode characters. For each language in your database, you must determine which non-Unicode 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 the PeopleTools database. For each new language you add, 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 the PeopleSoft database. For example, if you are using a DB2 UDB for OS/390 and z/OS database, you must specify the EBCDIC CCSID specified in your system's DSNZPARM configuration parameters.

This table shows some of the character sets that are supported by PeopleTools. Remember that this table is an excerpt from the complete list.

Character Set	Description
ISO-2022-KR	ISO-2022-KR (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)

Copyright © 1988, 2026, Oracle and/or its affiliates.

Character Set	Description
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)

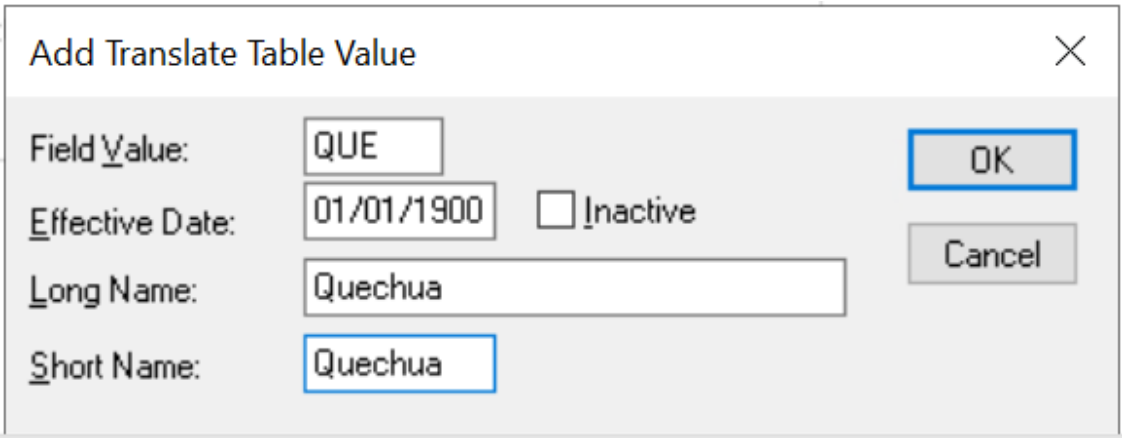
For the complete list, query the PSCHARSETS table in your SQL tool or in PeopleSoft Query. The complete list of character sets from PSCHARSETS is also provided in this PeopleBook.

See Character Sets Across the Tiers of the PeopleSoft Architecture.

Adding New Translate Values to the LANGUAGE_CD Field

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

This example illustrates the fields and controls on the Add Translate Table Value dialog box.



To add the language code to the Translate table:

1. In Application Designer, open the field named *LANGUAGE_CD*.
2. Select **File**, and then **Definition Properties**.
3. Move to the Translate Values tab and click the **Add** button.

   The Add Translate Table Value dialog box appears.
4. Add the new three-letter language code and a description of your new language.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Use an effective date of 01/01/1900 to ensure that the language is always accessible to your applications. When adding the long and short names 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.

5. Save the changes.

In this example, the database base language is English, so *Quechua* is used as both the long and short name of the new language. Once the new language is defined, Quechua translators might want to translate this new value *qheshwa.*

Managing Languages in the PSLANGUAGES Table

Access the Manage Installed Languages page (**PeopleTools**, and then **Utilities**, and then **International**, and then **Manage Installed Languages**).

This example illustrates the fields and controls on the Manage Installed Languages page.

Language

1-34 of 34

	Language Code	Language	Enabled	Directionality	ISO Locale	Default Character Set	Spell Check Language	Windows Character Set
1	ARA	Arabic	☑	Right-to-Left	ar	ISO_8859-6	English (US and UK)	CP1256
2	BUL	Bulgarian	☑	Left-to-Right	bg	ISO_8859-5	English (US and UK)	CP1251
3	CFR	Canadian French	☑	Left-to-Right	fr-ca	ISO_8859-1	French	CP1252
4	CRO	Croatian	☑	Left-to-Right	hr	ISO_8859-2	English (US and UK)	CP1250
5	CZE	Czech	☑	Left-to-Right	cs	ISO_8859-2	Czech	CP1250
6	DAN	Danish	☑	Left-to-Right	da	ISO_8859-1	Danish	CP1252
7	DUT	Dutch	☑	Left-to-Right	nl	ISO_8859-1	Dutch	CP1252
8	ENG	English	☑	Left-to-Right	en	ISO_8859-1	English (US and UK)	CP1252
9	ESP	Spanish	☑	Left-to-Right	es	ISO_8859-1	Spanish	CP1252
10	FIN	Finnish	☑	Left-to-Right	fi	ISO_8859-1	Finnish	CP1252
11	FRA	French	☑	Left-to-Right	fr	ISO_8859-1	French	CP1252
12	GER	German	☑	Left-to-Right	de	ISO_8859-1	German (new)	CP1252
13	GRK	Greek	☑	Left-to-Right	el	ISO_8859-7	Greek	CP1253
14	HEB	Hebrew	☑	Right-to-Left	he	ISO_8859-8	English (US and UK)	CP1255

You can update and add the languages used by your PeopleSoft system to the PSLANGUAGES table using the Manage Installed Languages page.

> **Note**
>
> Whenever modifications to the PSLANGUAGES table are made, such as on the Manage Installed Languages page, the application server's in-memory cache must be refreshed by restarting the application server in order to cache the new values.

Use the Manage Installed Languages page to:

- Enable or disable a language.

  The PSLANGUAGES table is used by many PeopleTools utilities to determine which languages defined in the database are actually in use, rather than merely being defined but unused.

Copyright © 1988, 2026, Oracle and/or its affiliates.

> **Note**
>
> English is always delivered as an enabled language. If you do not want to allow your users to sign into the PeopleSoft Pure Internet Architecture using English, you can disable English on this page. However, you must ensure that the language selected in psprcs.cfg is set to one of the languages that remains enabled for your system to operate properly. In addition, for the PeopleTools development environment, Configuration Manager must be set to use one of the languages that remains enabled.

- Specify a default, non-Unicode character set.

  The PSLANGUAGES 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.

- Identify the spell check language.

  You also use the PSLANGUAGES table to select the language of the spell check dictionary that is associated with a given language code. The spell check language that you select for a language code is the default spell check dictionary that is associated with that language when the user signs on to the system in that language. The user can override the default spell check dictionary through the My Personalizations option on the homepage.

  List of supported Spell Check Language dictionaries:

  – Afrikaans
  – Afrikaans and US English
  – Catalan
  – Catalan and US English
  – Czech
  – Danish
  – Danish and US English
  – Dutch
  – Dutch and US English
  – English (UK)
  – English (US and UK)
  – English (US)
  – Finnish
  – Finnish and US English
  – Finnish morphological database
  – French
  – French and US English
  – German (new)
  – German (new) and US English
  – German (old)

Copyright © 1988, 2026, Oracle and/or its affiliates.

- German (old) and US English
- Greek
- Hungarian
- Hungarian morphological database
- Italian
- Italian and US English
- Norwegian
- Norwegian and US English
- Polish
- Polish morphological database
- Portuguese
- Portuguese (Brazilian)
- Portuguese (European)
- Russian
- Spanish
- Swedish

See Applications User's Guide.

- Specify the character set for encoding query results to be downloaded in comma-separated values (CSV) format.

  The PeopleSoft system downloads query results to a file in CSV format. You can use the PSLANGUAGES table to select a character set value for a given language code to encode the results.

Field or Control	Description
Language Code	Enter the three-letter language code from the PSXLATITEM table.
Enabled	Select this option to indicate that the language is in use. When this option is selected, login or multi-language entry with this language is allowed. This option is selected automatically during the process of loading translations into the database. However, this option allows you to manually enable a language without loading the translations.
Directionality	Use the drop-down list box to select left-to-right or right-to-left as the language direction..
ISO Locale	Use the lookup to select the ISO locale code from the PSLOCALEDEFN table. The code consists of an ISO 639 language code, optionally followed by an ISO 3166 country code.
Default Character Set	Use the lookup to select the character set from the PSCHARSETS table. The default character set determines the default encoding for input and output files.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Field or Control	Description
Spell Check Language	Use the drop-down list box to select the spell check language from the PSXLATITEM table. This enables you to select the language of the spell check dictionary that is associated with a given language code.
Windows Character Set	Use the Lookup option to select a Microsoft code page to use to encode CSV formatted query results for the corresponding language code. The character set must be valid for Java encoding and mime-type, or downloads will fail with an error. The default value for all language codes is *CP1252*.

> **Note**
>
> You can override the information specified on the Manage Installed Languages page using the Look and Feel tab on the Web Configuration page.

Related Topics

- Portal Technology

Modifying Configuration Manager Windows Resources

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

To add your new language to the list of languages in PeopleSoft Configuration Manager:

1. In \SRC\RES\ENG\PSCFG, edit the file pscfg.rc using a text editor.

   This file stores the list of languages used in the Display tab of the PeopleSoft Configuration Manager. These languages are stored as strings with the name IDS_LANGx, where *x* is a sequential number. To add your new language, find the next available IDS_LANG entry and change the text to match your new language using the format *LanguageCode – Description.* Then, update the entry IDS_NUMLANGUAGES to specify the new highest IDS_LANG entry that you used.

2. Copy \src\res\eng\inc\version.rcx to \src\res\eng\PSCFG.

3. Set up your environment by running a batch files, as follows:

   ```
   set respath= C:\temp\RES
   set incdir=c:\temp\SRC\INC
   set vcinstall=c:\apps\dvlp\visualstudio\vc98
   set include=%include%;%incdir%
   call %vcinstall%\bin\vcvars32 x86
   ```

4. Use the MAKEALTL.BAT batch file to compile PSCFGENG.DLL.

Copyright © 1988, 2026, Oracle and/or its affiliates.

For example:

```
makealtl ENG PSCFG
```

Then copy the DLL file to the *PS_HOME*\bin\client\winx86 directory.

5. Use MAKEALTL.BAT to compile the new language version you added.

   In this case you will compile PSCFGxxx.dll, where xxx is the new language.

   If you plan to translate Windows resources into the new language, create another directory structure under \SRC\RES\ to include the new translations. To do so, copy the existing \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 the existing \SRC\RES\ directory, matching the three-letter PeopleSoft code for the new language.

   Then, translate the strings in the RC files in your new directories, and use the MAKEALTL.BAT batch file to compile the alternate language DLLs. For example:

   ```
   makealtl QUE PSCFG
   ```

   Then copy the DLL file to the *PS_HOME*\bin\client\winx86 directory.

6. Use MAKEARTL.BAT to compile the language(s) that will use the new language.

   For example, if users using the French language were going to use the Quechua language you would compile PSCFGFRA.DLL as follows:

   ```
   makealtl FRA PSCFG
   ```

   Then copy the DLL file to the *PS_HOME*\bin\client\winx86 directory.

Modifying the PeopleSoft Pure Internet Architecture Sign-in Page

The PeopleSoft Pure Internet Architecture enables users to select their sign-in language by selecting the language from the sign-in page. This page is maintained on the web server as a static HTML file and is not generated from a PeopleTools page definition. You must edit this HTML file to add the new language code.

Work with your web server administrator to edit the signin.html file 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 new language*. If the language does not use Latin characters, you might 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 appropriately for that language.

In the following example, we added the lines shown in boldface to the signin.html file.

```
<tr>
<td width="50%" class="pslogintext"><a href="?cmd=login&amp;languageCd=POL"
title="<%=psPOL%>">Polski</a></td>
<td width="50%" class="pslogintext"><a href="?cmd=login&amp;languageCd=POR"
title="<%=psPOR%>">Portugu&ecirc;s</a></td>
</tr>
<tr>
<td width="50%" class="pslogintext"><a href="?cmd=login&amp;languageCd=QUE"
title="<%=psQUE%>">Quechua</a></td>
```


Copyright © 1988, 2026, Oracle and/or its affiliates.

```
<td width="50%" class="pslogintext"><a href="?cmd=login&amp;languageCd=FIN"
title="<%=psFIN%>">Suomi</a></td>
</tr>
```

Modifying Text and Error Properties Files for New Languages

The text.properties and errors.properties files are located in *PS_HOME*\webserv*web_server*\applications\peoplesoft\PORTAL.war\WEB-INF\psftdocs*DBdomain*.

To modify text.properties and errors.properties files for new languages:

1. Edit the base text.properties file to add a language title for the new language title in the correct alphabetical position in the file.

   For example, for Quechua, its language title is entered in the following position:

   ```
   #language titles
   ...
   psNOR=Norwegian
   psPOL=Polish
   psPOR=Portuguese
   psQUE=Quechua
   psROM=Romanian
   psRUS=Russian
   ...
   ```

2. Copy the modified base text.properties and errors.properties to two files named after your three-letter language code in this format: *filename_XXX*.properties in which *XXX* is the three-letter language code.

   For example, for Quechua, the file names would be text_QUE.properties and errors_QUE.properties.

3. Set language-specific properties in the text_*XXX*.properties file.

   See Understanding the Addition of New Language Definitions.

4. Translate the non-setting specific information in the text_*XXX*.properties and errors_*XXX*.properties files as needed for your new language.

5. Run the native2ascii utility to convert your translated properties files if needed.

   See Adding New Language Codes to the System.

6. Add the new language to each of the language-specific versions of text.properties files that are relevant to your system.

   > **Note**
   >
   > Translate the new language title to each relevant language, run native2ascii on a temporary file that includes the translated language title, and copy and paste each encoded translated language title into the appropriate text_*XXX*.properties file.

7. After you have modified the text and error properties files, you must restart the web server.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Text Property File Variables

You must set the following properties in the text_*XXX*.properties file:

> **Note**
>
> Do not translate any of the values you enter for these properties.

Variable	Description
active	Set the ISO language code. Preface the *active* variable with the two character ISO code for the language and set the variable equal to *-s*. For example: qu_active=-s
direction	Set the direction that text is read. The valid options are: • *ltr*. Left to right. • *rtl*. Right to left.
alignstart	Set the starting point. The valid options are: • *right* • *left* The value you select must be consistent with the directionality that you specify. For example, if you set the *direction* variable to *ltr*, then you must set the *alignstart* variable *left*. If you set the *direction* variable to *rtl*, then you must set the *alignstart* variable to *right*.
alignend	Set the ending point. The valid options are: • *right* • *left* The value you select must be consistent with the directionality that you specify. For example, if you set the *direction* variable to *ltr*, then you must set the *alignend* variable *right*. If you set the *direction* variable to *rtl*, then you must set the *alignend* variable to *left*.

The following example shows a text.properties file entry for French:

```
fr_active=-s
direction=ltr
alignstart=left
alignend=right
```

File Format

Properties files are saved in a special format. US-ASCII Unicode characters must be represented by a Java-style Unicode escape sequence in the format \udddd, where dddd represents the hexadecimal value of the Unicode character.

Copyright © 1988, 2026, Oracle and/or its affiliates.

native2ascii

The native2ascii utility, a part of the JDK (Java Development Kit), can be used to transform your properties files into the correct format after it has been translated.

For example, to convert a non-European text.properties file, you will need to run the following:

```
native2ascii –encoding encoding sourcefile destinationfile
```

> **Note**
>
> Use the destination properties files that have the Java-style encoding.

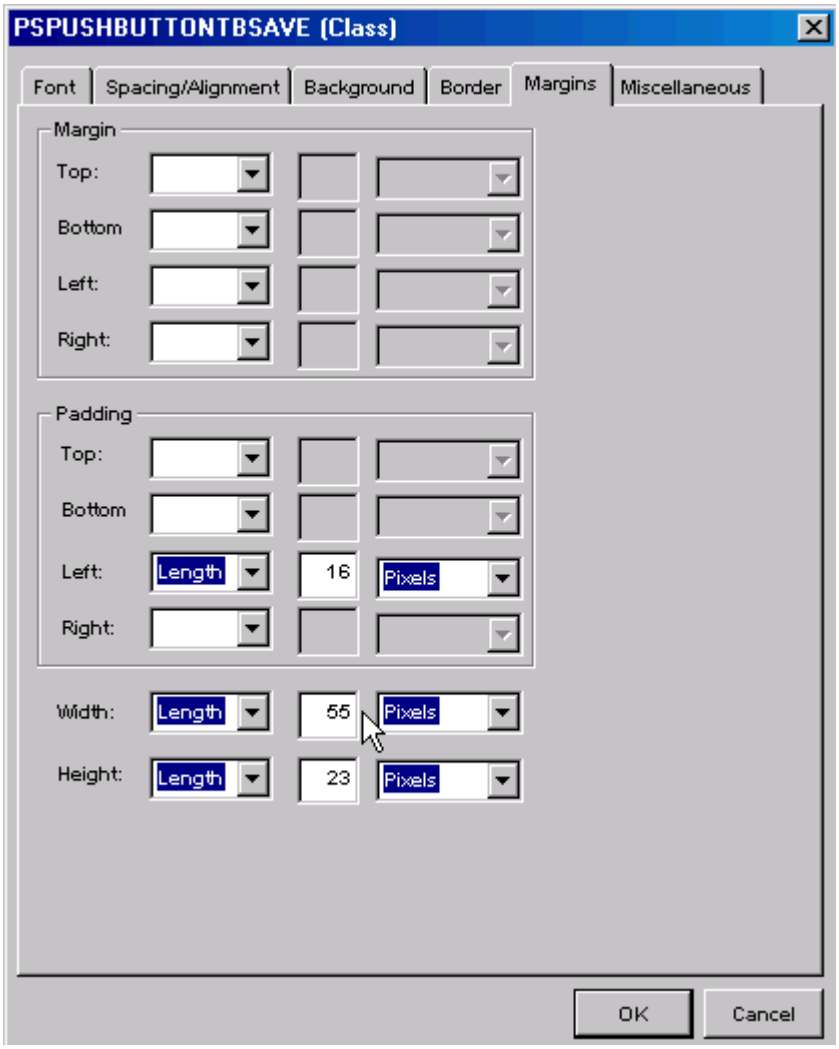
> **Note**
>
> For more information about the native2ascii utility, refer to the oracle's online documentation.

Adjusting Truncated Toolbar Buttons

After you add a new language you may find that images appear truncated on PeopleSoft toolbar buttons. To remedy truncated toolbar buttons, adjust the width of the styleclass.

This example illustrates the fields and controls on the Margins tab for the PSPUSHBUTTONTBSAVE styleclass.

Copyright © 1988, 2026, Oracle and/or its affiliates.



> **Note**
>
> All styleclasses that relate to toolbar buttons start with PSPUSHBUTTONTB*.

