
Enterprise PeopleTools 8.49 PeopleBook: Global Technology

March 2007

Enterprise PeopleTools 8.49 PeopleBook: Global Technology
SKU PT849GBL-B 0307
Copyright © 1988-2007, Oracle. All rights reserved.

The Programs (which include both the software and documentation) contain proprietary information; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. This document is not warranted to be error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose.

If the Programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are “commercial computer software” or “commercial technical data” pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software--Restricted Rights (June 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee’s responsibility to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and we disclaim liability for any damages caused by such use of the Programs.

The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

Oracle, JD Edwards, PeopleSoft, and Siebel are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Open Source Disclosure

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

Apache Software Foundation

This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>). Copyright © 2000-2003. The Apache Software Foundation. All rights reserved. Licensed under the Apache License, Version 2.0 (the “License”); you may not use this file except in compliance with the License. You may obtain a copy of the License at <http://www.apache.org/licenses/LICENSE-2.0>.

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

OpenSSL

Copyright © 1998-2005 The OpenSSL Project. All rights reserved.

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

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

Loki Library

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

Helma Project

Copyright © 1999-2004 Helma Project. All rights reserved. THIS SOFTWARE IS PROVIDED “AS IS” AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE HELMA PROJECT OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

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

Sarissa

Copyright © 2004 Manos Batsis.

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

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

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

ICU

ICU License - ICU 1.8.1 and later COPYRIGHT AND PERMISSION NOTICE Copyright © 1995-2003 International Business Machines Corporation and others. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, provided that the above copyright notice(s) and this permission notice appear in all copies of the Software and that both the above copyright notice(s) and this permission notice appear in supporting documentation. THE SOFTWARE IS PROVIDED "AS IS," WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

All trademarks and registered trademarks mentioned herein are the property of their respective owners.

Sun's JAXB Implementation – JSDK 1.5 relaxngDatatype.jar 1.0 License

Copyright © 2001, Thai Open Source Software Center Ltd, Sun Microsystems. All rights reserved.

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

W3C IPR SOFTWARE NOTICE

Copyright © 2000 World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University). All Rights Reserved.

Note: The original version of the W3C Software Copyright Notice and License could be found at <http://www.w3.org/Consortium/Legal/copyright-software-19980720>.

THIS SOFTWARE AND DOCUMENTATION IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE OR DOCUMENTATION WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS. COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE SOFTWARE OR DOCUMENTATION.

Contents

General Preface

About This PeopleBook xv

PeopleSoft Enterprise Application Prerequisites.....xv

Application Fundamentals.....xv

Documentation Updates and Printed Documentation.....xvi

 Obtaining Documentation Updates.....xvi

 Downloading and Ordering Printed Documentation.....xvi

Additional Resources.....xvii

Typographical Conventions and Visual Cues.....xviii

 Typographical Conventions.....xviii

 Visual Cues.....xix

 Country, Region, and Industry Identifiers.....xix

 Currency Codes.....xx

Comments and Suggestions.....xx

Common Elements Used in PeopleBooks.....xx

Preface

Global Technology Preface.....xxiii

PeopleTools Global Technology.....xxiii

Part 1 Getting Started

Chapter 1

Getting Started with Global Technology.....3

PeopleTools Global Technology Overview.....3

PeopleTools Global Technology Implementation.....4

Part 2 Configuring Your System for Global Implementation

Chapter 2

Controlling International Preferences.....	11
Understanding International Preferences.....	11
Setting Up Language Preferences in the PeopleSoft Pure Internet Architecture.....	12
Understanding the Session Language.....	12
Choosing a Session Language.....	12
Changing the Session Language While Signed In.....	13
Changing the Session Language Programmatically.....	13
Determining the Session Language When the Signon Page is Bypassed.....	14
Controlling the User Profile Language Preference.....	14
Determining the Language of Help Documentation.....	15
Applying Browser Language Preferences.....	15
Understanding Browser Language Preferences.....	15
Setting Browser Language Preferences.....	15
Using Browser Language Settings to Determine the Sign-in Language.....	16
Setting Up Language Preferences for Microsoft Windows-Based PeopleTools.....	17
Understanding Language Preferences for Microsoft Windows-Based PeopleTools.....	18
Controlling PeopleSoft Configuration Manager Language Settings.....	18
Setting Up Locale-Based Formatting for the PeopleSoft Pure Internet Architecture.....	19
Understanding Locale-Based Formatting Drivers.....	20
Pages Used to Set Up PeopleSoft Pure Internet Architecture Locale-Based Formatting.....	21
Defining User-Specific Formatting.....	21
Defining Locales for Locale-Specific Formatting.....	22
Defining Locale-Specific Formatting.....	23
Applying System-Wide Default Formatting.....	24
Setting Up Locale-Based Formatting for Microsoft Windows-based PeopleTools.....	24