To adjust truncated toolbar buttons:

1. Open the stylesheet that contains the truncated styleclass.
2. Open the styleclass.
3. Click the Margin tab.
4. In the Width field, adjust the values.

Copyright © 1988, 2026, Oracle and/or its affiliates.

16

Translating PeopleTools

Understanding PeopleTools Translation

The PeopleSoft system provides utilities and methods for translating all definitional *application objects,* such as pages and reports, built using PeopleTools. There are some *PeopleTools components* that are visible to end-users and should be translated as well. These include messages issued by the Application Server, the Windows components of PeopleTools (such as the PS/nVision designer and PeopleSoft Query) and images used for standard navigation in the PeopleSoft Pure Internet Architecture.

These topics explains the different translatable PeopleTools components, and how you can customize the translations delivered by the PeopleSoft system or translate PeopleTools into new languages.

Related Topics

- Using 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.

Using Alternate Language DLLs

Although the majority of PeopleTools user interface elements viewed by the end user are stored in the database, PeopleTools executable modules do contain some translatable elements. When running PeopleTools components (such as the application server, PS/nVision, and PeopleSoft Query) on Microsoft Windows, the translations are read from alternate language DLLs. An alternate language DLL is a file that stores translated strings for an associated PeopleTools module, a DLL or EXE file. At run time, the system uses the resources in the alternate language DLLs that correspond to the user's language preference.

The PeopleSoft system uses a naming convention that includes the name of the base executable (an executable with English strings) and the language code that identifies the language of the alternate language DLL. An alternate language DLL is named as follows:

PS*ZZZXXX*.dll

Field or Control	Description
ZZZ	Three-character code that identifies the PeopleTools module associated with the alternate language DLL.
XXX	Three-character code that identifies the language of the alternate language DLL.

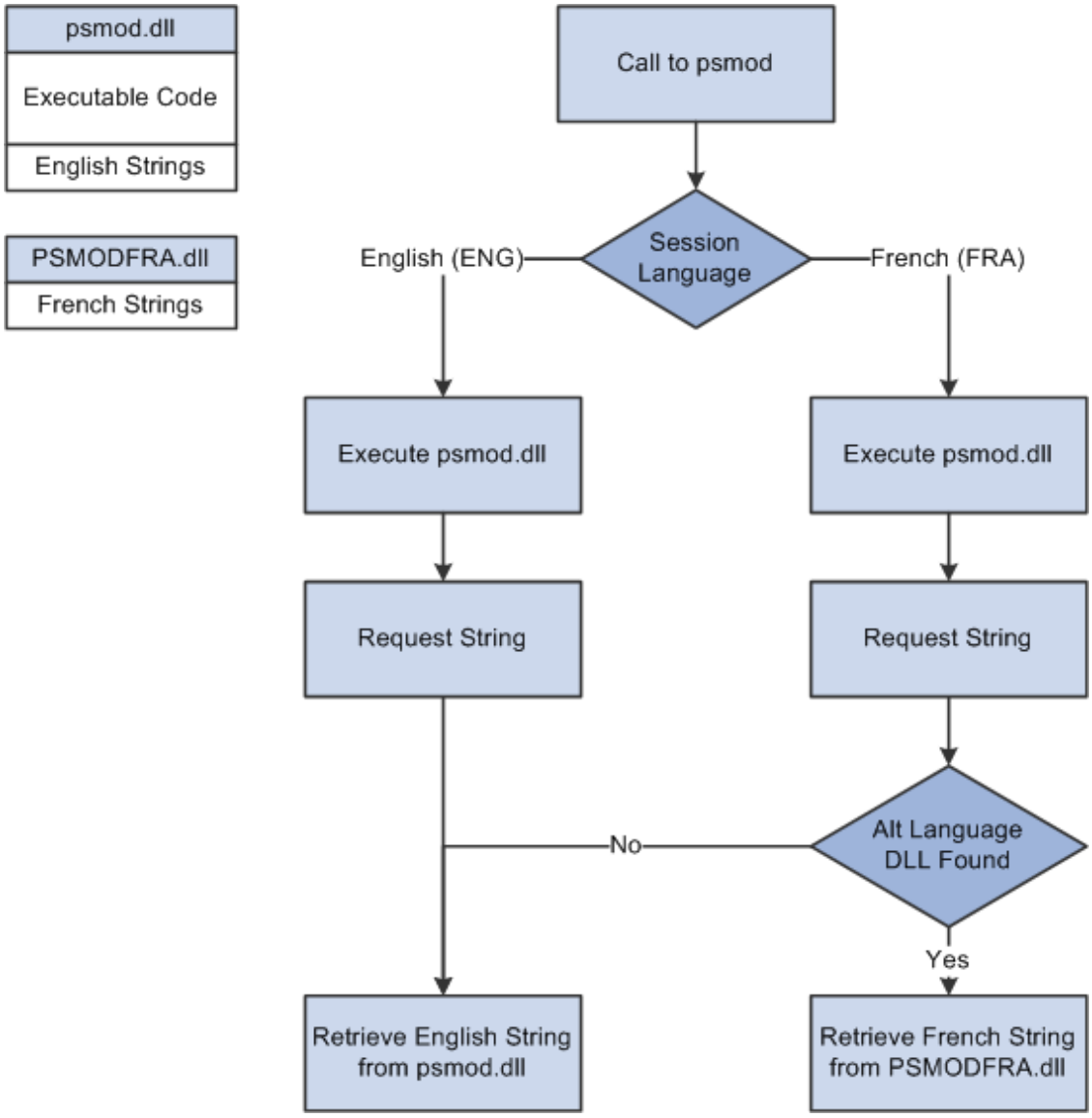
Copyright © 1988, 2026, Oracle and/or its affiliates.

The following diagram illustrates the structure and execution flow of a base module (psmod.dll) and its French alternate language DLL (PSMODFRA.dll). The base executable, psmod.dll, contains all executable code plus English strings. The alternate language DLL is used only to store translated resources. Program execution follows this flow:

1. The system calls psmod.
2. The session language is determined: English (ENG) or French (FRA).
3. In both cases, psmod.dll is executed.
4. Execution of psmod.dll requests a string.
5. If the session language is English, psmod.dll retrieves the English string from itself, psmod.dll.
6. If the session language is French, then the system determines whether the alternate language DLL, PSMODFRA.dll exists.
7. If the alternate language DLL exists, then psmod.dll retrieves the French string from PSMODFRA.dll.
8. If the alternate language DLL does not exist, psmod.dll retrieves the English string from itself, psmod.dll.

The following diagram illustrates the structure and execution flow of a psmod.dll module and its French alternate languagePSMODFRA.dll

Copyright © 1988, 2026, Oracle and/or its affiliates.



Locating Resource Directories

Alternate language DLLs must be present on all application servers. PeopleTools development environments in a three-tier installation also must have the alternate language DLLs present.

Microsoft Windows application servers and PeopleTools development environment workstations use Microsoft Windows resources from the base language and alternate language DLLs. Unix/Linux 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.

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.

Copyright © 1988, 2026, Oracle and/or its affiliates.

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.
\SRC\RES\ENG	A prototypical alternate language development directory for the English language. One of these directories exists for each alternate language. Copy this directory for each new alternate language that you create.
\SRC\RES\ENG\INC	Holds header (.H) files that are common to several DLLs and various icons and bitmaps.
\SRC\RES\ENG\xxxxx	This directory contains directories for most PeopleTools DLL and EXE files found in \SRC\BIN. These directories contain the following file types: • *.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 Microsoft Windows Explorer.

Related Topics

- Compile Translated Resource Files Using MAKEUNIX.BAT
  MAKEUNIX.BAT compiles the translated resources into a single file that is readable by the PeopleSoft Unix/Linux 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.

Translating Resource Files

To translate resource files into a new language, copy the source code for the English translations provided by the PeopleSoft system and use this as a starting point for the new language. Alternatively, if you are only making minor modifications to an existing translation provided by the PeopleSoft system (such as to change terminology), you can use any of the languages provided by the PeopleSoft system as a starting point. This section assumes that you are using the English translations as your start, and that you're adding a new translation.

> **Note**
>
> The PeopleSoft system provides pre-translated PeopleTools alternate language DLLs for many languages. You must follow these steps only if you want to translate PeopleTools into a language that is not provided by the PeopleSoft system or if you want to modify one of the provided translations.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Translating Resource Files

To translate resource files:

1. Determine the three-letter designation of the new language.

   In this example, *QUE* is the designation for Quechua. 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. The PeopleSoft system does not use standard ISO codes for languages, but three-letter codes instead. You can decide the appropriate three-letter code for your language, but you should ensure that it is consistently used across PeopleTools whenever you use this language.

   See Understanding the Addition of New Language Definitions.

2. Copy the contents of \SRC\RES\ENG to the target resource directory.

   For Quechua, the target resource directory is \SRC\RES\QUE.

3. Use Microsoft Developer Studio to convert the English resources to the alternate language.

   A text editor is adequate to change string resources contained in the *.RCX files, however when modifying *.RC files that contain Microsoft Windows dialog boxes and other graphical elements, you should use the resource editor provided in Microsoft Developer Studio bundled with Microsoft Visual C++. The resource editor enables you to size and position elements in dialog boxes and to edit bitmaps, cursors, and other graphic components of the resource files.

4. Set up the environment variables.

   Before compiling your translated resource files, you must 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, set the environment variable TOOLBIN to the *PS_HOME*\BIN\CLIENT\WINX86 directory of your file server, and append the *PS_HOME*\SRC\RES\ENG\INC directory to the environment variable INCLUDE.

5. Compile alternate language DLLs using MAKEALTL.BAT.

6. (Microsoft Windows application servers only) Copy the resultant alternate language DLLs to the BIN\SERVER\WINX86 directory of your Microsoft Windows application server.

7. (Unix/Linux application servers only) Compile Unix/Linux application server resources and transfer to the Unix/Linux application server.

   If you are using a Unix/Linux application server, you must run MAKEUNIX.BAT to compile the resource files into a format that is readable by the Unix/Linux application server.

> **Note**
>
> If you have changed any menu item text, update the Security settings to give users access to the menu items.

Related Topics

- Compiling Translated Resource Files Using MAKEALTL.BAT
  The MAKEALTL.BAT file calls the Microsoft Visual C++ Resource Compiler and Linker to compile translated resources into a Windows DLL. Before running MAKEALTL.BAT, ensure that the environment variables required by the Resource Compiler and Linker are set

Copyright © 1988, 2026, Oracle and/or its affiliates.

correctly in your current DOS window. This table describes and shows an example of these environment variables:

Compiling Translated Resource Files Using MAKEALTL.BAT

The MAKEALTL.BAT file calls the Microsoft Visual C++ Resource Compiler and Linker to compile translated resources into a Windows DLL. Before running MAKEALTL.BAT, ensure that the environment variables required by the Resource Compiler and Linker are set correctly in your current DOS window. This table describes and shows an example of these environment variables:

Environment Variable	Description	Example
%path%	Must include the Microsoft Visual C++ executables directory.	C:\MSDEV\BIN
%include%	Must include the 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:\PT850\SRC\RES\ENG\INC
%lib%	Must include the Microsoft Visual C++ library directory.	C:\MSDEV\LIB
%toolbin%	Must point to the directory containing the PeopleTools executables.	C:\PT850\BIN\CLIENT\WINX86

For customer builds, the PS_HOME variable must be set. For internal builds, the PSVER variable must be set.

For customer builds, another option is to SET CUSTOMER_BUILD=NO and set the PSVER variable.

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 as follows:

   ```
   MAKEALTL language PeopleTools_module
   ```

- *language* is the three-letter code for the language that you are compiling (such as QUE).
- *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.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Compile Translated Resource Files Using MAKEUNIX.BAT

MAKEUNIX.BAT compiles the translated resources into a single file that is readable by the PeopleSoft Unix/Linux 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 also have the Microsoft Visual C++ development environment installed on your workstation. MAKEUNIX.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 you set the %PSVER% environment variable. Set this variable to the base directory in which you installed PeopleTools.

If you have added any new languages as part of your PeopleTools translations, you must edit MAKEUNIX.BAT to include the new languages so that they are included in the PSAPPSRV.RES file.

To run MAKEUNIX.BAT:

1. Open a DOS command prompt window.
2. Change the directory to your \PT850\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/Linux application server resources for all languages and modules in one pass.

   The output from MAKEUNIX.BAT, and PSAPPSRV.RES is placed in *PS_HOME*\APPSERV\UNIX. You must transfer PSAPPSRV.RES to the BIN directory of your Unix/Linux application server using a network file transfer utility, such as FTP.

Translating Objects

Not all translatable elements used internally by PeopleTools can be contained in resource files. Some strings referenced by Java and other components of PeopleTools are stored in Java-style property files. The PeopleSoft Pure Internet Architecture makes use of images for many navigational and user interface elements. Some of these images contain text which must also be translated.

This section discusses:

Translating Properties Files

Java components of PeopleTools, such as the servlets on the web server, derive some strings from Java-style properties files. Java properties files are simple text files that are read at runtime by the servlet; these files contain strings that are displayed to the user, typically in error messages.

These files can be translated using any standard text editor, and are located on the web server. This table show the files that contain translatable text:

Base Properties File	Translated Properties File
text.properties	text_*language_code*.properties

Copyright © 1988, 2026, Oracle and/or its affiliates.

Base Properties File	Translated Properties File
errors.properties	errors_*language_code*.properties

The base version of the file (without a language code) contains the English version of the strings. Copy this file to a new file containing the language code of your target language before translating the contents. For example, if you are translating the resources into Quechua, you would create text_QUE.properties and errors_QUE.properties, and update these files with Quechua translations of the contents.

See Understanding the Addition of New Language Definitions.

When translating into a language that cannot be represented by the Latin-1 character set, such as Japanese, Chinese and Korean, the contents of the Java properties files must be converted into escaped Unicode character references. The Java Development Kit (JDK) contains the native2ascii utility that you can use to convert Unicode character references in a Java properties file into a native character set and vice versa. Refer to the native2ascii utility documentation in the JDK for more information.

Translating Images

The PeopleSoft Pure Internet Architecture uses many images as part of its user interface component, and some of these images contain text. You will typically translate only images that contain textual elements or elements that are specific to one language or culture.

Translating an Image

To translate an image:

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

   Choose **File**, and then **Open** and select the image you want to open.
4. Export the image to a file.

   From the **File** menu, select **Export Image** and then select a filename and image format to save the image to disk.
5. Use a graphics editor to modify the saved image file and update the translation.
6. From the **Edit** menu, select **Update Image.**

   A standard Open dialog box appears.
7. Open the file with the translated image.

   The new graphic replaces the original graphic in the image.
8. Save the image.

   The translated image is saved into the related language table under the language that you selected in the Configuration Manager in the first step.

Copyright © 1988, 2026, Oracle and/or its affiliates.

17

Translating Application Definitions

Understanding Application Definition Translation

If you are running PeopleSoft applications with a multiple language user interface, you should translate any customizations you make to the product using PeopleTools into each of the languages that you are using. PeopleTools provides a full suite of tools to help you translate the application user interface; these are the same tools that PeopleSoft uses internally to provide the translations we ship on the global multi-language CDs.

The main definitions requiring translation in order to provide a fully multilingual user interface are pages and fields. However, PeopleTools allows the translation of many other definitions; everything that the end user sees via the PeopleSoft Pure Internet Architecture can be translated into multiple languages.

You can translate PeopleTools definitions using Application Designer, PeopleSoft Tree Manager, or the translations utilities. The translations utilities are a particularly powerful option for translating fields, translate values, and hard-coded text on pages (that is, text that is not derived from the field description). Given the definitional approach to PeopleTools applications, however, the order in which you translate objects is critical in order to obtain the most leverage from your translation work.

Using Translation Designer

The Translation Designer within Application Designer provides an efficient mechanism for translating PeopleSoft pages, which are the highest volume and highest profile definitions.

Page definitions include:

- Field labels.
- Page text (not derived from field labels).
- Translate values (which normally appear either as radio button labels or as values in drop-down lists).

Translation Designer 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 their translations side-by-side instead of simply overwriting source text with a translation.

This section discusses how to:

- Open Translation Designer.
- Use Translation Designer display options.
- Work in Translation Designer.

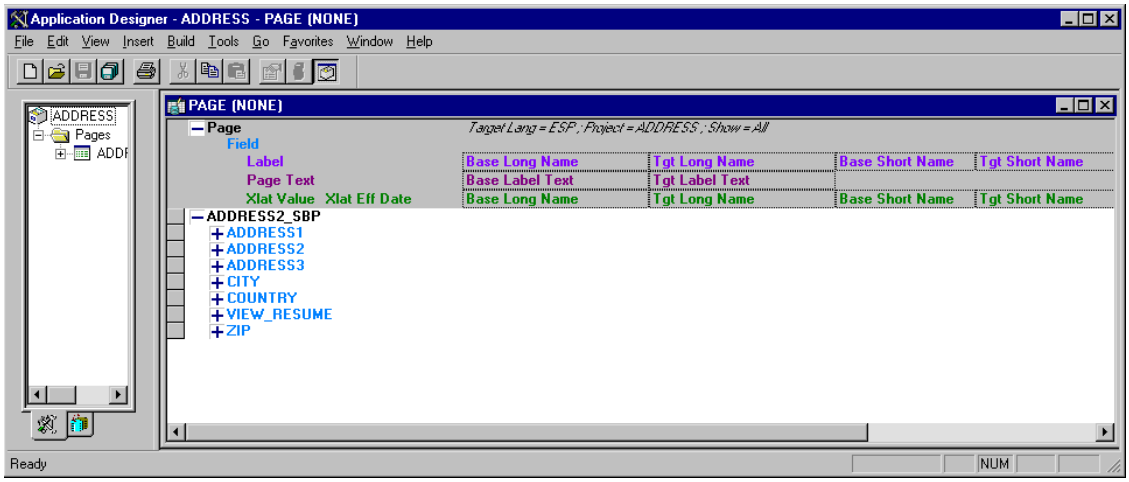
Opening Translation Designer

Translation Designer shows all the translatable definitions in your current Application Designer project, along with their base language and translated text.

Copyright © 1988, 2026, Oracle and/or its affiliates.

The grid contents are based on the Application Designer project that is open when Translation Designer is started. This association with Application Designer projects is a handy mechanism for organizing your translation effort. However, it also means that, to use Translation Designer, you must have a project open, and it must contain one or more pages. If you modify the project, Translation Designer won't pick up the new objects in the project until you save the changes and refresh.

This example illustrates the fields and controls on the Translating Page Definitions in Translation Designer.



To open Translation Designer:

1. Set your language preference to the target language.

   Because you use the Application Designer to access Translation Designer, set your language preference using PeopleSoft Configuration Manager.

2. Open Application Designer.

3. Create or open the project containing the definitions 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 definitions on the pages. If you are creating a new project, you must save it before opening Translation Designer. Keep the number of pages in the project being translated to a minimum, as all the fields and other translatable definitions on each page in the project are loaded into memory when Translation Designer starts. A project with a lot of pages may take some time to open due to the sheer volume of fields on the page. Try to keep the number of pages in a project to be translated using Translation Designer under 20.

4. Open Translation Designer.

   - To translate the pages in your project, from the Tools menu select **Translate** and choose **Translate Page Definitions**.

     This is the most versatile option because it shows you all translatable definitions 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 grid.

   - To translate only the fields in your project, from the Tools menu select **Translate** and choose **Translate Fields**.

Copyright © 1988, 2026, Oracle and/or its affiliates.

This option lets you translate fields that are directly included as definitions 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 translate values in your project, from the Tools menu select **Translate** and choose **Translate Xlats**.

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

  After you select one of the translate options, Translation Designer appears in the object workspace.

  Depending on which translation grid you opened, different translatable definitions from the active project appear in the grid. All definitions 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.

  Use standard tree controls to expand and collapse the view in order to display definitions at different levels. You can also collapse or expand all levels of the tree by right-clicking on the tree node and selecting the appropriate option from the popup menu.

5. (Optional) Arrange the window to maximize Translation Designer.

   If you want more space for the translation grid, maximize the Application Designer window and then maximize Translation Designer within the window. For even more space, hide the other frames in the Application Designer window. To toggle the display of the project workspace, from the View menu select **Project Workspace**, or press **Alt+0** (zero). To toggle the display of the output window, from the View menu select **Output Window**, or press **Alt+1**.

Related Topics

- Setting Up Language Preferences for Microsoft Windows-Based PeopleTools
  This section provides an overview of language preferences in PeopleTools for Microsoft Windows and discusses how to control PeopleSoft Configuration Manager language settings.

Using Translation Designer Display Options

This section discusses how to:

- Use the information bar.
- Expand and collapse nodes.
- Choose which rows to view.
- Resize columns and rows.

Using the Information Bar

Across the top of the translation grid, an information bar displays helpful information about the project. The target language, current project, and current viewing option (all, translated, untranslated, or modified) are set in italic.

On the left side of the bar, you can see the hierarchical organization that is used to display definitions in the grid: pages (at the highest level), fields (at the next level), and translatable definitions—field labels, page text, and translate values—at the lowest level.

Each definition is color-coded to remind you with which definition type you're working.

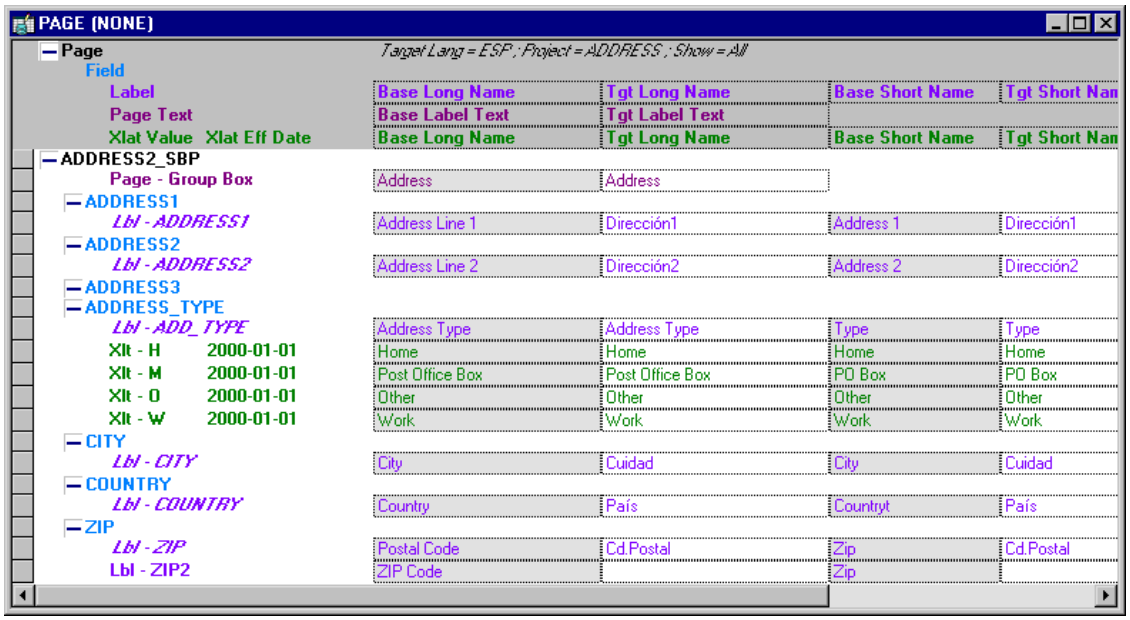
Copyright © 1988, 2026, Oracle and/or its affiliates.

- Fields: blue.
- Field labels: light purple.
- Page text (not derived from a field label): dark purple.
- Translate values: green.

Further, the following graphical cues help you understand the display:

- All fields on the page are displayed, even if they are hidden fields.
- The default label ID appears in italics, but not boldface.
- Any label ID that is used but that is not the default label appears in boldface italics.
- Page text from hyperlinks is displayed.
- If the same field has multiple label IDs on the same page, it appears only once for that page.

This example illustrates the fields and controls on the Color-coded definitions.



In this example, notice 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 translate values. All other translatable definitions are field labels.

Notice also that the default label ID is italicized but not bold, while any non-default label IDs used are in bold italics. This is apparent when you look at the ZIP field, which has two labels. The other field labels appearing in the grid are associated with the field, but not used on the page being translated; they are displayed primarily so the translator can see the context in which they are translating the label and ensure consistency of terminology across labels.

Expanding and Collapsing Nodes

The left side of the translation grid is a hierarchical tree control that displays all the definitions on the page.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Field or Control	Description
+	Click once to expand the definition by one level.
−	Click once to collapse the definition by one level.

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

Choosing Which Rows to View

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

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

Keep in mind that the options you select in the popup menu work together. If you select **Expand Current Object** while **Untranslated** is activated, you might not see anything. This means that everything in the current definition 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, drag the column divider to the desired position. You can resize only translatable columns; you cannot resize the columns that display the hierarchical tree of page definitions. Use the same technique to change row height.

Working in Translation Designer

This section discusses how to:

- Enter translations.
- Navigate in Translation Designer.
- Search and replace.
- Use Application Designer functionality.
- Integrate with other applications.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Entering Translations

The working area of Translation Designer 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).

As the column names imply, the two base language columns display the base language labels for the definitions in the grid. To translate the labels, type the translation into 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, the same translation is entered into any other occurrence of that field in the current grid. This ensures consistency and saves time.

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 definition, changes are not permanent until you save them.

> **Note**
>
> To save related language data, you must have values in all fields. You cannot partially translate a single PeopleTools definition. 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 translate values for a particular field, any untranslated values pick up the base language text, which is then saved to the related language table.

Navigating Translation Designer

Click to move to a cell or node in Translation Designer, or use the standard Microsoft Windows keyboard shortcuts:

- 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, press **TAB**.

  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 definition.
- Use the **UP ARROW** and **DOWN ARROW** keys to move around in the hierarchical tree—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 Designer. Then move up or down to the desired node. Then use the **LEFT ARROW** key to move back into the tree.
- If there are several pages in your project, scroll directly to the node for a particular page by right-clicking that page in the project workspace and from the menu select **Translate** and choose **Translate Page Definitions** from the popup menu.

Copyright © 1988, 2026, Oracle and/or its affiliates.

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**.

  To search a column you cannot edit, from the Edit menu choose **Find 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 the **Match case** check box if you want to enable case-sensitive searching.
- The **Find** option provides enables you 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, ensure 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, ensure that you're in the top row when you press **Replace**.

Using Application Designer Functionality

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

- Using the **Find Definition References** option to research where in the system a particular definition is used. To use this option, from the Edit menu select **Find Definition References**.
- Opening a translated page in order to realign translated definitions.
- Opening definitions to see the properties.

  Remember, as long as you're logged on using a non-base language, modifying labels in the definition updates the related language tables, not the base language tables.

There are several ways to open a definition:

- Use standard Application Designer functionality to open the definition from the project window.
- From the File menu, select **Open**.
- Double-click or right-click the definition within Translation Designer and select **View Definition** from the popup menu that appears.

  When you have selected page text that is not associated with a field definition, the page itself opens.

If there is a version of the page for the user's language preference, that version appears; if not, the base language page appears.

> **Note**
>
> Security controls access to different Application Designer definitions. If you cannot open a particular type of definition from within Translation Designer, you may not have the necessary level of access.

Copyright © 1988, 2026, Oracle and/or its affiliates.

See Security Administration.

Integrating with Other Applications

Because Translation Designer emulates a Microsoft Excel spreadsheet, you can use the standard Windows copy and paste tools to copy translations from Translation Designer into Excel or another Windows application. This is useful when running spell check against your translations or when saving your translations in a spreadsheet for offline review.

To copy the entire contents of the currently open Translation Designer into the clipboard:

1. Select the entire grid.

   From the Edit menu choose **Select All** or click in the top left cell of the grid border.
2. From the Edit menu select **Copy**.

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

Translating Definitions

This topic provides an overview of the translation pages and discusses how to translate them. Each section displays the object name, related language table name, base table and the location to translate in a tabular format. It is followed by the steps to translate the pages.

Understanding the Translation Process

For definitions that cannot be translated via Translation Designer, PeopleTools contains several translation pages that are especially useful if the translator has limited experience using PeopleTools. A translator who has good technical skills and who is familiar with PeopleTools may prefer to translate using the design tools as described later in this topic.

To locate the translation pages, select **PeopleTools**, and then **Translations**, and then **Translate System Definitions**. The translation pages will help you to translate system definitions, including records, menus, menu items, components, queries, and other definitions.

You should translate application definitions in the following order to ensure the least amount of duplication during your translation process:

1. Translate page objects, messages, and menu and component objects—in any order.
2. Translate business processes, PeopleSoft Process Scheduler, and portals—in any order.
3. Translate other definitions—for example, trees, queries, application data, HTML definitions, and so on—in any order.

Searching the Object Definitions Alphabetically

In PeopleTools, if you are searching through the object definitions on a control field alphabetically, enter the first few characters of the object in the control field. If you want to translate a specific object definition, enter the object definition name in the control field or use the Search button to prompt for a specific object definition.

Copyright © 1988, 2026, Oracle and/or its affiliates.

You can also leave the control field name blank to display the complete list of object definitions. You can use **Alt+5** which is the shortcut for the **Look up** button to select a specific definition.

ACE Analytic Model ID Definitions

The Translate Analytic Model page enables you to translate short and long descriptions for Analytic Calculation Engine (ACE) analytic model ID definitions into a specific target language.

The following table lists where to translate analytic model ID definitions.

Object	Base Table	Related Language Table	Where to Translate
AnalyticModelIDDefinitions	PSACEMDLDEFN	PSACEMDLDEFNLNG	Translate Analytic Model page

To translate an ACE analytic model ID definition:

1. Select **PeopleTools**, and then **Translations**, and then **Translate System Definitions**, and then **Translate Analytic Model**.

   The Translate Analytic Model page appears.
2. In the **Model ID** field, select the analytic model ID to translate.

   See Searching the Object Definitions Alphabetically for tips to search for a analytic model ID.
3. In the **Target Language** drop-down list, select the language into which to translate the record descriptions.
4. Click the **Get Model ID** button in the **Record** field, select a set of record definitions.

   A set of model ID definitions appears in the Analytic Model List based on the value you entered into the **Model ID** field.
5. Translate the short and long descriptions into the target language.

   The base language descriptions for each model ID appear on the left; enter the translations in the **Target Description** fields on the right.
6. Click the **Save** button.

Application Data

The Translate Application Data page enables you to translate system and customer data. This page does not allow you to translate managed objects.

> **Note**
>
> To use the Translate Application Data page you must be logged into the PeopleSoft Pure Internet Architecture in the base language.

Before you can translate application data, you must enable access to the Translate Application Data page. This section describes how to:

- Enable access to the Translate Application Data page.
- Translate record fields or ignore record fields for translation.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Enabling Access to the Translate Application Data Page

To enable access to the Translate Application Data page:

1. Select **PeopleTools**, and then **Security**, and then **Permissions and Roles**, and then **Permission Lists**.

   The Permissions List page appears.
2. Select a permission list that is assigned to your user ID.

   The permission list displays.
3. Select the Pages page.
4. Click the **Edit Components** link for the Translate menu item.

   The Component Permissions list displays.
5. Click the **Edit Pages** link for the Translate Application Data item.

   The Page Permissions page appears.
6. Click the **Select All** button.
7. Click the **OK** button, click the next **OK** button, and then click the **Save** button.
8. Repeat steps 2 through 7 for other users or roles that need access to the page.
9. Select **PeopleTools**, and then **Portal**, and then **Structure and Content**.
10. Click the **PeopleTools** link.
11. Click the **Translations** link.
12. Click the **Translate System Definitions** link.
13. Click the **Edit** link for the **Application Data** item.
14. Clear the **Hide in Navigation** check box.
15. Save your changes.

Translating Application Data

Before you can translate application data, you must enable access to the Translate Application Data page.

See Translating Definitions.

To translate application data:

1. Select **PeopleTools**, and then **Translations**, and then **Translate System Definitions**, and then **Application Data**.

   The Translate Application Data page appears.
2. In the **Record to Translate** field, enter the base table name.
3. From the **Target Language** drop-down list box, select the target language.
4. Optionally, click the **Filter with SQL?** link to select the record fields to translate.
5. Click Search.

   When you click **Search**, the page appears the related language record, the number of rows or values found, and the first of all key names and key values for the table. For each value, the translatable fields display in the **Base Text** column in the base language of the database. The **Lang Text** fields display translations for the fields, if they exist. Use the

Copyright © 1988, 2026, Oracle and/or its affiliates.

View All link or the **Show Previous** and **Show Next** arrows to navigate through the results retrieved.

> **Note**
>
> The following information describes how to work with all rows in a record or specific rows in a record.

- To work with all rows in a record, click **Search**.
- To select specific rows in a record, click the **Filter with SQL?** link.

  A WHERE clause page appears that provides you with an area to enter an SQL statement that will filter the record data. Enter a WHERE clause to filter the data.

  The field names that you can use in the WHERE clause display in the **Record Fields** box. To expand the list, click the **View All** link.

  After you enter a WHERE clause, click **OK** to return to the Translate Application Data page. Click **Search**.

6. Translate or ignore the record fields.

   Some related language tables contain fields that should never be translated. You can choose to translate the record fields or tell the system to ignore them for translation.

   - To translate a field, in the **Lang Text** box, enter a translation for the item in the **Base Text** field.
   - To ignore a field for translation, check the **Ignore?** box next to the **Lang Text** box. The following message displays:

     ```
     Ignoring field: Translations for this field will be overwritten with
     base⇒
      language values, proceed? (102,83)

     By ignoring this field, any translation for this field will be
     overwritten with⇒
      the base language values. Do you want to proceed?
     ```

     Click **Yes**.

     > **Note**
     >
     > This message displays the first time you check the **Ignore?** box for a translatable field. If you check this box again for the same field, the message will not display and the system copies, in the base language, the text in the Base Text field over to the Lang Text field. This text will appear in all rows in the table. Therefore, do not use the **Ignore?** option to simply copy the information in the **Base Text** field to the **Lang Text** box. To remove an entry in the **Lang Text** box made under these circumstances, clear the **Ignore?** box and delete the entry.

7. Click **Save**.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Application Engine Descriptions

The following table lists where to translate Application Engine definitions.

Object	Base Table	Related Language Table	Where to translate
Application Engine Definitions	PSAEAPPLDEFN	PSAEAPPLLANG	Translate Application Engine Programs page.

Translating Application Engine Programs

This page enables you to translate Application Engine program descriptions into a specific target language.

To translate Application Engine programs:

1. Select **PeopleTools**, and then **Translations**, and then **Translate System Definitions**, and then **Translate AE Programs**. The Translate Application Engine Programs page appears.
2. In the **Target Language** drop-down list, select the language into which to translate.
3. In the **Program** field, select an Application Engine program.

   See Searching the Object Definitions Alphabetically for tips to search for an Application Engine program.
4. Click the **Get Objects** button to bring the program name and base descriptions into the object list.

   A set of base program descriptions appears in the **Object List** field based on the value that you entered in the **Program** field.
5. 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.
6. Click the **Save** button.

See Application Engine: Application Engine Fundamentals.

BI Publisher Definitions

The following table lists the objects, base table, related language table, and tool used to translate Oracle's Business Intelligence (BI) Publisher objects:

Object	Base Table	Related Language Table	Where to Translate
BI Publisher Data Source Definitions	PSXPDATASRC	PSXPDATASRCLNG	Translate BIP Data Source Definitions page
BI Publisher Report Definitions	PSXPRPTDEFN	PSXPRPTDEFNLNG	Translate BIP Report Definitions
BI Publisher Template Definitions	PSXPTMPLDEFN	PSXPTMPLDEFNLNG	Translate BIP Template Definitions

This section discusses how to translate the *descriptions* for:

Copyright © 1988, 2026, Oracle and/or its affiliates.

- BI Publisher data source definitions.
- BI Publisher report definitions.
- BI Publisher template definitions.

In addition to translating the descriptions for these definitions, you can also translate the "translatable units" for report templates.

See BI Publisher for PeopleSoft: Maintaining Template Translations.

Translating BI Publisher Data Source Definitions

The Translate BI Publisher Data Source Definitions page enables you to translate the description for a BI Publisher data source definition into a specific target language.

To translate BI Publisher data source definitions:

1. Select **PeopleTools**, and then **Translations**, and then **Translate System Definitions**, and then **Translate BIP Data Sources.**

   The Translate BIP Data Source Definitions page appears.
2. In the **Data Source Type** field, enter the name of the BI Publisher data source type to translate.

   See Searching the Object Definitions Alphabetically for tips to search for a record.
3. In the **Target Language** drop-down list, select the language into which to translate the object definitions.
4. Click the **Get Objects** button to bring the BI Publisher data source definitions into the object list.

   A set of Oracle BI Publisher data source definitions appears in the **Data Source List** based on the value you entered in the **Data Source Type** field.
5. Translate the descriptions into the target language.

   The base language description for each object appears on the left; enter the translation in the **Target Description** field on the right.
6. Click the **Save** button.

Translating BI Publisher Report Definitions

The Translate BIP Report Definitions page enables you to translate the description for a BI Publisher report definition into a specific target language.