Chapter 3

Setting and Maintaining Time Zones.....	27
Understanding Time Zones.....	27
Setting the Base Time Zone.....	28
Maintaining Time Zones.....	30
Understanding Time Zone Maintenance.....	30
Viewing Time Zone Definitions.....	30
Generating Time Zone Offsets.....	31
Viewing Daylight Savings Start and End Dates.....	32
Complying with the Energy Policy Act of 2005.....	33
Controlling Time Zone Display When Developing PeopleSoft Application Objects.....	35
Understanding Time Zone Display Settings.....	35

Understanding Time Zone Precedence.....	36
Setting Time Zone Options and Related Date Fields in Record Field Properties.....	37
Setting Time Zone Options in Page Field Properties.....	38
Using PeopleCode Time Functions.....	38

Chapter 4

Selecting and Configuring Character Sets and Language Input and Output.....	39
Understanding Character Sets.....	39
Character Sets.....	39
Common Character Sets.....	40
The Unicode Standard.....	42
Non-Unicode Character Sets.....	46
Character Sets in the PeopleSoft Pure Internet Architecture.....	47
Selecting Character Sets.....	54
Selecting Database Character Sets.....	54
Selecting Application Server Character Sets.....	55
Selecting and Managing Client Workstation Character Sets.....	55
Selecting Email Character Sets.....	57
Validating Field Length.....	58
Understanding Application Designer Field Length Semantics.....	58
Understanding Field Length Checking for Non-Unicode Databases.....	60
Enabling or Disabling Data Field Length Checking.....	61
Handling International Characters in PeopleTools.....	61
Standard Name Formats for Chinese, Japanese, and Korean Characters.....	62
Detecting and Converting Characters.....	62

Chapter 5

Controlling Currency Display Format.....	63
Understanding Currency-Specific Settings.....	63
Setting Up Currency Amount Fields When Developing Applications.....	64
Setting Currency Field Display Properties When Developing Applications.....	64
Using System-Wide Multicurrency Settings.....	65
Understanding Multicurrency Settings.....	65
Connecting a Currency Control Field to the Multi-Currency Check Box.....	66
Enabling or Disabling System-Wide Currency Settings.....	66
Resizing Currency Fields by Using the International Field Size Utility.....	67

Chapter 6

Running COBOL in a Unicode Environment.....	69
Understanding COBOL in a Unicode Environment.....	69
Unicode Encodings in PeopleSoft COBOL.....	69
Expanded Storage Space Requirements.....	70
Special Logic for Parsing Unicode Strings.....	71
COBOL Sorting.....	71
Unicode-Specific Error Messages.....	71
Running the COBOL Conversion Utility.....	72
Understanding the COBOL Conversion Utility.....	72
Running the Conversion Utility.....	73
Identifying Converted COBOL Programs.....	74
Understanding What Is Expanded.....	74
Using Utility Directives.....	79
Viewing Error Logs.....	81
Fine-Tuning COBOL Programs.....	82
Identifying Unicode and Non-Unicode Data.....	83
Specifying Column Lengths in Dynamic SQL.....	83
Defining Single Character Arrays.....	85

Chapter 7

Running COBOL in a z/OS Unicode Environment.....	95
Understanding Running COBOL in a z/OS Unicode Environment.....	95
Unicode Encodings in PeopleSoft COBOL on z/OS.....	95
Data Usage for Unicode Storage.....	96
Input/Output in the z/OS Unicode Environment.....	97
Understanding the COBOL Unicode Conversion Utility for z/OS.....	98
COBOL Unicode Conversion Utility for z/OS.....	98
Applying Patches.....	98
Running the COBOL Unicode Conversion Utility for z/OS.....	98
Automatically Running the COBOL Unicode Conversion Utility for z/OS.....	99
Manually Running the COBOL Unicode Conversion Utility for z/OS.....	99
Identifying Converted COBOL Programs.....	100
Understanding Converted Data.....	100
Copybook Conversion.....	100
Data Division Conversion Rules.....	101
Exceptions to Data Division Conversion Rules.....	101
Procedure Division Conversion Rules.....	103
Columns.....	105