To translate BI Publisher report definitions:

1. Select **PeopleTools**, and then **Translations**, and then **Translate System Definitions**, and then **Translate BIP Reports.**

   The Translate BIP Report Definitions page appears.
2. In the **Report Name** field, enter the name of the Oracle BI Publisher report definition to translate.

   See Searching the Object Definitions Alphabetically for tips to search for a Oracle BI Publisher report definition.
3. In the **Target Language** drop-down list, select the language into which to translate the object definitions.
4. Click the **Get Objects** button to bring the Oracle BI Publisher report definitions into the object list.

Copyright © 1988, 2026, Oracle and/or its affiliates.

A set of BI Publisher report definitions appears in the **Report Definition List** based on the value you entered in the **Report Name** field.

5. Translate the descriptions into the target language.

   The base language description for each object appears on the left; enter the translation in the **Target Description** field on the right.

6. Click the **Save** button.

Translating BI Publisher Template Definitions

The Translate BIP Template Definitions page enables you to translate the description for a BI Publisher template definition into a specific target language.

To translate BI Publisher template definitions:

1. Select **PeopleTools**, and then **Translations**, and then **Translate System Definitions**, and then **Translate BIP Templates.**

   The Translate BIP Template Definitions page appears.

2. In the **Template ID** field, enter the name of the BI Publisher template definition to translate.

   See Searching the Object Definitions Alphabetically for tips to search for a Oracle BI Publisher template definition.

3. In the **Target Language** drop-down list, select the language into which to translate the object definitions.

4. Click the **Get Objects** button to bring the BI Publisher template definitions into the object list.

   A set of BI Publisher definitions appears in the **Template List** based on the value you entered in the **Template ID** field.

5. Translate the descriptions into the target language.

   The base language description for each object appears on the left; enter the translation in the **Target Description** field on the right.

6. Click the **Save** button.

Business Processes

Translate menus before business processes, PeopleSoft Process Scheduler, and portals because the terms are related.

The following table lists the objects, base table, related language table, and tool used to translate business processes:

Object	Base Table	Related Language Table	Where to Translate
Business Process Definitions	PSBUSPROCDEFN	PSBUSPROCLANG	Application Designer.
Business Process Items	PSBUSPROCITEM	PSBUSPROCITEMLANG	Application Designer.
Component Interfaces	PSBCDEFN	PSBCDEFNLANG	Translate Component Interfaces page.
Activities	PSACTIVITYDEFN	PSACTIVITYLANG	Translate Activity Definitions page.

Copyright © 1988, 2026, Oracle and/or its affiliates.

This section discusses how to translate:

- Business process definitions (and their related objects).
- Component Interface messages.
- Activity definitions.

> **Note**
>
> PeopleSoft Business Interlinks is no longer supported. This translation page currently exists for backward compatibility only. New integrations can be built using Integration Broker.

Translating Business Process Definitions (and Their Related Objects)

To translate business process definitions:

1. Set the language in PeopleSoft Configuration Manager.
2. Open Application Designer.
3. Open the Business Process Map.
4. Overwrite the English with the translation.
5. Save.

Translating Component Interface Messages

The Translate Component Interfaces page enables you to select a component interface and then translate the description of the component interface into a specific target language.

To translate component interface messages:

1. Select **PeopleTools**, and then **Translations**, and then **Translate System Definitions**, and then **Translate Component Interfaces**.

   The Translate Component Interfaces page appears.
2. In the **Name** field, enter a component interface.

   See Searching the Object Definitions Alphabetically for tips to search for a component interface.
3. In the **Target Language** drop-down list, select the language into which to translate the message set.
4. Click the **Get Objects** button to bring the component interface into the definition list.

   A set of component interface names appears in the Object List.
5. Translate the component interface description into the target language.
   a. Click the **Update** button to display the Add Message Set page.
   b. Enter descriptions for the message set in the target language.
   c. Click the **OK** button to accept the change and return to the Translate Component Interfaces page.
6. Click the **Save** button.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Translating Activity Definitions

The Translate Activity Definitions page enables you to translate the short and long descriptions for an activity definition into a specific target language.

To translate activity definitions:

1. Select **PeopleTools**, and then **Translations**, and then **Translate System Definitions**, and then **Translate Workflow Activities.**

   The Translate Activity Definitions page appears.
2. In the **Activity Name** field, enter the name of the activity to translate.

   See Searching the Object Definitions Alphabetically for tips to search for an object definition.
3. In the **Target Language** drop-down list, select the language into which to translate the object definitions.
4. Click the **Get Objects** button to bring the activity definitions into the object list.

   A set of activity definitions appears in the **Activity List** based on the value you entered in the **Activity Name** field.
5. Translate the short and long descriptions into the target language.

   The base language descriptions for each activity appear on the left; enter the translations in the Target Description fields on the right.
6. Click the **Save** button.

Composite Query

The following table lists objects, base tables, related language tables, and pages used to translate composite queries.

Object	Base Table	Related Language Table	Where to translate
Composite Query	PSCPQDEFN	PSCPQDEFN_LANG	Translate Composite Query page
Composite Query Binds	PSCPQBIND	PSCPQBINDLANG	Translate Composite Query page
Composite Query Field	PSCPQFIELD	PSCPQFIELDLANG	Translate Composite Query page

The Translate Composite Query page enables you to translate descriptions for Composite Query definitions into a specific target language.

To translate a Composite Query:

1. Select **PeopleTools**, and then **Translations**, and then **Translate System Definitions**, and then **Translate Composite Queries**. The Translate Composite Query page appears.
2. In the **Composite Query** field, select the Composite Query to translate.

   See Searching the Object Definitions Alphabetically for tips to search for the object definitions.
3. In the **Target Language** drop-down list, select the language into which to translate the record descriptions.

Copyright © 1988, 2026, Oracle and/or its affiliates.

4. Click the **Get Objects** button to display the Composite Query objects under the Composite Query List.

   A set of Composite Query appears in the Composite Query List based on the value you entered into Composite Query field.

5. Translate the description into the target language.

   The base-language descriptions for each Composite Query appear on the left; enter the translations in the **Target Description** fields on the right.

6. Click the **Show Item Details** button to display the Composite Query Headings page.

   Translate the Composite Query Field Headings and the Composite Query Prompts and then click the **OK** button to accept the change and return to the Translate Composite Query page.

7. Click the **Save** button.

Connected Query Definitions

The following table lists the objects, base table, related language table, and tool used to translate connected query definitions:

Object	Base Table	Related Language Table	Where to Translate
Connected Query Definitions	PSCONQRSDEFN	PSCONQRSDEFNLAN	Translate Connected Query Definitions page

The Translate Connected Query Definitions page enables you to translate the short and long descriptions for a connected query definition into a specific target language.

To translate connected query definitions:

1. Select **PeopleTools**, and then **Translations**, and then **Translate System Definitions**, and then **Translate Connected Queries.**

   The Translate Connected Query Definitions page appears.

2. In the **Connected Query** field, enter the name of the connected query.

   See Searching the Object Definitions Alphabetically for tips to search for a connected query.

3. In the **Target Language** drop-down list, select the language into which to translate the object definitions.

4. Click the **Get Objects** button to bring the connected query definitions into the object list.

   A set of connected query definitions appears in the **Connected Query List** based on the value you entered in the **Connected Query** field.

5. Translate the short and long descriptions into the target language.

   The base language descriptions for each connected query appear on the left; enter the translations in the Target Description fields on the right.

6. Click the **Save** button.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Data Set Definitions

The following table lists objects, base tables, related language tables, and pages used to translate data set definitions.

Object	Base Table	Related Language Table	Where to translate
Data Set Definition	PSADSDEFN	PSADSDEFNLANG	Translate Data Set Definitions page
Data Set Group	PSADSGROUP	PSADSGROUPLANG	Translate Data Set Definitions page
Data Set Relation	PSADSRELATION	PSADSRELLANG	Translate Data Set Definitions page

The Translate Data Set page enables you to translate descriptions for Data Set definitions into a specific target language.

To translate a Data Set Definition:

1. Select **PeopleTools**, and then **Translations**, and then **Translate System Definitions**, and then **Translate Data Set Definitions**. The Translate Data Set Definitions page appears.
2. In the **Data Set Name** field, select the Data Set Name to translate.

   See Searching the Object Definitions Alphabetically for tips to search for the object definitions.
3. In the **Target Language** drop-down list, select the language into which to translate the record descriptions.
4. Click the **Get Objects** button to bring a set of Data Set objects into the Data Set List group box based on the value you entered into **Data Set Name** field.
5. Translate the description into the target language. The base-language descriptions for each Data Set appear on the left; enter the translations in the Target Description fields on the right.
6. Translate Data Set Group.

   Click the **Show ADS Group** link to go to the Translate Data Set Group page. **Show ADS Group** link is displayed if the Data Set has a Data Set Group.

   The base-language description and name for each Data Set Group appear on the left; enter the translations in the **Target Description** fields on the right.
7. Translate Data Set Relation.

   Click the **Show ADS Relation** link to go to the Translate Data Set Relation page. **Show ADS Relation** link is shown if the Data Set has Data Set Relation.

   The base-language descriptions for each Data Set Relation appear on the left; enter the translations in the **Target Description** fields on the right.
8. Click the **Save** button.

Feed Definitions

The following table lists the objects, base tables, related language tables, and tools used to translate feed definitions and other associated feed objects:

Copyright © 1988, 2026, Oracle and/or its affiliates.

Object	Base Table	Related Language Table	Where to Translate
Feed Definitions	PSFP_FEED	PSFP_FEED_LANG	Translate Feed Definitions page
Feed Data Source Parameters	PSFP_PARMS	PSFP_PARMS_LANG	Translate Feed Definitions page
Feed Parameter Valid Values	PSFP_PVALS	PSFP_PVALS_LANG	Translate Feed Definitions page
Feed Attributes	PSFP_ATTRS	PSFP_ATTRS_LANG	Translate Feed Definitions page
Feed Categories	PSFP_CATEGORY	PSFP_CATG_LANG	Translate Feed Categories page
Feed Data Types	PSFP_DATATYPE	PSFP_DTYPE_LANG	Translate Feed Data Types page
Default Feed Attributes	PSFP_DTYPE_ATTR	PSFP_DATTR_LANG	Translate Feed Data Types page

This section discusses how to translate:

- Feed definitions
- Feed categories
- Feed data types

Translating Feed Definitions

The Translate Feed Definitions page enables you to translate the short and long descriptions for feed definitions, feed data source parameters, feed parameter valid values and feed attributes into a specific target language.

To translate feed definitions:

1. Select **PeopleTools**, and then **Translations**, and then **Translate System Definitions**, and then **Translate Feed Definitions**.

   The Translate Feed Definitions page appears.

2. In the **Feed ID** field enter the name of the field ID.

   See Searching the Object Definitions Alphabetically for tips to search for the object definitions.

3. In the **Target Language** drop-down list, select the language into which to translate the object definitions.

4. Click the **Get Objects** button to bring the feed definitions into the object list.

   A set of feed definitions appears in the Feed List based on the value you entered in the **Feed ID** field.

5. Translate the descriptions into the target language.

   The base language description for each object appears on the left; enter the translation in the **Target Description** field on the right.

6. If any exist, translate the feed data source parameters.

   Click the **Feed Data Source Parameters** link.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Translate the feed data source parameters and then click **OK** to accept the change and return to the Translate Feed Definitions page.

7. If any exist, translate the feed parameter valid values.

   Click the **Feed Parameter Valid Values** link.

   Translate the Feed Parameter Valid Values and then click **OK** to accept the change and return to the Translate Feed Definitions page.

8. If any exists translate the feed attributes.

   Click the **Feed Attributes** link.

   Translate the Feed Attributes and then click OK to accept the change and return to the Translate Feed Definitions page.

9. Click the **Save** button.

Translating Feed Categories

The Translate Feed Categories page enables you to translate the short and long descriptions for feed categories into a specific target language.

To translate feed categories:

1. Select **PeopleTools**, and then **Translations**, and then **Translate System Definitions**, and then **Translate Feed Categories.**

   The Translate Feed Categories page appears.

2. In the **Category ID** field, enter the name of the Category ID to translate.

   See Searching the Object Definitions Alphabetically for tips to search for the object definitions.

3. In the **Target Language** drop-down list, select the language into which to translate the object definitions.

4. Click the **Get Objects** button to bring the feed category into the object list.

   A set of feed categories appears in the Feed Categories List based on the value you entered in the **Category ID** field.

5. Translate the short and long descriptions into the target language.

6. Click the **Save** button.

Translating Feed Data Types

The Translate Feed Data Types page enables you to translate the short and long descriptions for feed data types, and default feed attributes into a specific target language.

To translate feed data types:

1. Select **PeopleTools**, and then **Translations**, and then **Translate System Definitions**, and then **Translate Feed Data Types.**

   The Translate Feed Data Types page appears.

2. In the **Data Type** field, enter the name of the content reference to translate.

   See Searching the Object Definitions Alphabetically for tips to search for the object definitions.

3. In the **Target Language** drop-down list, select the language into which to translate the object definitions.

Copyright © 1988, 2026, Oracle and/or its affiliates.

4. Click the **Get Objects** button to bring the feed data types into the object list.

   A set of feed data types appears in the Feed Data Type List based on the value you entered in the **Data Type** field.

5. Translate the short and long descriptions into the target language.
6. Translate the default feed attributes.

   Click the **Default Feed Attributes** link.

   Translate the Default Feed Attributes and then click **OK** to accept the change and return to the Translate Feed Data Types page.

7. Click the **Save** button.

Integration Broker Definitions

The following table lists the objects, base table, related language table, and tool used to translate Integration Broker objects:

Object	Base Table	Related Language Table	Where to Translate
Message Definitions	PSMSGDEFN	PSMSGDEFNLANG	Translate Message Definitions page
Queues	PSQUEUEDEFN	PSQUEUEDEFNLANG	Translate Queues page
Routings	PSIBRTNGDEFN	PSRTNGDFNLANG	Translate IB Routings page
Service Operation Handlers	PSOPRHDLR	PSOPRHDLRLANG	Translate Service Operation Handlers page
Service Operations	PSOPERATION	PSOPERATIONLANG	Translate Service Operations page
Service Operation Versions	PSPRSMATTR	PSPRSMATTRLANG	Translate Service Operation Versions page
Services	PSSERVICE	PSSERVICELANG	Translate IB Services page

PeopleTools features one page for each Integration Broker object that you can use to translate object descriptions into a specific target language.

> **Note**
>
> The following steps for translating Integration Broker objects apply to all of the objects in the preceding table.

To translate a Integration Broker object:

1. Select **PeopleTools**, and then **Translations**, and then **Translate System Definitions.**

   The Translate System Definitions menu appears.

2. Select one of the following menu items to translate the corresponding Integration Broker object descriptions:

Copyright © 1988, 2026, Oracle and/or its affiliates.

- Message Definitions
- Queues
- Routings
- Service Operation Handlers
- Service Operations
- Service Operation Versions
- Services

The corresponding translation page appears.

3. Based on the object descriptions that you are translating, do the following:

Object Descriptions to Translate	Action
Message Definitions	In the **Message** field, enter the name of the message definition to translate or use the Look up button (Alt+5) to select a specific definition.
Queues	In the **Queue Name** field, enter the name of the queue to translate or use the Look up button (Alt+5) to select a specific definition.
Routings	In the **Routing Name** field, enter the name of the routing to translate or use the Look up button (Alt+5) to select a specific definition.
Service Operation Handlers	In the **Operation** field, enter the name of the service operation handler to translate or use the Look up button (Alt+5) to select a specific definition.
Service Operations	In the **Operation** field, enter the name of the service operation to translate or use the Look up button (Alt+5) to select a specific definition.
Service Operation Versions	In the **Operation** field, enter the name of the service operation that contains the version to translate or use the Look up button (Alt+5) to select a specific definition.
Services	In the **Service** field, enter the name of the service to translate or use the Look up button (Alt+5) to select a specific definition.

> **Note**
>
> Leave the field blank if you want to select all object definitions in the system.

4. In the **Target Language** drop-down list, select the language into which to translate the object definitions.
5. Click the **Get Objects** button to bring the object definitions into the object list.

Copyright © 1988, 2026, Oracle and/or its affiliates.

> **Note**
>
> If no descriptions exist in the base language, no results are returned.

6. Translate the descriptions into the target language.

   The base language description for each object appears on the left; enter the translations in the **Target Description** fields on the right.

7. Click the **Save** button.

Integration Groups

The following table lists the objects, base tables, related language tables, and tool used to translate integration groups:

Object	Base Table	Related Language Table	Where to Translate
Integration Group	PSIBGROUPDEFN	PSIBGROUPLANG	Translate Integration Groups page
Integration Subgroup	PSIBGROUPTRAN	PSIBGRPTRNLANG	Translate Integration Groups page

The Translate Integration Groups page enables you to translate the short and long descriptions for integration groups and subgroups into a specific target language.

To translate integration groups:

1. Select **PeopleTools**, and then **Translations**, and then **Translate System Definitions**, and then **Translate Integration Groups.**

   The Translate Integration Groups page appears.

2. In the **Integration Group Name** field, enter the name of the Integration Group Name to translate.

   See Searching the Object Definitions Alphabetically for tips to search for the object definitions.

3. In the **Target Language** drop-down list, select the language into which to translate the object definitions.

4. Click the **Get Objects** button to bring the integration group into the object list.

   A set of integration groups and integration subgroups appears in the Integration Group List and Integration Subgroup List based on the value you entered in the **Integration Group Name** field.

5. Translate the descriptions into the target language.

   The base language description for each object appears on the left; enter the translation in the **Target Description** field on the right.

6. Click the **Save** button.

Menu and Component Objects

The following table lists the objects, base tables, related language tables, and tools used to translate menu and component objects.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Object	Base Table	Related Language Table	Where to Translate
Menu Definitions	PSMENUDEFN	PSMENUDEFNLANG	Translate Menu page.
Menu Items	PSMENUITEM	PSMENUITEMLANG	Translate Menu Item page.
Component Descriptions	PSPNLGRPDEFN	PSPNLGDEFNLANG	Translate Components page.
Components	PSPNLGROUP	PSPNLGROUPLANG	Translate Components page.

This section discusses how to:

- Translate menu definitions.
- Translate menu items.
- Translate folder tabs in Application Designer.
- Translate folder tab labels using the translation utilities.

Translating Menu Definitions

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

> **Note**
>
> In a portal, the names that appear in the menu are actually content references. You must translate these using the Translate Menu Items page.

To translate menus definitions:

1. Select **PeopleTools**, and then **Translations**, and then **Translate System Definitions**, and then **Translate Menu Definitions**. The Translate Menu page appears.
2. In the **Menu Name** field, select the menu.