Using Error, Exception and Summary Logging.....	106
Understanding Error, Exception and Summary Logs.....	106
Setting Logging Parameters.....	106
Viewing Exception Log Messages.....	106
Viewing Summary Log Messages.....	107
Fine-Tuning COBOL Programs for the z/OS Unicode Environment.....	108
Understanding Fine-Tuning Converted COBOL Programs for the z/OS Unicode Environment.....	108
Identifying Unicode Data for z/OS, Unicode Data for Microsoft Windows NT/Unix and Non-Unicode Data.....	108
Understanding Character Fields and Byte Size.....	109
Understanding Character Set Conversion.....	110

Chapter 8

Sorting in PeopleTools.....	111
Understanding Sort Orders.....	111
Sorting Overview.....	111
PeopleTools Sorts.....	112
Database-Level SQL ORDER BY Sorts.....	113
PeopleTools In-Memory Sorts.....	114
Binary Sorts.....	114
Setting the Sort Order.....	115
Forcing a Binary Sort in SQL.....	115
Sorting in COBOL.....	116
Sorting in SQR.....	117

Chapter 9

Using Global Reporting and Data Analysis Tools.....	119
Using Language-Sensitive Queries.....	119
Using Scheduled Queries.....	120
Using PeopleSoft Cube Manager for Global Implementations.....	120
Using Cognos PowerPlay Local-Language Cubes.....	120
Using Hyperion Essbase Multiple-Language Cubes.....	121
Using the Strings Table for Language-Sensitive Text in Reports.....	122
Evaluation of PS/nVision String Variables.....	122
Using SQR for PeopleSoft Globalization.....	124
Printing for A4 Paper.....	124
Currency Precision.....	125
Date and Time Formatting.....	125
Report Translation.....	126

The PSSQR.INI and PSSQR.UNX Files.....	129
SQR Configuration for Processing International Text.....	129
SQR Configuration for Printing International Text.....	131
International Text in SQR for PeopleSoft Programs.....	139
SQR for PeopleSoft-Supported Character Set Encodings.....	142
Using PS/nVision Reporting for Global Implementations.....	154
PS/nVision Design-Time Globalization Features.....	155
PS/nVision Run-Time Language Features.....	157
International Versions of Microsoft Excel.....	159

Chapter 10

Developing Global Applications.....	161
Developing Easily Translatable Applications.....	161
Designing Global-Ready Pages.....	162

Part 3

Working With PeopleSoft Applications in Multiple Languages

Chapter 11

Using Related Language Tables.....	167
Understanding Related Language Tables.....	167
Related Language Tables Overview.....	167
Related Language Table Structure.....	170
How Related Language Tables Store Translations.....	171
How Related Language Tables Are Used.....	172
Installing PeopleSoft-Provided Translations.....	173
Understanding Translation Installation.....	173
Adding Translations to Existing Databases.....	174
Upgrading Customizations into Multilingual Databases.....	176
Swapping the Base Language.....	177
Understanding Base Language Swapping.....	177
Prerequisites.....	178
Running the Swap Audit Report.....	178
Checking Space and Resources.....	180
Running the SWAP_BASE_LANGUAGE Data Mover Command.....	180
Creating Related Language Tables.....	181
Creating Related Language Views.....	182

Understanding Related Language Views.....	182
Creating Views for One Base Table and One Related Language Table.....	183
Creating Views for Two Base Tables and One Related Language Table.....	184
Creating Views for Two Base Tables and Two Related Language Tables.....	185

Chapter 12

Working With Language-Sensitive Application Data.....	189
Understanding Data Editing in Related Language Tables and Base Tables.....	189
Editing Data in Multiple Languages.....	190
Understanding Multi-Language Entry.....	191
Enabling Multi-Language Entry.....	191
Entering Data in Multiple Languages.....	192

Chapter 13

Implementing Bidirectional Language Support.....	195
Understanding Bidirectional Language Support.....	195
Features.....	195
Translations Provided.....	195
Platforms Supported.....	196
Accessing Bidirectional PeopleSoft Pure Internet Architecture Pages.....	196
Navigating in Bidirectional PeopleSoft Pure Internet Architecture Pages.....	197
Enabling the Hijri Calendar.....	198
Customizing PeopleSoft Pure Internet Architecture Pages for Bidirectionality.....	198
Developing HTML Areas.....	199
Working with Images.....	199

Part 4

Translating PeopleSoft Applications

Chapter 14

Modifying Terminology.....	203
Understanding Terminology Management.....	203
Terminology Management Overview.....	204
Terminology Searches.....	204
Terminology Replacement.....	205
Terminology Replacement Undo Process.....	206

Terminology Search Statuses.....	206
Prerequisites.....	207
Granting Terminology Search Privileges.....	207
Searching Terminology.....	208
Defining Search Criteria.....	208
Cloning Existing Search Criteria.....	213
Running the Search Process.....	214
Viewing and Replacing Terminology Search Results.....	215
Replacing Non-Message Text.....	215
Replacing Message Text.....	217
Running the Replace Process.....	218
Viewing and Undoing Terminology Replacement Results.....	219
Undoing Data Replacements.....	219
Undoing Text Replacements.....	220
Running the Undo Process.....	220
Reviewing Terminology Search Information.....	221
Reviewing Search Criteria.....	221
Viewing Results for Search Only Searches.....	222
Viewing Search Results for All Searches.....	222
Selecting Terminology Search Objects.....	222
Viewing Searchable Objects.....	223
Modifying Searchable Objects.....	223
Reviewing Upgrade Considerations.....	223

Chapter 15

Adding New Languages.....	225
Understanding the Addition of New Language Definitions.....	225
Adding New Language Codes to the System.....	225
Determining PeopleSoft and ISO Codes for Your Language.....	226
Determining Appropriate Non-Unicode Character Sets.....	228
Adding New Translate Values to the LANGUAGE_CD Field.....	229
Adding New Languages to the PSLANGUAGES Table.....	230
Modifying Configuration Manager Windows Resources.....	232
Modifying the PeopleSoft Pure Internet Architecture Signon Page.....	233
Adding Text and Error Properties Files for Languages.....	234
Adding Associated Images in the New Language.....	235
Adjusting Truncated Toolbar Buttons.....	235

Chapter 16

Translating PeopleTools.....	237
Understanding PeopleTools Translation.....	237
Prerequisites.....	237
Using Alternate Language DLLs.....	238
Locating Resource Directories.....	239
Translating Resource Files.....	240
Translating Resource Files.....	240
Compiling Translated Resource Files Using MAKEALT.BAT.....	241
Compile Translated Resource Files Using MAKEUNIX.BAT.....	243
Translating Objects.....	243
Translating Properties Files.....	243
Translating Images.....	244

Chapter 17

Translating Application Definitions.....	245
Understanding Application Definition Translation.....	245
Using the Translation Designer.....	245
Opening Translation Designer.....	246
Using Translation Designer Display Options.....	247
Working in Translation Designer.....	249
Using the Translation Pages.....	252
Understanding the Translation Pages.....	252
Translating Page Definitions.....	253
Translating Messages.....	254
Translating Menus.....	255
Translating Business Processes.....	258
Translating PeopleSoft Process Scheduler.....	260
Translating Mobile Data.....	261
Translating Portals.....	265
Translating Trees.....	266
Translating Records.....	268
Translating Queries.....	268
Translating Strings.....	270
Translating Application Engine Descriptions.....	271
Translating Review Page Text.....	272
Translating Application Data.....	272
Translating HTML Definitions.....	274
Translating PeopleSoft Integration Broker Definitions.....	275

Translating XML Publisher Definitions.....	276
Translating ACE Analytic Model ID Definitions.....	277
Translating Time Zone Labels.....	278