   See Searching the Object Definitions Alphabetically for tips to search for a menu.
3. In the **Target Language** drop-down list, select the language into which you want to translate the menu definition.
4. Click the **Get Menu** button to bring the menu descriptions into the 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 PeopleTools development environment 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. Click the **Save** button.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Translating Menu Items

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

To translate menu items:

1. Select **PeopleTools**, and then **Translations**, and then **Translate System Definitions**, and then **Translate Menu Items**. The Translate Menu Items page appears.
2. In the **Menu Name** field, select the menu.

   See Searching the Object Definitions Alphabetically for tips to search for a menu.
3. In the **Target Language** drop-down list, select the language into which you want to translate the menu definition.
4. Click the **Get Menu Items** button to bring the menus into the Menu Item List.
5. 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.

   The translated menu bar label appears at runtime if the user's language preference is set to the target language. It will also appear if one or more menu items that appear in the menu bar have a menu bar label translated into the target language.

   To include accelerator keys for use with PeopleTools development environment 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 must select whether to use the same translation for all occurrences of the bar label:

   - Click the *Yes* button if you want the translation you just entered to be 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 click the **Save** button.
   - Click the *No* button if you do not want to translate all the menu items.
6. Click the **Save** button.

Translating Tile Title

You can translate the title or name of a tile using the Tile Wizard.

For more information, see Portal Technology: Using Tile Wizard.

Translating Folder Tabs in Application Designer

Folder tabs should be translated directly in the Translate Components page. However, if you are familiar with design tools you can also translate folder tabs directly in the Application Designer, Translation Designer.

Component definitions include two labels for each page in the component: an item label and a folder tab label. In the PeopleSoft Pure 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.

To translate folder tab labels in Application Designer:

1. Change the PeopleSoft Configuration Manager language preference setting to the target language.

Copyright © 1988, 2026, Oracle and/or its affiliates.

2. Log in to the PeopleSoft system.
3. Open Application Designer.
4. Open the component definition.
5. Translate the item labels and folder tab labels into the target language.

   Edit the labels directly 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 appears in the folder tab.

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

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

Translating Folder Tab Labels Using Translation Utilities

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 the portal, only one of these labels appears 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 Microsoft Windows, 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.

To translate folder tab labels using the translation utilities:

1. Select **PeopleTools**, and then **Translations**, and then **Translate System Definitions**, and then **Translate Components**.

   The Translate Components page appears.
2. In the **Component Name** field, select a set of component definitions.

   See Searching the Object Definitions Alphabetically for tips to search for a component definitions.
3. In the **Target Language**drop-down list, select the language into which to translate the component labels.
4. Click the **Get Component** button to bring the components into the component 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 PeopleTools development environment 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. Click the **Save** button.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Messages

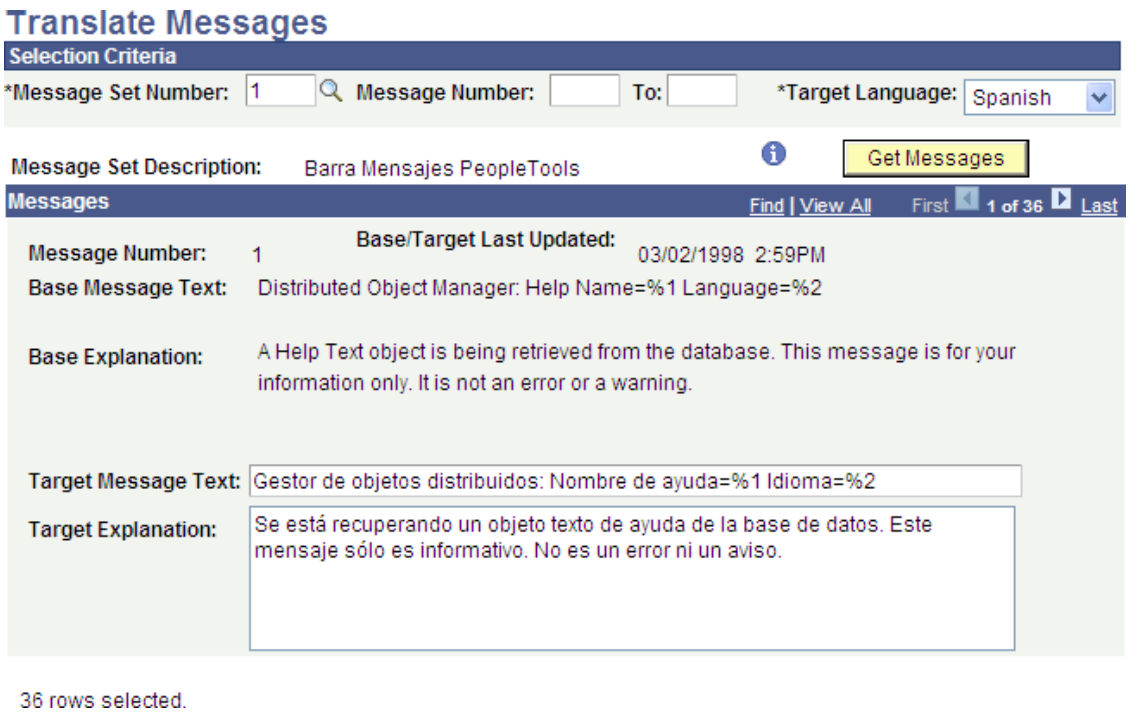
The following table lists the objects, base tables, related language tables, and tools used for translating message set descriptions and messages.

Object	Base Table	Related Language Table	Where to Translate
Message Set Description	PSMSGSETDEFN	PSMSGSETLANG	Translate Messages page.
Messages	PSMSGCATDEFN	PSMSGCATLANG	Translate Messages page.
Message Explanations	PSMSGCATDEFN	PSMSGCATLANG	Translate Messages page.

The Translate Messages page enables you to translate message set descriptions, messages, and detailed message explanations into a specific target language.

Access the Translate Messages page (**PeopleTools**, and then **Translations**, and then **Translate System Definitions**, and then **Translate Message Sets**).

This example illustrates the fields and controls on the Translate Messages page.



To translate messages:

1. Select a message set with which to work.

   Enter a **Message Set Number** in the field or click the **Prompt** button to search for a specific message set number.

2. Select messages within the message set with which to work.

Copyright © 1988, 2026, Oracle and/or its affiliates.

a. To select all messages within the message set, leave the **Message Number** field blank.

b. To select a range of messages within the message set, enter the range in the **Message Number** fields.

3. Select the target language.

   a. From the **Target Language** drop-down list, select a target language.

   b. Click the **Get Messages** button.

      The message set description you selected appears under the **Message Set Number**.

      The first message in the set or in the range you selected displays in the Messages section. The **Base Message Text** and **Base Explanation** fields display the message and its explanation in the base language.

      Use the arrow buttons or the **First** and **Last** links on the Messages bar to navigate to the messages retrieved.

4. Translate the message set description into the target language.

   a. Click the **Information** icon next to the **Get Messages** button. The Add Message Set page appears.

   b. Enter a **Description** in the target language for the message set.

   c. Enter a **Short Description** for the message set in the target language. The short description must be 10 characters or less.

   d. Click **OK** to return to the Translate Messages page.

5. Translate the message text and explanation into the target language.

   a. In the **Target Message Text** field, enter a translation in the target language for the information in the **Base Language Text** field.

   b. In the **Target Explanation** field, enter a translation in the target language for the information that appears in the **Base Explanation** field.

6. Click **Save**.

7. Repeat steps 4 to 6 for each message in the message set.

   On the Messages bar, use the **Forward** or **Back** arrows or the **First** and **Last** links to navigate to other messages within the set to translate.

Page Objects

All page objects except the HTML definitions can be translated directly in Translation Designer. However, you can also translate page definitions, fields, and translate values directly in Application Designer if you are familiar with the designer tools. The following table lists the objects, base tables, related language tables, and tools used to translate page objects.

Definition	Base Table	Related Language Table	Where to Translate
Field labels	PSDBFLDLABL	PSDBFLDLABLLANG	Translation Designer
Translate values	PSXLATITEM	PSXLATITEMLANG	Translation Designer
Page definitions	PSPNLDEFN	PSPNLDEFNLANG	Translation Designer
Page fields (page text)	PSPNLFIELD	PSPNLFIELDLANG	Translation Designer
Page buttons	PSPNLBTNDATA	PSPNLBTNLANG	Translation Designer

Copyright © 1988, 2026, Oracle and/or its affiliates.

Definition	Base Table	Related Language Table	Where to Translate
Page html	PSPNLHTMLAREA	PSPNLHTMLLANG	Application Designer
Page grid/scroll area (summary text)	PSPNLCNTRLDATA	PSPNLCTLDATALNG	Application Designer

> **Note**
>
> Don't be confused by the term *translate value* or *translate table.* The Translate table (PSXLATITEM) is a common prompt table used throughout PeopleTools and PeopleSoft applications that is used to store values for fields that don't need individual prompt tables of their own. The Translate table is not used for translation purposes.

Translating Translate Values

To translate values from the Translate table in Application Designer:

1. Change the PeopleSoft Configuration Manager language preference to the target language.
2. Open Application Designer.
3. Open the field whose translate values you want to access.
4. Select **File**, and then **Definition Properties**.
5. Select the Translate Values tab in the Field Properties dialog box.
   a. Click the **Properties** icon (or right-click in the field definition and choose **Field Properties**).
   b. Click the Translate Values tab (this tab is visible only for fields with a length of four or less).

      If the descriptions have not been translated, they appear in the base language.
6. Translate the descriptions into the target language.
   a. For each translate value, click the **Change** button to display the Change Translate Table dialog box.
   b. Translate the **Long Name** and **Short Name** fields.

      Don't change any other fields.
   c. Click **OK** to accept the changes and close the dialog box.
7. Click **OK** to close the Field Properties dialog box.
8. Save the field.

PeopleSoft Process Scheduler Objects

PeopleSoft Process Scheduler has three language-sensitive definitions:

- Process type definitions.
- Job definitions.
- Process definitions.

Copyright © 1988, 2026, Oracle and/or its affiliates.

The following table lists where to translate PeopleSoft Process Scheduler objects:

Object	Base Table	Related Language Table	Where to Translate
Process Definitions	PRCSDEFN	PRCSDEFNLANG	Translate Process Definitions page.
Process Types	PRCSTYPEDEFN	PRCSTYPEDEFNLNG	PeopleTools, Process Scheduler, Process Scheduler Process Types.
Process Jobs	PRCSJOBDEFN	PRCSJOBDEFNLANG	PeopleTools, Process Scheduler, Process Scheduler Jobs.
Process Recurrences	PRCSRECUR	PRCSRECURLANG	PeopleTools, Process Scheduler, Process Scheduler Recurrences.

To translate a PeopleSoft Process Scheduler Types, Jobs and Recurrences definitions:

1. Sign in to the PeopleSoft system using the target language.
2. Select **PeopleTools**, and then **Process Scheduler** to open PeopleSoft Process Scheduler.
3. Open the definition to translate.
   - To open a process type definition, select **Process Type**.
   - To open a job definition, select **Jobs**.
   - To open a process definition, select **Recurrences**.
4. Enter the translation of the description in target language in the **Description** field.
5. Click the **Save** button.

Translating Process Definitions

The Process Definitions page enables you to select a process name and type and then translate the long descriptions of the Process Definitions into a specific target language.

To translate process definitions:

1. Select **PeopleTools**, and then **Translations**, and then **Translate System Definitions**, and then **Translate Process Definitions**.

   The Translate Process Definitions page appears.
2. In the **Target Language** drop-down list, select the language into which to translate the Process Definitions.
3. Enter **Process Name** and process type into the appropriate fields.

   Click the **Search** button to prompt for a list of names.
4. Click the **Get Objects** button to bring the Process Definitions into the definition list.

   A set of process definitions appears in the Object List.
5. Translate the long and short descriptions of process definitions into the target language.
6. Click the **Save** button.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Portals

The following table lists where to translate PeopleSoft portal objects:

Object	Base Table	Related Language Table	Where to Translate
Portal Structures	PSPRSMDEFN	PSPRSMDEFNLANG	Translate Portal Objects page.
Portal Attribute Values	PSPRSMATTRVAL	PSPRSMATTRVALNG	Translate Portal Objects page.
Portal Attributes	PSPRSMATTR	PSPRSMATTRLANG	Translate Portal Objects page.

The Translate Portal Objects page enables you to select a portal objects name and portal reference type and then translate the label and description of the portal into a specific target language.

To translate portal objects:

1. Select **PeopleTools**, and then **Translations**, and then **Translate System Definitions**, and then **Translate Portal Objects**. The Translate Portal Objects page appears.
2. In the **Target Language** drop-down list, select the language into which to translate the Portal Objects.
3. Enter the portal object name in the **Name** field. Enter the **Portal Reference Type** into the appropriate fields.

   Click the **Search** button to prompt for a list of names.
4. Click the **Get Objects** button to bring the **Portal Objects** into the definition list.

   A set of portal objects appears in the Objects List.
5. Translate the Portal Objects label into the target language.

   Enter descriptions for the portal objects in the target language.
6. Click the **Save** button.

PeopleSoft Search Definitions

The following table lists the objects, base table, related language table, and tool used to translate PeopleSoft Search Framework object:

Object	Base Table	Related Language Table	Where to Translate
PTSF Search Definitions	PSPTSF_SD	PSPTSF_SD_LANGG	Translate PTSF Search Definitions page
PTSF Search Categories	PSPTSF_SRCCAT	PSPTSF_CAT_LANG	Translate PTSF Search Categories page

This section discusses how to translate:

- PTSF Search Definitions.
- PTSF Search Categories.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Translating PTSF Search Definitions

The Translate PTSF Search Definitions page enables you to translate the description, the display title, the keywords and the display body into a specific target language.

To translate PTSF search definitions:

1. Select **PeopleTools**, and then **Translations**, and then **Translate System Definitions**, and then **Translate Search Definitions.**

   The Translate PTSF Search Definitions page appears.
2. In the **Search Definition** field, enter the name of the Search Definition to translate. See Searching the Object Definitions Alphabetically for tips to search for the object definitions.
3. In the **Target Language** drop-down list, select the language into which to translate the object definitions.
4. Click the **Get Objects** button to bring the search definition into the object list.

   A set of search definition appears in the PTSF Search Definition List based on the value you entered in the **Search Definition** field.
5. Translate the descriptions into the target language.

   The base language description for each object appears on the left; enter the translation in the **Target Description** field on the right.
6. Click the **Save** button.

Translating PTSF Search Categories

The Translate PTSF Search Categories page enables you to translate the description for a PTSF Search Categories into a specific target language.

To translate PTSF search categories:

1. Select **PeopleTools**, and then **Translations**, and then **Translate System Definitions**, and then **Translate Search Categories.**

   The Translate PTSF Search Categories page appears.
2. In the **Source Category Name** field, enter the name of the Source Category to translate.

   See Searching the Object Definitions Alphabetically for tips to search for the object definitions.
3. In the **Target Language** field, select the language into which to translate the object definitions.
4. Click the **Get Objects** button to bring the Search Category into the PTSF Search Category list.

   A set of PTSF search category appears in the PTSF Search Category List based on the value you entered in the **Source Category Name** field.
5. Translate the description into the target language.
6. Click the **Save** button.

Records

The following table lists where to translate record definitions:

Copyright © 1988, 2026, Oracle and/or its affiliates.

Object	Base Table	Related Language Table	Where to Translate
Record Definitions	PSRECDEFN	PSRECDEFNLANG	Translate Records page.

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

To translate record descriptions:

1. Select **PeopleTools**, and then **Translations**, and then **Translate System Definitions**, and then **Translate Records**. The Translate Records page appears.
2. In the **Target Language** drop-down list, select the language into which to translate the record descriptions.
3. In the **Record** field, select a set of record definitions.

   See Searching the Object Definitions Alphabetically for tips to search for a 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** field based on the value you entered into the **Record** field.
5. Translate the short and long descriptions into the target language.

   The base language descriptions for each record appear on the left; enter the translations in the fields on the right.
6. Click the **Save** button.

Related Content Definitions

The following table lists the objects, base table, related language table, and tool used to translate Related Content Service definitions:

Object	Base Table	Related Language Table	Where to Translate
Related Content Service Definitions	PSPTCSSRVDEFN	PSPTCSSRVDEFNLG	Translate Related Content Definitions page
Related Content Service Parameters	PSPTCS_PARAMS	PSPTCS_PARAMSLG	Translate Related Content Definitions page
Related Content Configurations	PSPTCSSRVCONF	PSPTCSSRVCONFLG	Translate Related Content Configurations page
Related Content Layouts	PSPTCS_MNUFLDRS	PSPTCSMNUFLDRLG	Translate Related Content Layouts page

This section discusses how to translate:

- Related content service definitions.
- Related content service configurations.
- Related content layouts.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Translating Related Content Definitions

The Translate Related Content Definitions page enables you to translate the description for related content definitions, the service help text and the service parameters into a specific target language.

To translate related content definitions:

1. Select **PeopleTools**, and then **Translations**, and then **Translate System Definitions**, and then **Translate RC Services.**

   The Translate Related Content Definitions page appears.

2. In the**Service ID** field, enter the name of the Service ID to translate.

   See Searching the Object Definitions Alphabetically for tips to search for the object definitions.

3. In the **Target Language** field, select the language into which to translate the object definitions.

4. Click the **Get Objects** button to bring the related content definitions into the object list.

   A set of related content definitions appears in the Related Content Service List based on the value you entered in the Service ID field.

5. Translate the descriptions into the target language.

   The base language description for each object appears on the left; enter the translation in the **Target Description** field on the right.

6. Translate the related content service parameters

   Click the **Show Service Parameters** button on the left of the **Service ID** field.

   Translate the service parameters and then click **OK** to accept the change and return to the Translate Related Content Definitions page.

7. Click the **Save** button.

Translating Related Content Configurations

The Translate Related Content Configurations page enables you to translate the service label for a related content service configurations into a specific target language.

To translate related content configurations:

1. Select **PeopleTools**, and then **Translations**, and then **Translate System Definitions**, and then **Translate RC Configurations.**

   The Translate Related Content Configurations page appears.

2. In the **Content Reference** field, enter the name of the content reference to translate.

   See Searching the Object Definitions Alphabetically for tips to search for the object definitions.

3. In the **Target Language** field, select the language into which to translate the object definitions.

4. Click the **Get Objects** button to bring the related content configurations into the object list.

   A set of related content configurations appears in the Service Label List based on the value you entered in the **Content Reference** field.

5. Translate the service labels into the target language.

Copyright © 1988, 2026, Oracle and/or its affiliates.

6. Click the **Save** button.

Translating Related Content Layouts

The Translate Related Content Layouts page enables you to translate the folder label for related content layouts into a specific target language.

To translate related content layouts:

1. Select **PeopleTools**, and then **Translations**, and then **Translate System Definitions**, and then **Translate RC Layouts.**

   The Translate Related Content Layouts page appears.

2. In the **Content Reference** field, enter the name of the content reference to translate.

   See Searching the Object Definitions Alphabetically for tips to search for the object definitions.

3. In the **Target Language** field, select the language into which to translate the object definitions.

4. Click the **Get Objects** button to bring the related content layouts into the object list.

   A set of related content layouts appears in the Folder Label List based on the value you entered in the Content Reference field.

5. Translate the folder labels into the target language.

6. Click the **Save** button.

Review Page Text

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

PeopleSoft also provides a Page Text inquiry page that identifies override text in pages to help you identify areas where you may have missed translating. You can translate the override text in the target-language page definition in Application Designer.

To check for override text in pages:

1. Select **PeopleTools**, and then **Translations**, and then **Translate System Definitions**, and then **Review Pages for Translation**. The Review Page Text page appears. The Review 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. In the **Page Name** field, select a set of page names.

   See Searching the Object Definitions Alphabetically for tips to search for a page.

3. In the **Target Language** drop-down list, select the target language of the page set.

   For example, if the Page Name field contains *B,* and the **Target Language** field contains *French,* the page set will consist of page definitions that begin with *B* and whose target language is *French.*

4. Click the **Get Page Text** button to display the overridden page text.

Copyright © 1988, 2026, Oracle and/or its affiliates.

A set of page definitions appears based on the settings in the **Page Name** and **Target Language** 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 has not been translated, use Translation Designer to do the translation.

See Using Translation Designer.

5. Click the **Save** button.

Queries

The following table lists where to translate queries:

Object	Base Table	Related Language Table	Where to Translate
Query Definitions	PSQRYDEFN	PSQRYDEFNLANG	Translate Queries page.
Query Fields	PSQRYFIELD	PSQRYFIELDLANG	Translate Queries page.
Query Binds	PSQRYBIND	PSQRYBINDLANG	Translate Queries page.

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

To translate queries using the Translate Query utility:

1. Select **PeopleTools**, and then **Translations**, and then **Translate System Definition**, and then **Translate Queries**. The Translate Query page appears.
2. In the **Target Language** drop-down list, select the language into which to translate the query descriptions and labels.
3. In the **Query Name** field, select a query.

   See Searching the Object Definitions Alphabetically for tips to search for a queries.
4. Click the **Get Query** button to bring the query descriptions into the query list.

   A set of query descriptions appears in the field **Query List** based on the value that you entered in the **Query Name** field.
5. 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.
6. Translate query field headings and prompt names.

   Click the **Show Item Details** button to display the 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 appears any field heading 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.

Copyright © 1988, 2026, Oracle and/or its affiliates.

7. Click the **Save** button.

Related Topics

- Query

Strings

The following table lists where to translate strings:

Object	Base Table	Related Language Table	Where to translate
Strings	STRINGS_TBL	STRINGS_LNG_TBL	Translate Strings page.

This example illustrates the fields and controls on the Translate Strings page. You can find definitions for the fields and controls later on this page.

Translating String Definitions into a Non-Base Language

To translate string definitions into a non-base language:

1. Select **PeopleTools**, and then **Translations**, and then **Translate System Definitions**, and then **Strings**. The Translate Strings page appears.
2. In the **Program ID** field, enter or search for a program ID.
3. From the **Target Language** drop-down list, select the target language.
4. Click the **Find Strings** button.

   The page refreshes and displays the first of all strings for the program ID.

   Click the **View All** link or use the **Show Next** and **Show Previous** arrows to navigate through the results.
5. In the **String Text** field, enter the a translation for the string that displays directly above the field.

   There are three situations in which you can translate strings IDs.

   - The string ID matches a field name in the database whose labels have been translated. In this case, you can use the Translate Strings page to choose among any of the labels of that field or static text for the string label. If none of the labels is appropriate for this context (or they are too long) then you can select Text as the label type and enter the translation. It is preferable that, whenever possible, you use field labels.

Copyright © 1988, 2026, Oracle and/or its affiliates.

- The String ID matches a field name in the database whose labels have *not* been translated. In this case, the message “Untranslatable Field” displays in red. This message means that there is a field in the database with the same name as the string ID, but it has not been translated, so you cannot use the translated labels for the string. In this situation, you must translate the field first and then return to the Translate Strings page and select the label that is more appropriate for the string.
- The string ID does not match a field name in the database. In this case, you may not select a label type. Text is the only option.

6. Click the **Save** button.

Related Topics

- Using the Strings Table for Language-Sensitive Text in Reports
  The PeopleTools Strings table (STRINGS_TBL) stores text strings used for language-sensitive labels and other text for PS/nVision and SQR reports. This avoids hard-coding labels into the report files themselves. The use of strings rather than hard-coded text in reports enables translators to translate the report layout in the database without editing 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 and report logic.

Time Zone Labels

There are two ways to translate time zone labels in PeopleTools:

- Using the Translate Application Data page.
- Using the Time Zone IDs page.

Translating Time Zone Labels Using the Translate Application Data Page

To translate time zone labels using the Translate Application Data page.

1. Ensure that the Translate Application Data page has been enabled.

   See Application Data.
2. Select **PeopleTools**, and then **Translations**, and then **Translate System Definitions**, and then **Application Data**.
3. In the Record to Translate field, enter *PSTIMEZONE*.
4. From the Target Language drop-down list box, select the language into which to translate the time zone label.
5. Click the **Save** button.

Translating Time Zone Labels Using the Time Zone IDs Page

To translate time zone labels using the Time Zone IDs page:

1. Sign into the PeopleSoft system using the target language.

   > **Note**
   >
   > Alternatively, if you have enabled multi-language entry, then you can select the target language in the Data Language drop-down list box on the Time Zone IDs page.

2. Select **PeopleTools**, and then **Utilities**, and then **International**, and then **Time Zones**.

Copyright © 1988, 2026, Oracle and/or its affiliates.

The Time Zone IDs page appears.

3. Locate a time zone with which to work from the list.
4. In the Description field, enter a translated description for this time zone.
5. Optionally, in the ID for Standard Time field, enter a label for the time zone for when standard time is in effect.
6. Optionally, in the ID for DST field, enter a label for the time zone for when daylight saving time is in effect (if applicable).
7. Click the **Save** button.

Additional information on using the Translate Application Data page can be found in Application Data.

Trees

The following table lists where to translate tree definitions:

Object	Base Table	Related Language Table	Where to Translate
Tree Definitions	PSTREEDEFN	PSTREEDEFNLANG	Tree Manager, Tree Manager.

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

Definition	Where Displayed	Where Translated With Design Tools
Tree	Search dialogs in Tree Manager dialog boxes.	Tree definition in Tree Manager.
Tree Level	Tree Level label.	Application page accessed via edit level command in PeopleSoft Tree Manager.
Tree Detail Value	Tree Detail Value label.	Application page accessed via edit detail value command in PeopleSoft Tree Manager.
Tree node	Tree node label.	Application page accessed via edit tree node in PeopleSoft Tree Manager.

The translated tree descriptions appear in **PeopleSoft 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 transfers 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 is language-sensitive and can be translated using this page in the same way that you translate other application data.

This section discusses how to:

- Translate trees.
- Translate the tree structure description.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Translating Trees

To translate trees:

1. Ensure you have the specific language installed.
2. Log in to the PeopleSoft system in the target language.
3. Select **Tree Manager**, and then **Tree Manager**.
4. Open the tree whose description you want to translate.

   Click on the root or children.
5. Click the **Edit Data** button (the pencil icon). The Tree Node page appears.
6. In the **Description** field, enter a translation in the target language.
7. Click **OK**.
8. Click the **Save** button.

Translating Tree Structure Descriptions

To translate the tree structure description:

1. Ensure you have the specific language installed.
2. Log in to the PeopleSoft system in the target language.
3. Select **Tree Manager**, and then **Maintain Tree Structure**.
4. Select the **Structure ID** to translate.
5. Open the tree whose description you want to translate.

   Click on the root or children.
6. In the **Description** field, enter a translation in the target language.
7. Click **OK**.
8. Click the **Save** button.

Translating HTML Definitions

HTML objects are considered code, similar to PeopleCode objects, and do not have translated versions. To represent translatable strings within HTML objects, use dynamic text such as message catalog entries. For instance, instead of a hard coded string Sample Text, use `%Message(1001,1234,"Sample Text")`, and translate the string in the message catalog . When the page is displayed in the browser, the HTML object will display the text for the session language.

An example of this in a delivered HTML object is in PT_PROCESSING:

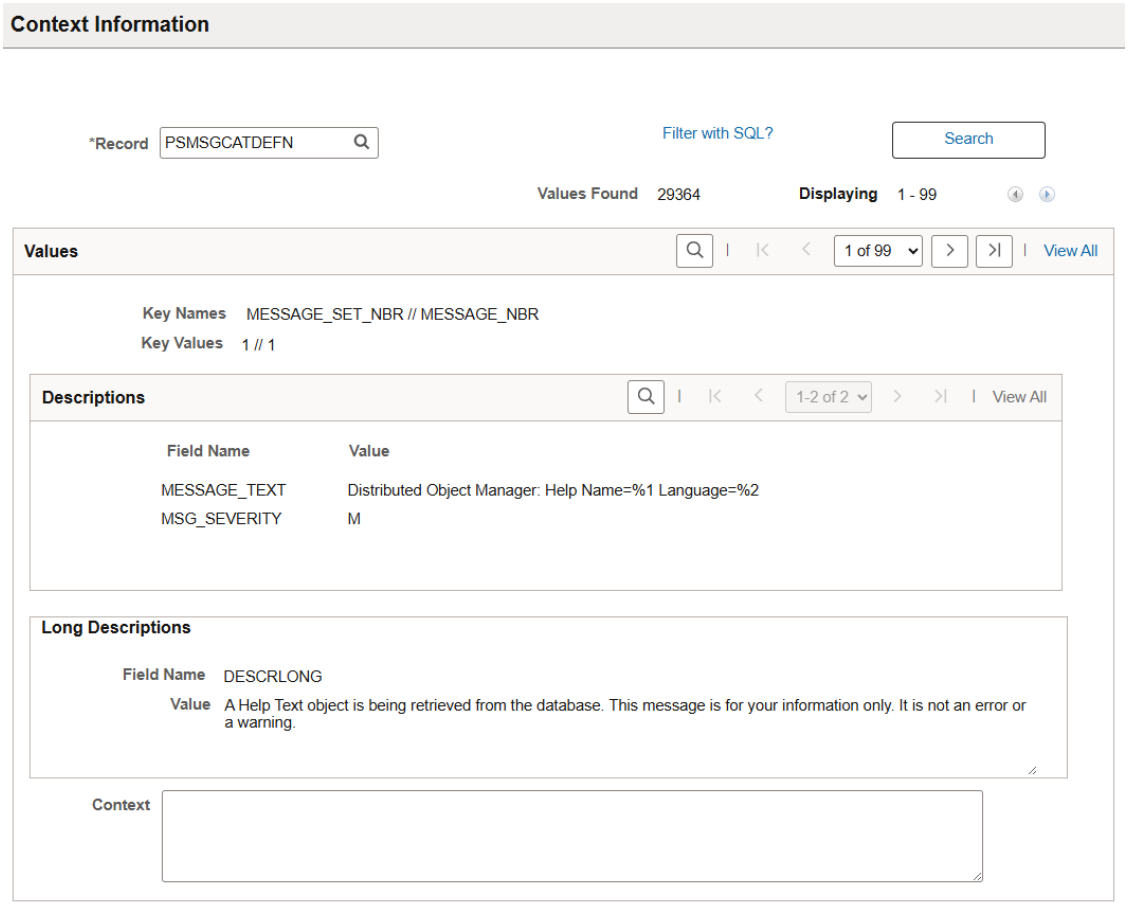
```
delayobj.innerHTML="<span> %Message(124, 537, Processing Please wait) </
span>";
```

Providing Context Information

Access the Context Information page (**PeopleTools**, and then **Translations**, and then **Provide Context**, and then **Translation Context Info**).

Copyright © 1988, 2026, Oracle and/or its affiliates.

This example illustrates the fields and controls on the Context Information page. You can find definitions for the fields and controls later on this page.



When translating field labels in translation factory, a context description sometimes is needed to help translators determine the right way to translate the field label. During the translation process, the Context field displays text entered by the software development team to assist the translation team determine the context of the field label to be translated.

Field or Control	Description
Context	Enter context information in the Context field to assist the translation team when they are translating the text that appears in any of the translatable fields such as the . In this example based on the PSMSGCATDEFN table, the values in the MESSAGE_TEXT field and the Descr Long field are translatable. The Context field could be used to provide context information on either of or both of these fields.

Copyright © 1988, 2026, Oracle and/or its affiliates.

18

Converting PeopleSoft Systems to Unicode Databases

Understanding Converting PeopleSoft Systems to Unicode Databases

PeopleSoft can manage data in many languages in a single database by using Unicode. The Unicode Standard, published by The Unicode Consortium, and synchronized with the International Standard ISO 10646, provides a character set that can support all the characters needed to write virtually every business language in use today.

By creating a PeopleSoft database using Unicode, you can maintain application data, reports, user interface components, and other linguistic elements of your PeopleSoft system in as many languages as you want, all in a single database.

Creating a Unicode database will be of benefit to your organization if you plan on maintaining multiple languages in your PeopleSoft system, or if one or more of the languages you are using are non-Western European languages.

Understanding Planning Unicode Conversions

This section describes considerations to take into account when planning to convert to a Unicode database.

Platform Support

Unicode data storage is currently supported on the following PeopleSoft database platforms:

- Oracle
- Microsoft SQL Server
- IBM DB2/UDB

For specific versions or required patches for these database platforms, see the Certifications tab on My Oracle Support for hardware and software requirements for your PeopleTools release.

Database Sizing

When planning to convert to a Unicode database consider the impact on the physical size of your data.

Depending on the transformation of Unicode that your database platform uses and the mix of languages you plan to use in the PeopleSoft database, you can expect to see the database grow.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Coexistence of Unicode and Non-Unicode Databases

PeopleSoft supports the coexistence of Unicode and non-Unicode databases in a single implementation. Should you need to maintain databases in both Unicode and non-Unicode character sets, most PeopleSoft utilities and commands can operate across both databases seamlessly.

Depending on the character set of the non-Unicode database, some character data may be lost when transferring data between databases. For example, if you load data from a Unicode Japanese and English database into a Latin-1 non-Unicode database, the English data will load correctly, but the Japanese characters will be lost and converted into replacement characters such as question marks.

Although it is easy to have Unicode and non-Unicode databases co-exist, you will need to maintain two separate PeopleSoft file server and batch server environments for each database type, because the COBOL code supplied with some PeopleSoft applications differs significantly between Unicode and non-Unicode databases. Therefore, if you will be maintaining Unicode and non-Unicode databases and you require COBOL, you will need to install the PeopleTools and application CDs twice, and answer the question about Unicode during each installation appropriately

Data Mover

DAT files created by Data Mover are always encoded in Unicode. Data Mover can read these Unicode DAT files and load their contents into either Unicode or non-Unicode databases. When creating a new Unicode database, ensure that the Data Mover command SET UNICODE ON is issued before the IMPORT * command.

Upgrade Compare/Copy

Upgrade Compare and Upgrade Copy supports copying objects between Unicode and non-Unicode databases. There are no restrictions on the source and target database character sets.

Cache Files

Cache files on the PeopleSoft application server are always stored in Unicode format. If you rebuild your non-Unicode database as a Unicode database, you can still use the cache files you may have pre-loaded on the application server.

SQR

On database platforms supporting Unicode, SQR always connects to the database using a Unicode character set via the database vendor's API. Therefore, you can use SQR to copy data to and from Unicode and non-Unicode databases. When writing and reading files with SQR, you can select the character set that should be used to encode the file using the ENCODING parameter to the SQR OPEN command.

SQR may require some settings to be defined in the PSSQR.INI (or PSSQR.UNX) file for SQR to correctly read and write files in Unicode.

Converting to Unicode on Oracle Databases

This section discusses conversion to Unicode on an Oracle database.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Understanding Converting PeopleSoft Systems on Oracle Databases to Unicode

Converting a PeopleSoft system on an Oracle database to Unicode may require creating a new Oracle instance with a Unicode character set. If your current database uses byte semantics (typical for PeopleSoft 8.9 applications or earlier), you must also create new column sizes for PeopleSoft columns created as VARCHAR2.

> **Note**
>
> The AL32UTF8 character set is the preferred character set for the Oracle database. However, the UTF8 character set is also supported.
>
> Be sure to specify CHARACTER SET AL32UTF8 (or CHARACTER SET UTF8) at the end of the CREATE DATABASE command.

For more information on CREATE DATABASE syntax information:

See *Oracle Database SQL Language Reference* SQL Statements: CREATE CLUSTER to CREATE JAVA CREATE DATABASE

For small databases, the simplest way of converting a PeopleSoft database on Oracle to Unicode is to export the entire database using PeopleSoft Data Mover, create a new Oracle instance using a Unicode character set, and re-import the entire database using PeopleSoft Data Mover. The details of this process are *not* provided in this appendix.

For larger databases, a better process is to use PeopleSoft Data Mover to pre-build the structure of the PeopleSoft tables and then use Oracle Data Pump to move the bulk of the data between the old and new Oracle instances. The details of these processes are provided in this appendix.

The Oracle Database Migration Assistant for Unicode (DMU) is a migration tool that can implement character set migration without exporting and importing data. This method does not require creating a new Oracle instance with a Unicode character set, as it modifies the existing instance. You can use DMU instead of Data Mover for converting Oracle databases to Unicode. It is a simpler, faster utility that also provides a validation mode to identify any data that may have issues converting to Unicode.

If you are using the DMU tool for your database conversion, then you can skip to the detailed information on this utility in the section Converting to Unicode on Oracle Databases later in this topic, or else read the following sections on the other available methods to migrate Oracle databases to Unicode.

Exporting PeopleSoft Table Structures Using PeopleSoft Data Mover

This section describes how to use PeopleSoft Data Mover to export the PeopleSoft database structure to a DAT file.

Copyright © 1988, 2026, Oracle and/or its affiliates.

> **Note**
>
> This process exports the database table structure only and does not include the database contents.

Sign into the PeopleSoft database using PeopleSoft Data Mover in user mode(non-bootstrap mode), and run the following script.

```
SET NO TRACE;
SET OUTPUT output_file.dat;
SET NO DATA;
EXPORT *;
```

The DAT file created by PeopleSoft Data Mover will contain only the structure of your PeopleSoft tables and indexes, not the data.

Save the file created by this process. It will be referred to as the *structural export* in this appendix.

Exporting Database Contents

This section provides an overview of exporting database contents and discusses how to:

- Set the NLS_LANG variable.
- Export database contents using Oracle Data Pump.
- Export database contents using the Oracle Export utility.

Setting the NLS_LANG Variable

Before running the export process, verify that the character set specified by NLS_LANG is set to match the character set of the database. NLS_LANG is an environment variable on Unix and a registry setting under HKEY_LOCAL_MACHINE\Software\Oracle on Microsoft Windows.

To verify the current character set of the Oracle database, sign in the Oracle SQL utility of your choice as the database administrator. Then, run the following SQL statement:

```
SELECT VALUE FROM SYS.V_$NLS_PARAMETERS
WHERE PARAMETER ='NLS_CHARACTERSET';
```

For example, if the database character set is WE8ISO8859P1, NLS_LANG should be set to AMERICAN_AMERICA.WE8ISO8859P1, or another language combination with WE8ISO8859P1 as the character set.

Creating and Running the Export Preparation Script for Oracle Data Pump

- Create a directory for the export file such as c:\temp\dmpdir
- In a command prompt, set your Oracle SID

  ```
  set ORACLE_SID = <SID>
  ```

- In SQL*Plus run three commands to specify your directory and grant permissions to the username you will use to run the export.

Copyright © 1988, 2026, Oracle and/or its affiliates.

- Sign into the Oracle SQL utility of your choice as the database administrator:

  ```
  sqlplus / as sysdba
  ```

- Grant EXP_FULL_DATABASE to *username*;
- Check for invalid objects with this command

  ```
  select object_name from dba_objects where status='INVALID';
  ```

  If any rows are returned, you may be able to fix them by running dbmsdb.sql and/or utlrp.sql. There should be no invalid rows before proceeding.
- Quit sql plus.

The data export will specify two schemas: PS,and the ownerid specified in the PSSTATUS.OWNERID table.On the command line,run the expdp command:

```
expdp username/password@SID dumpfile=filename
schemas=OWNERID,ps
```

For instance:

```
 expdp system/manager@HR  dumpfile=exp.dmp schemas=SYSADM,ps
```

At the end of the export, look for the final line such as :"`Job 'OWNERID'.'SYS_EXPORT_SCHEMA_01'` successfully completed.

The export dump file will appear in the dpdump directory under the admin\SID directory. To change the output directory refer to Oracle documentation on the directory parameter of expdp.

In case of errors, you can rerun and add the parameter `trace =1FF0F00`, to generate trace files.

> **Note**
>
> Ensure that the file system on which you will be creating the export file has sufficient space available to accommodate all the data in the PeopleSoft database. A good estimate of the space needed is to calculate the amount of space currently taken by tables owned by the PeopleSoft owner ID and add 20 percent. You can get this data from querying the DBA_FREE_SPACE and DBA_DATA_FILES catalog views. Remember that index data is not stored in export files, only the index definition.
>
> The export (.dmp) file created by this process will be referred to as the *database export* in this document.

Creating a New Oracle Database Instance

This section provides an overview of creating a new Oracle database instance and discusses how to:

- Create the new Oracle database instance.
- Pre-create objects in the new Oracle database instance.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Creating the new database instance involves creating the directories to hold the database, the init <SID>.ora file, and the tnsnames.ora file. On windows,the oradim command is used to start the instance such as.

```
oradim -new sid HR -intpwd manager -startmode auto -pfile
C:\HR\initHR.ora
```

Sign into the Oracle SQL utility of your choice using the database administrator ID and password., run the startup nomount command such as this:

```
startup nomount pfile=c:\HR\initHR.ora
```

The first four are run as sysdba (for instance,sqlplus / as sysdba), and the others are run as the system user (for instance, sqlplus system / manager).

Some of these scripts must be edited to contain the information specific to your database . Check the sizes of the DBF files in the xxDDL.sql script and ensure they will be large enough for the new database.

Understanding the New Oracle Database Instance

First, you must create a new Oracle instance with a Unicode encoding. This database will become the new PeopleSoft Unicode database.

This database must contain the same tablespaces as the original database, if one or more objects owned by the PeopleSoft owner ID were in those tablespaces.

For example, if the original database had 10 tablespaces, and PeopleSoft used five, you must create those five tablespaces with the same names (but not necessarily the same file paths) in the new Unicode instance.

> **Important**
>
> The AL32UTF8 character set is the preferred Unicode character set for the Oracle database.

When you create the new Oracle instance, be sure to specify CHARACTER SET AL32UTF8 (or CHARACTER SET UTF8) at the end of the CREATE DATABASE command.

For More Information on CREATE DATABASE sql command:

See *Oracle Database SQL Language Reference* SQL Statements: CREATE CLUSTER to CREATE JAVA CREATE DATABASE

Pre-Creating Objects in the New Oracle Database Instance

You must pre-create the following objects in the new Oracle database instance:

- The SYSTEM tablespace (automatically created by the CREATE DATABASE command).
- The PSTEMP tablespace.
- Sufficient active rollback segments to complete the import. One segment must be large enough to contain the entire contents of the largest table for the import to succeed without incremental commits.

Copyright © 1988, 2026, Oracle and/or its affiliates.

- The same tablespaces as used by the PeopleSoft owner ID in the existing PeopleSoft database.
- The PS user ID and the PSDBOWNER table. To create these, run the DBOWNER.SQL script provided with the PeopleSoft software. You must manually populate this table with a row for the PeopleSoft database you are moving to the new instance. Check the contents of the existing PS.PSDBOWNER table for a guide.
- The PeopleSoft owner ID and connect ID with the same names as in the existing PeopleSoft database. You can create these IDs by running the PSROLES.SQL, PSADMIN.SQL and CONNECT.SQL scripts. See *PeopleSoft 9.2 Application Installation* for details on running these scripts at installation time.
- Set up appropriate LISTENER.ORA and TNSNAMES.ORA entries for the new instance.

You must set the NLS_LENGTH_SEMANTICS=CHAR variable in your init.ora file after running connect.sql, and restart the database. The installation guide provides information on the use of these scripts and on setting this and other variables.

See *PeopleSoft 9.2 Application Installation for Oracle*.

Adding NLS_LENGTH_SEMANTICS=CHAR

For Apps 8.9 or later, make sure that NLS_LENGTH_SEMANTICS=CHAR is added to the init.ora file after running CONNECT.SQL and before running Data Mover.

Add to init.ora file if not there already: NLS_LENGTH_SEMANTICS=CHAR, and then restart the database. Check with this SQL:

```
select * from nls_database_parameters where parameter in
('NLS_LENGTH_SEMANTICS','NLS_CHARACTERSET');
```

Assuming Apps 8.9 or later, the character set should be UTF8 or AL32UTF8, and the length semantics should be CHAR.

To create a new oracle database instance:

- Create a directory in the filesystem in for the new database
- Create init<SID>.ora file with appropriate paths and control file details

> **Note**
>
> After database creation and before running Data Mover, you will add the entry NLS_LENGTH_SEMANTICS=CHAR to init.ora

- Create directories bdump, cdump, and udump as specified in the init.ora file.
- Set up appropriate LISTENER.ORA and TNSNAMES.ORA entries for the new instance.

Open Command Prompt and set the ORACLE_SID

```
set ORACLE_SID=HR
```

If on Windows, create the new service using oradim, for instance for a database named HR, and a system password of manager:

```
oradim -new -sid HR -intpwd manager -startmode auto -pfile C:\HR\initHR.ora
```


Copyright © 1988, 2026, Oracle and/or its affiliates.

Sign into the Oracle SQL utility of your choice using the database administrator ID and password.

Open each sql file from PS_HOME\scripts\nt folder and edit the file to add necessary parameters and execute the sql in sqlplus.

The scripts are located in the PS_HOME\scripts\nt directory.

Script	Location
CREATEDB10.SQL	PS_HOME\scripts\nt\createdb10.sql;
UTLSPACE.SQL	PS_HOME\scripts\nt\utlspace.sql;
DBOWNER.SQL	PS_HOME\scripts\nt\dbowner.sql;
xxDDL.SQL Tablespace size defaults may need to be increased in the script.	PS_HOME\scripts\nt\ptddl.sql;
xxDDL.SQL (for app example HCDDL.SQL) – Tablespace size defaults may need to be increased in the script.	PS_HOME\scripts\nt\hcddl.sql;
PSROLES.SQL	PS_HOME\scripts\nt\psroles.sql;
PSADMIN.SQL	PS_HOME\scripts\nt\psadmin.sql; Examples at prompts: SYSADM, SYSADM, PSDEFAULT
CONNECT.SQL	PS_HOME\scripts\nt\connect.sql;

After running connect.sql, edit your init.ora file and add the parameter NLS_LENGTH_SEMANTICS=CHAR and restart the database.

For more information on scripts:

See the product documentation for *PeopleSoft 9.2 Application Installation for Oracle*.

Pre-Creating the PeopleSoft Table Structures Using PeopleSoft Data Mover

Sign on to the newly created Oracle database with PeopleSoft Data Mover in bootstrap mode using the database administrator ID and password.

Once signed in, execute the following Data Mover script, substituting *output_file* with the name of the structural export file you created in PeopleSoft Data Mover previously:

```
SET NO TRACE;
SET NO DATA;
SET INPUT output_file;
SET LOG log_file;
SET UNICODE ON;
SET STATISTICS OFF;
SET ENABLED_DATATYPE 9.0;
IMPORT *;
```

This script creates all the tables in the original PeopleSoft database in the new database but with no rows of data. You can speed up the data load time by adding the command `SET NO INDEX;` If you do that, remember to create them in Application Designer later.

Copyright © 1988, 2026, Oracle and/or its affiliates.

> **Note**
>
> The ENABLED_DATATYPE 9.0 command applies only if the PeopleSoft application version is 9.0 or higher. With this flag set, the VARCHAR2 columns will use character semantics instead of byte semantics..The ENABLED_DATATYPE command is related to the NLS_CHARACTER_SEMANTICS setting in the init.ora file. The NLS_CHARACTER_SEMANTICS is not set at the beginning of database creation, but is added to the init.ora file before connecting to Data Mover.
>
> To check the effect of these parameters after the structural import, in sqlplus run the command desc PSSTATUS and verify that the CHAR fields are defined as VARCHAR2(n CHAR) rather than VARCHAR2 (nx3).

Importing Database Contents

This section provides an overview of importing database contents and discusses how to:

- Clear the NLS_LANG variable.
- Import database contents using Oracle Data Pump.
- Create an import parameter file for the Oracle Import utility.
- Run the Oracle Import utility.
- Rebuild PeopleSoft temporary tables.

Understanding Importing Database Contents Using the Oracle Data Pump Utility

Once the PeopleSoft tables have been pre-created by PeopleSoft Data Mover, you are ready to import the database export using Oracle Data Pump (impdp).

> **Important**
>
> Ensure that you follow only the instructions for the pair of utilities that you have chosen to use.

Clearing the NLS_LANG Variable

Before running the Oracle import process, clear the NLS_LANG environment variable (Unix) or remove the NLS_LANG registry setting (Microsoft Windows). Doing so ensures that the Oracle import process reads the character set information from the export file and performs the appropriate conversion to the database's selected Unicode character set.

Importing Database Contents Using Oracle Data Pump

In a command prompt, set your new Oracle SID

Set ORACLE_SID=<SID>

Sign into the Oracle SQL utility of your choice using the database administrator ID and password. Grant permissions to the username you will use to run the import.

```
sqlplus / as sysdba
```


Copyright © 1988, 2026, Oracle and/or its affiliates.

1. Grant IMP_FULL_DATABASE to *username*;
2. Quit SQLPlus.

On the command line, run the impdp command.

`impdp username/password@SID dumpfile=filename schemas=OWNERID,ps content=data_only` For instance:

```
impdp system/manager@HR dumpfile=exp.dmp ⇒
schemas=SYSADM,ps content=data_only
```

Look for: Job *"SYSTEM"."SYS_IMPORT_SCHEMA_01"* successfully completed.

In case of errors, you can rerun and add the parameter `trace=1FF0F00`, to generate trace files. One common error is running out of space in a tablespace, due to setting it too small in the xxDDL.SQL scripts. To fix this, use a command like the following:

```
alter database datafile 'filename.DBF'resize new_size;
```

Ensure that the import completes successfully with no errors before continuing.

Rebuilding PeopleSoft Temporary Tables

After you import the database, you must rebuild the temporary tables. To rebuild the temporary tables, sign into PeopleSoft Data Mover in user mode (non-bootstrap mode), and run the following command:

```
CREATE_TEMP_TABLE *
```

Building Indexes

If you included the SET INDEXES OFF line of the Data Mover import, you need to build the indexes. Log into Application Designer and create a project called ALLRECORDS, and insert all record definitions into the project. Build the project to build the indexes defined for the records.

Specifying the Unicode Database and Data Types in Your PeopleSoft System

When the import has completed successfully, you must specify in your PeopleSoft system that the database is now a Unicode database. To do this, sign into the Oracle SQL utility of your choice using the database owner ID. Run the following SQL statement:

```
UPDATE PSSTATUS SET UNICODE_ENABLED=1;
```

> **Note**
>
> You must set the PSSTATUS field of UNICODE_ENABLED to 1 for all Unicode databases and 0 for all Non-Unicode databases.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Since you set ENABLED_DATATYPE previously, you also need to specify that the database uses Oracle CLOB and BLOB fields, as follows:

```
UPDATE PSSTATUS SET DATABASE_OPTIONS=34;
```

Running GRANT.SQL

Before signing into the new PeopleSoft Unicode database, edit the GRANT.SQLscript (provided in your PeopleTools installation in the scripts directory) to include the connect ID you created previously. Then, run GRANT.SQL to set up the appropriate grants to the connect ID.

Sign into the Oracle SQL utility of your choice using the database administrator ID and password. Run the GRANT.SQL script.

You should now be able to configure your connection with Configuration Manager, and log in with Application Designer. It is recommended to check the database integrity by running the SQR audits DDDAUDIT and SYSAUDIT.

Rerunning an Oracle Database Import

When the PSSTATUS table is imported from a non-Unicode export file, the UNICODE_ENABLED flag is 0.

Therefore, before re-running any PeopleSoft Data Mover imports, or any time PSSTATUS is re-imported from a non-Unicode export file, set the UNICODE_ENABLED flag to *1*.

Converting to Unicode on Oracle Databases Using Database Migration Assistant for Unicode

If your database supports it, you can convert your databases to Unicode using Database Migration Assistant for Unicode (DMU). It is a faster method than exporting and importing the non-Unicode files. You will need a minimum Apps version of 9.0, Tools version of 8.48, and DMU version of 2.1.2 or later.

> **Note**
>
> A user with SYSDBA privilege, such as SYS can connect the DMU client to the database.
>
> The database will need the PL/SQL package prvtdumi.plb, for instance:
>
> ```
> sqlplus / as sysdba
> ```
>
> ```
> SQL>@?/rdbms/admin/prvtdumi.plb
> ```

> **Note**
>
> Java SE Development Kit (JDK) is required to run the DMU client. The main DMU documentation has details of the JDK.

To convert the database in Unicode using DMU:

Copyright © 1988, 2026, Oracle and/or its affiliates.

1. Verify the configuration requirements for DMU and install the required patches.
   a. Verify if the database and the operating system supports the current DMU release.
   b. Check for any restrictions for the databases to be migrated in the DMU Release Notes.
   c. Install required database patch. You can download the patch from My Oracle Support.
   d. Install the PL/SQL package in the database.
   e. Install Oracle XML DB component (XDB) in your database.

> **Note**
>
> You can check if a valid XDB is installed in your system. To check, enter following in the command prompt:
>
> `select comp_id, status from dba_registry;`
>
> If you do not have XDB then to install XDB, enter the following in the command prompt:
>
> `run catqm.sql, for instance as catqm.sql xdb SYSAUX TEMP NO.`

2. Setup the client for DMU and connect to the database:
   a. Install the JDK version required for the DMU version.

> **Note**
>
> Make a note of the directory where you installed JDK. You will need to provide the path to it in the DMU.

   b. Install the DMU software from Oracle Technology Network download page.
   c. Start the DMU. When you start the DMU for the first time, it asks for the JDK installation directory.
   d. Create a database connection. Enter user Id, password and the network connection data of the database administrator; the target database host name, the TNS listener port, and the database service name. The user must have the SYSDBA privilege in the database.

> **Note**
>
> Before connecting to the DMU client, grant SYSDBA privilege to your user using a password file from orapwd utility.
>
> For Windows, this file must be named `ORACLE_HOME\database\PWD<ORACLE_SID>.ora`, and for Unix this file must be named `ORACLE_HOME/dbs/orapw<ORACLE_SID>`.

3. Run the Unicode migration in DMU:
   a. Install the DMU repository.

Copyright © 1988, 2026, Oracle and/or its affiliates.

> **Note**
>
> DMU will prompt you to update the `NLS_LENGTH_SEMANTICS` in init.ora at the end of the conversion.

b. Start the migration process of the database to Unicode. The utility guides you through the three steps to Scan, Cleanse, and Convert.

> **Note**
>
> If issues are reported during the scan, fix the issues outside of DMU and then re-scan until the scan reports no issues. PeopleSoft does not support cleansing data using the DMU Cleansing Editor. Use PeopleSoft Pure Internet Architecture or Application Designer to fix reported issues. For more information on working with translated data, see Understanding Application Definition Translation.

c. Validate that the character set in the database is in Unicode.

4. Set `NLS_LENGTH_SEMANTICS=CHAR` in your init.ora and restart your database. PeopleSoft Unicode databases use character length semantics.

5. Run Audit:

   Run the audits such as DDDAUDIT, SYSAUDIT, and an Alter Audit.

> **Note**
>
> With DMU versions before 2.1.2, any tables reported in the alter audit should be rebuilt in Application Designer to ensure that fields defined as LONG are implemented as CLOB when needed. With DMU 2.1.2 and later, the alter audit should show that no changes are needed.

If you use the DMU method, you don't need to rebuild tables and indexes, or do imports, unless tables are reported in the alter audit. After the migration, the character set will be AL32UTF8, and the VARCHAR2 columns lengths will specify the number of characters instead of the number of bytes. For instance "descr psmsgcatlang" will show that message_text is varchar2(100 char).

This method is also documented for PeopleTools 8.48 and above. You can access the document from My Oracle Support, Document ID: 1627714.1.

For more information on the Oracle Database Migration Assistant for Unicode on supported configuration, DMU documentation, and accessing DMU client for download, go to the Oracle Technology Network.

Verifying a Database After Migrating to Unicode:

Verify the following after DMU conversion is complete:

Copyright © 1988, 2026, Oracle and/or its affiliates.

1. The NLS parameters should be AL32UTF8 and CHAR:

   ```
   select * from V$NLS_PARAMETERS where parameter in
   ('NLS_CHARACTERSET','NLS_LENGTH_SEMANTICS');
   ```

2. The PSSTATUS field of UNICODE_ENABLED should be 1:

   ```
   select unicode_enabled from psstatus;
   ```

3. There should be only two columns with CHAR_USED as B: PTIA_VWNAME and PTIA_VWDEPNAME:

   ```
   SELECT TABLE_NAME, COLUMN_NAME, DATA_TYPE, CHAR_USED FROM USER_TAB_COLUMNS
   WHERE DATA_TYPE='VARCHAR2' AND CHAR_USED='B';
   ```

4. Run Audit:

   Run the audits such as DDDAUDIT, SYSAUDIT, and an Alter Audit. There should be no new SYSAUDIT or DDDAUDIT exceptions that were not there before the migration.

> **Note**
>
> These checks also apply to any database that is Unicode and is not specific to DMU migrations.

Rerunning Microsoft SQL Server and DB2 Database Imports

When the PSSTATUS table is imported from a non-Unicode export file, the UNICODE_ENABLED flag is set to *0*. Before re-running any PeopleSoft Data Mover imports, or any time PSSTATUS is re-imported for a non-Unicode export file, set the UNICODE_ENABLED flag to *1*.

Final Database Cleanup

After importing your data, run the Final Database Cleanup section from dbsetup.dms, generated by the Database Creation Wizard. Comment out the IMPORT line of the DMS script, since you have already imported all table structures and data. Run the security steps and later lines from dbsetup.dms. Besides the security steps, the dbsetup.dms script may contain these lines:

```
ENCRYPT_PASSWORD *;


CREATE_TRIGGER *;


REPLACE_VIEW *;


CREATE_TEMP_TABLE *;
```


Copyright © 1988, 2026, Oracle and/or its affiliates.

The tables referenced by these lines are the ones in the structural export files, or full datamover export file, depending on which one you originally exported.

You should now be able to configure your connection with Configuration Manager, and log in with Application Designer. It is recommended to check the database integrity by running the SQR audits DDDAUDIT and SYSAUDIT.

Converting to Unicode on Microsoft SQL Server and DB2 Databases

This section discusses converting to Unicode on Microsoft SQL server and DB2 databases.

Understanding Converting to Unicode on Microsoft SQL Server and DB2 Databases

On Microsoft SQL Server and DB2 databases, PeopleSoft supports converting a PeopleSoft database to Unicode by re-creating the database using Data Mover to import database contents. With Microsoft SQL Server databases, for the best performance, run Data Mover on the same machine as the database server.

PeopleSoft requires that an entire database be created as a Unicode database or as a non-Unicode database – mixing CHAR and NVARCHAR columns in a single PeopleSoft database is not supported.

Exporting PeopleSoft Databases Using PeopleSoft Data Mover

Sign into the PeopleSoft database using PeopleSoft Data Mover in non-bootstrap mode, and run the following script.

```
SET NO TRACE;
SET OUTPUT outputfile;
EXPORT *;
```

The DAT file created by PeopleSoft Data Mover will contain all the data in the PeopleSoft database, so be sure that the volume to which it will be written has sufficient disk space to handle and store the exported data.

This process will create a file containing the structure and contents of your PeopleSoft database.

Creating New SQL Server or DB2 Unicode Databases

You will need to create a new SQL Server or DB2 database corresponding to your target PeopleSoft Unicode database. On SQL Server this database can co-exist on the same server as your original non-Unicode database, as SQL Server allows Unicode and non-Unicode databases to reside on a single server, regardless of the character set selected during SQL Server installation.

To create this database, follow the steps in the appendix “Creating a Database Manually” in the PeopleTools installation guide, from the beginning of the appendix. Stop at the step titled “Creating Data Mover Import Scripts” and continue with the steps below. You can also create a new database using the Database Configuration Wizard, but do not allow the wizard to start importing data.

Copyright © 1988, 2026, Oracle and/or its affiliates.

See the product documentation for *PeopleSoft 9.2 Application Installation for Microsoft SQL Server*, “Creating a Database Manually.”

See the product documentation for *PeopleSoft 9.2 Application Installation for DB2 for z/OS*, “Creating a Database Manually.”

Importing Database Contents Using PeopleSoft Data Mover

Run PeopleSoft Data Mover in bootstrap mode against the new database. Login in bootstrap mode using the OwnerID and password. Once signed in, run the following script:

```
SET NO TRACE;
SET INPUT inputfile;
SET UNICODE ON;
SET ENABLED_DATATYPE 9.0;
IMPORT *;
```

> **Note**
>
> `SET ENABLED_DATATYPE 9.1` is also valid and has the same effect as `SET ENABLED_DATATYPE 9.0`.

This will import the entire contents of your original database into the new database, while creating character columns using the NVARCHAR data type (for SQL Server) or VARBINARY data type (for DB2), and converting your data to Unicode.

Specifying Unicode Databases in PeopleSoft Systems

When the import has completed successfully, you must specify in your PeopleSoft system that the database is now a Unicode database. To do this, sign into the Oracle SQL utility of your choice using the database owner ID. Run the following SQL statement:

```
UPDATE PSSTATUS SET UNICODE_ENABLED=1
```

> **Note**
>
> You must set the PSSTATUS field of UNICODE_ENABLED to 1 for all Unicode databases and 0 for all Non-Unicode databases.

Since you set ENABLED_DATATYPE previously, you also need to specify that the database uses Oracle CLOB and BLOB fields, as follows:

```
UPDATE PSSTATUS SET DATABASE_OPTIONS=34;
```

Rerunning GRANT.SQL

Before signing onto Application Designer to create your views, you must first re-run GRANT.SQL (provided in your PeopleTools installation in the SCRIPTS directory) to grant

Copyright © 1988, 2026, Oracle and/or its affiliates.

access to the newly created tables to your connect ID. Once this is completed, you should be able to sign into Application Designer.

Copyright © 1988, 2026, Oracle and/or its affiliates.

19

Updating Time Zone Definitions

Understanding the U.S. Energy Policy Act of 2005

From time to time, legislative changes can impact the definitions of time zones, such as whether a time zone observes daylight saving time and the start or end date for daylight saving time observance. The PeopleSoft system provides updated time zone definitions in the next release following such a legislative change, and new customers as of that release receive the updated time zone data. However, existing customers upgrading to that new release do not receive the benefit of updated time zone data because customer time zone data is not changed during upgrade. In these cases, you can manually modify your time zone definitions to comply with changing rules.

As a specific example, the U.S. Energy Policy Act of 2005 changed time zone definitions in the U.S. and Canada. PeopleSoft-delivered data in the current PeopleTools release reflects these changes. Depending on your status as a new or existing customer, you might be required to manually update your time zone data:

- If you are a new PeopleTools customer, you have received time zone data that complies with the Energy Policy Act of 2005. You do not have to manually update your time zone definitions.
- If you are an existing PeopleTools customer upgrading to the current release *and* you already manually updated your time zone definitions to comply with the Energy Policy Act of 2005, your updated time zone data will not be changed during upgrade. You do not have to manually update your time zone definitions again.
- If you are an existing PeopleTools customer upgrading to the current release and you have not updated your time zone definitions to comply with the Energy Policy Act of 2005, your noncompliant time zone data will not be changed during upgrade. You will have to manually update your time zone definitions to comply with the Energy Policy Act of 2005. See the remaining subsections in this section.

As a result of the U.S. Energy Policy Act of 2005, daylight saving time starts earlier and ends later beginning in 2007. In the U.S. and Canada, daylight saving time now starts the second Sunday in March and ends the first Sunday in November.

To comply with this change, you need to add two new daylight saving time IDs (DST IDs), set your affected time zones to use the new DST IDs, and regenerate your offset data. Depending on your business needs, you may also want to define new time zones and associate them with existing DST IDs.

Separately, you should apply operating system patches according to your vendor’s directions so that file system time stamps will also be in compliance.

Adding New DST IDs

To add two new DST IDs:

1. Select PeopleTools, Utilities, International, Time Zones, DST IDs.
2. Click the Add button and add the following information:

Copyright © 1988, 2026, Oracle and/or its affiliates.

Field	Value
DST ID	22ndSunMar
Absolute	cleared
Month	Mar
Day	2
Day of Week	Sunday
Hour	2
Minute	0
Description	Second Sunday in March, 2:00am

3. Click the Add button again and add the following information:

Field	Value
DST ID	2FirstSunNov
Absolute	cleared
Month	Nov
Day	1
Day of Week	Sunday
Hour	2
Minute	0
Description	First Sunday in Nov, 2:00am

4. Click Save to save your changes.

Update Affected Time Zones

To update affected time zones:

1. Select PeopleTools, Utilities, International, Time Zones, Time Zone IDs. Select the Daylight Saving Data tab.
2. Choose the DST start and end IDs for the affected time zones.

   In the U.S. and Canada, the affected time zones that PeopleTools delivers are the following:

   - AKST Alaska Time (U.S.)
   - AST Atlantic Time (Canada)
   - CST Central Time (U.S.)
   - EST Eastern Time (U.S.)
   - MST Mountain Time (U.S.)
   - NST Newfoundland Time (Canada)
   - PST Pacific Time (U.S.)

Generate New Query Offsets

To generate new query offsets:

Copyright © 1988, 2026, Oracle and/or its affiliates.

1. On the Time Zone IDs page, click the Generate Query Offsets button.
2. Enter start and end dates for the time range to generate, for instance, 1/1/2009 to 1/1/2011, and click OK.

   This updates the time zone offset table (PSTZOFFSET) to reflect the current time zone and DST definitions. This data can be re-generated as needed.

   See the additional time zone documentation available on My Oracle Support.

Define New Time Zones

You may want to define new time zones. For instance, Canada has decided to follow the U.S. standard, while Mexico has decided to stay with the older definition. This means that for several weeks each year, PST in Los Angeles will be different from PST in Tijuana.

You can define a new time zone, such as PSTM (for PST Mexico) with the older DST ID, to distinguish it from the PST (for PST U.S. and Canada) with the new DST ID.

See Understanding Time Zones.

After defining your new time zones, run Generate Query Offsets again so that the offsets include the new time zones.

Copyright © 1988, 2026, Oracle and/or its affiliates.

20

Troubleshooting

Understanding Troubleshooting

We recommend these solutions for some of the more common issues encountered by Peoplesoft customers with global implementations.

> **Note**
>
> Do not construe the information provided here as consulting or implementation advice for your specific industry or your individual organization. You should adapt or disregard this information based on the needs of your organization. Oracle does not guarantee that the information included here will work as intended within your customized environment.

PeopleSoft Hot Keys Do Not Function As Expected on a non-U.S. Keyboard

Certain PeopleSoft hot keys do not work as expected on alternate, non-U.S. keyboard layouts.

Description

Certain PeopleSoft hot keys do not work as expected on alternate, non-U.S. keyboard layouts. For example, **Alt+'**, **Alt+**, and **Alt+/** do not produce the expected results on the AZERTY keyboard. This occurs because some keys on non-U.S. keyboards produce different key codes than the same key on a U.S. keyboard (also known as a QWERTY keyboard).

The function of PeopleSoft hot keys are documented in the Using PeopleSoft Applications PeopleBooks.

See Accessibility Guide: Using Keyboard Shortcuts.

Solution

To access PeopleSoft hot key functions that are not working on a non-U.S. keyboard, you need to identify the key on your keyboard that produces the desired key code. A simple HTML program can be used in your browser to help you identify the key codes generated by U.S. and non-U.S. keyboards.

Use the following procedure to determine which keystroke combinations on a non-U.S. keyboard will produce the desired key code, and therefore, the desired PeopleSoft hot key function:

Copyright © 1988, 2026, Oracle and/or its affiliates.

1. Save the following program code to an HTML file (for example, getkeycode.htm):

```
<html>
<body>
<script language="JavaScript">
function keydown_keycode() {
x = event.keyCode; alert("Key: " + String.fromCharCode(x) + " and Keycode:
" + x);
}
</script>
<p>Press Any Key</p>
<body onKeyDown="keydown_keycode();">
</body>
</html>
```

2. In your browser, open the getkeycode.htm file.
3. Using the following table, identify the PeopleSoft hot key that does not produce the expected results. In the same table, note the key code for this when it is used alone (that is, not in combination with the **Alt** key).

> **Note**
>
> You can also verify the key code by pressing the key on a U.S. keyboard. Use the HTML program in your browser to display key codes for various keys.

PeopleSoft Hot Key	Key Code (Alone)	PeopleSoft Function
Alt+1	49	Saves a page in a transaction. Moves to the Search or Add button on a search or look up page. Moves to the OK button on a secondary page.
Alt+2	50	Returns to the search page from the transaction page.
Alt+3	51	View the next row in the list when the button is active.
Alt+4	52	View the previous row the in list when the button is active.
Alt+5	53	Accesses the Look Up page. Opens the calendar prompt.
Alt+6	54	Opens the pop-up window on a page.
Alt+7	55	Inserts a row in a grid or scroll area.
Alt+8	56	Deletes a row in a grid or scroll area.
Alt+9	57	Accesses the Help line.

Copyright © 1988, 2026, Oracle and/or its affiliates.

PeopleSoft Hot Key	Key Code (Alone)	PeopleSoft Function
Alt+0	48	When in Expert Entry mode, activates the Refresh button, which validates the data entered on the page.
Alt+.	190	Displays the next set of rows in a grid or scroll area.
Alt+,	188	Displays previous set of rows in a grid or scroll area.
Alt+/	191	Finds data in a grid or scroll area.
Alt+'	222	Displays all rows of data in a grid, scroll area, or search page results list.
Alt+	220	Toggles between Add a New Value and Find an Existing Value on a search page. Toggles between Update/ Display, Include History, and Correct History action modes on the toolbar on a transaction page.
Ctrl+J	74	Displays the system information page.
Ctrl+K	75	When on a search or transaction page, accesses a page with a list of keyboard navigation shortcuts using hot keys and access keys.
Ctrl+Y	89	Toggles the menu pagelet between collapse and expand.
Ctrl+Z	90	Accesses the menu search box.
Ctrl+Tab	9	Toggles the focus through the frame set.
Enter	13	Activates the OK button, where appropriate. On a search page, activates the Search button. On a look up page, activates the lookup button.
Esc	27	Activates the Cancel button, where appropriate.

4. On your non-US keyboard, identify the key that produces this key code. Use the HTML program in your browser to display key codes for various keys.

For example, on a U.S. keyboard, the **** key has a key code of 220 as shown in the table. On an AZERTY keyboard, ***** has the key code of 220.

The following table shows the AZERTY keyboard equivalents for three PeopleSoft hot keys that do not work as expected:

Copyright © 1988, 2026, Oracle and/or its affiliates.

PeopleSoft Hot Key	AZERTY Equivalent	Key Code (Alone)	PeopleSoft Function
Alt+/	**Alt+:**	191	Finds data in a grid or scroll area.
Alt+'1	**Alt+²**	222	Displays all rows of data in a grid, scroll area, or search page results list.
Alt+	**Alt+***	220	Toggles between Add a New Value and Find an Existing Value on a search page. Toggles between Update/Display, Include History, and Correct History action modes on the toolbar on a transaction page.

Daylight Saving Time Issues on Oracle WebLogic Server

The Java platform no longer supports these three-letter time zones — EST, EDT, HST, HDT, MST, and MDT — and has replaced them with Olson time zones such as Americas\New_York. PeopleTools functionality is not affected by this change; however, your Oracle WebLogic Server might be affected. As a result, daylight saving time is calculated incorrectly by Oracle WebLogic Server for these time zones under certain circumstances.

For more information, see Timezones, Daylight Savings, and the TZupdater for the Java Runtime Environment (JRE)

Special International Characters Appear Corrupted in the View Log/Trace Files

Special international characters in the View Log/Trace files appear corrupted when viewed directly in the browser from the **View Log/Trace** link in Process Monitor.

Description

Browsers automatically assume that files are in native encoding such as Latin-1. For example, if a UTF-8 file does not have a byte order mark specified, the browser automatically and incorrectly assumes that the file is encoded in Latin-1. Therefore, when you view the trace files directly from the **View Log/Trace** link in Process Monitor, special international characters are displayed as corrupted characters.

However, these characters display correctly when the files are viewed in a text editor, such as Notepad or Notepad++.

Solution

To view special international characters correctly in the browser from the **View Log/Trace** link in Process Monitor, you can implement either of the following solutions:

- Change the web.xml file.
- Change the Process Scheduler character set.

Copyright © 1988, 2026, Oracle and/or its affiliates.

Although you can use either of these two methods, changing the web.xml file is the preferred option because it keeps all the characters in UTF-8 as delivered. Also, if you are updating a web.xml file, then changing the web.xml file is preferred.

Change the web.xml File

To change the web.xml file:

1. Navigate to webserv/<DOMAIN>/applications/peoplesoft/PORTAL.war/WEB-INF/web.xml.
2. Take a backup of the file.
3. Open the file web.xml in a text editor.
4. Change the line:

   ```
   <mime-mapping> <extension> log </extension> <mime-type> text/plain </mime-
   type> </mime-mapping>
   ```

   to:

   ```
   <mime-mapping> <extension> log </extension> <mime-type> text/
   plain;charset=utf-8 </mime-type> </mime-mapping>
   ```

5. Add the line:

   ```
   <mime-mapping> <extension> stdout </extension> <mime-type> text/
   plain;charset=utf-8 </mime-type> </mime-mapping>
   ```

6. Save the file.
7. Bounce the Web Server and Process Scheduler, and then clear their **Cache** folders.
8. Test by submitting a new process.

> **Note**
>
> You can also use this method to configure files with other extensions besides .stdout and .log.

Change the Process Scheduler Character Set

To change the Process Scheduler character set:

1. Open the Process Scheduler configuration file (psprcs.cfg) and Application Server configuration file (psappsrv.cfg).
2. Navigate to the **[PSTOOLS]** parameters.
3. Set the character set to the native encoding character set on your system. For example, for native Japanese encoding, you can set the character set to SJIS.
4. Save the files.
5. Bounce the Web Server and Process Scheduler, and then clear their **Cache** folders.
6. Test by submitting a new process.

Copyright © 1988, 2026, Oracle and/or its affiliates.