Appendix A

Converting PeopleSoft Systems to Unicode Databases.....	281
Understanding Converting PeopleSoft Systems to Unicode Databases.....	281
Understanding Planning Unicode Conversions.....	281
Platform Support.....	281
Database Sizing.....	282
Coexistence of Unicode and Non-Unicode Databases.....	282
Converting to Unicode on Oracle Databases.....	283
Understanding Converting PeopleSoft 8.x Systems on Oracle Databases to Unicode.....	283
Exporting Database Table Structures Using PeopleSoft Data Mover.....	283
Exporting Database Contents Using the Oracle Export Utility.....	284
Creating New Oracle8i Database Instances Using the UTF-8 Character Set.....	285
Pre-Creating Unicode Database Table Structures Using PeopleSoft Data Mover.....	286
Importing Database Contents Using the Oracle Import Utility.....	286
Specifying Unicode Databases in PeopleSoft Systems.....	287
Running GRANT.SQL.....	288
Rerunning Oracle Database Imports.....	288
Converting to Unicode on Microsoft SQL Server and DB2/UDB Databases.....	288
Understanding Converting to Unicode on Microsoft SQL Server and DB2/UDB Databases.....	288
Exporting PeopleSoft Databases Using PeopleSoft Data Mover.....	288
Creating New SQL Server or DB2/UDB PTSYS Unicode Databases.....	289
Importing Database Contents Using PeopleSoft Data Mover.....	289
Specifying Unicode Databases in PeopleSoft Systems.....	289
Rerunning GRANT.SQL.....	290
Rerunning Microsoft SQL Server and DB2/UDB Database Imports.....	290

Glossary of PeopleSoft Enterprise Terms.....	291

Index	317

About This PeopleBook

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

This preface discusses:

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

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

PeopleSoft Enterprise Application Prerequisites

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

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

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

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

Application Fundamentals

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

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

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

Documentation Updates and Printed Documentation

This section discusses how to:

- Obtain documentation updates.
- Download and order printed documentation.

Obtaining Documentation Updates

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

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

See Also

Oracle's PeopleSoft Customer Connection, http://www.oracle.com/support/support_peoplesoft.html

Downloading and Ordering Printed Documentation

In addition to the complete line of documentation that is delivered on your PeopleBook CD-ROM, Oracle makes PeopleSoft Enterprise documentation available to you via Oracle's website. You can:

- Download PDF files.
- Order printed, bound volumes.

Downloading PDF Files

You can download PDF versions of PeopleSoft Enterprise documentation online via the Oracle Technology Network. Oracle makes these PDF files available online for each major release shortly after the software is shipped.

See Oracle Technology Network, <http://www.oracle.com/technology/documentation/psftent.html>.

Ordering Printed, Bound Volumes

You can order printed, bound volumes of selected documentation via the Oracle Store.

See Oracle Store, http://oraclestore.oracle.com/OA_HTML/ibeCCtpSctDspRte.jsp?section=14021

Additional Resources

The following resources are located on Oracle's PeopleSoft Customer Connection website:

Resource	Navigation
Application maintenance information	Updates + Fixes
Business process diagrams	Support, Documentation, Business Process Maps
Interactive Services Repository	Support, Documentation, Interactive Services Repository
Hardware and software requirements	Implement, Optimize + Upgrade; Implementation Guide; Implementation Documentation and Software; Hardware and Software Requirements
Installation guides	Implement, Optimize + Upgrade; Implementation Guide; Implementation Documentation and Software; Installation Guides and Notes
Integration information	Implement, Optimize + Upgrade; Implementation Guide; Implementation Documentation and Software; Pre-Built Integrations for PeopleSoft Enterprise and JD Edwards EnterpriseOne Applications
Minimum technical requirements (MTRs)	Implement, Optimize + Upgrade; Implementation Guide; Supported Platforms
Documentation updates	Support, Documentation, Documentation Updates
PeopleBooks support policy	Support, Support Policy
Prerelease notes	Support, Documentation, Documentation Updates, Category, Release Notes
Product release roadmap	Support, Roadmaps + Schedules
Release notes	Support, Documentation, Documentation Updates, Category, Release Notes
Release value proposition	Support, Documentation, Documentation Updates, Category, Release Value Proposition
Statement of direction	Support, Documentation, Documentation Updates, Category, Statement of Direction
Troubleshooting information	Support, Troubleshooting
Upgrade documentation	Support, Documentation, Upgrade Documentation and Scripts

Typographical Conventions and Visual Cues

This section discusses:

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

Typographical Conventions

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

Typographical Convention or Visual Cue	Description
Bold	Indicates PeopleCode function names, business function names, event names, system function names, method names, language constructs, and PeopleCode reserved words that must be included literally in the function call.
<i>Italics</i>	Indicates field values, emphasis, and PeopleSoft Enterprise or other book-length publication titles. In PeopleCode syntax, italic items are placeholders for arguments that your program must supply. We also use italics when we refer to words as words or letters as letters, as in the following: Enter the letter <i>O</i> .
KEY+KEY	Indicates a key combination action. For example, a plus sign (+) between keys means that you must hold down the first key while you press the second key. For ALT+W, hold down the ALT key while you press the W key.
Monospace font	Indicates a PeopleCode program or other code example.
“ ” (quotation marks)	Indicate chapter titles in cross-references and words that are used differently from their intended meanings.
. . . (ellipses)	Indicate that the preceding item or series can be repeated any number of times in PeopleCode syntax.
{ } (curly braces)	Indicate a choice between two options in PeopleCode syntax. Options are separated by a pipe ().

Typographical Convention or Visual Cue	Description
[] (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.

Visual Cues

PeopleBooks contain the following visual cues.

Notes

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

Note. Example of a note.

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

Important! Example of an important note.

Warnings

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

Warning! Example of a warning.

Cross-References

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

Country, Region, and Industry Identifiers

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

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

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

Country Identifiers

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

Region Identifiers

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

- Asia Pacific
- Europe
- Latin America
- North America

Industry Identifiers

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

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

Currency Codes

Monetary amounts are identified by the ISO currency code.

Comments and Suggestions

Your comments are important to us. We encourage you to tell us what you like, or what you would like to see changed about PeopleBooks and other Oracle reference and training materials. Please send your suggestions to your product line documentation manager at Oracle Corporation, 500 Oracle Parkway, Redwood Shores, CA 94065, U.S.A. Or email us at appsdoc@us.oracle.com.

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

Common Elements Used in PeopleBooks

As of Date	The last date for which a report or process includes data.
Business Unit	An ID that represents a high-level organization of business information. You can use a business unit to define regional or departmental units within a larger organization.
Description	Enter up to 30 characters of text.
Effective Date	The date on which a table row becomes effective; the date that an action begins. For example, to close out a ledger on June 30, the effective date for the ledger closing would be July 1. This date also determines when you can view and change the information. Pages or panels and batch processes that use the information use the current row.

Once, Always, and Don't Run	Select Once to run the request the next time the batch process runs. After the batch process runs, the process frequency is automatically set to Don't Run. Select Always to run the request every time the batch process runs. Select Don't Run to ignore the request when the batch process runs.
Process Monitor	Click to access the Process List page, where you can view the status of submitted process requests.
Report Manager	Click to access the Report List page, where you can view report content, check the status of a report, and see content detail messages (which show you a description of the report and the distribution list).
Request ID	An ID that represents a set of selection criteria for a report or process.
Run	Click to access the Process Scheduler request page, where you can specify the location where a process or job runs and the process output format.
SetID	An ID that represents a set of control table information, or TableSets. TableSets enable you to share control table information and processing options among business units. The goal is to minimize redundant data and system maintenance tasks. When you assign a setID to a record group in a business unit, you indicate that all of the tables in the record group are shared between that business unit and any other business unit that also assigns that setID to that record group. For example, you can define a group of common job codes that are shared between several business units. Each business unit that shares the job codes is assigned the same setID for that record group.
Short Description	Enter up to 15 characters of text.
User ID	An ID that represents the person who generates a transaction.

Global Technology Preface

This preface provides an overview of the contents of the *Enterprise PeopleTools 8.49 PeopleBook: Global Technology*.

PeopleTools Global Technology

This PeopleBook discusses PeopleTools features that are critical to a global implementation of PeopleSoft applications. It is divided into four parts:

- PART 1: Getting Started.

This part provides an overview of the setup and maintenance tasks that you may need to perform to implement PeopleSoft applications for a multilanguage, multinational environment.

- PART 2: Configuring Your System for Global Implementation.

This part discusses some of the technological considerations to take into account if you are implementing in multiple countries (even if you only use English language data).

- PART 3: Working with PeopleSoft Applications in Multiple Languages.

This part discusses tasks to perform if you are using the non-English translations delivered by the PeopleSoft system and processing application data in multiple languages.

- PART 4: Translating PeopleSoft Applications.

This part discusses tasks to perform if you need to modify delivered PeopleSoft terminology, translate objects into languages not supplied by the PeopleSoft system, translate products not translated by the PeopleSoft system, or translate your customizations.

PART 1

Getting Started

Chapter 1

Getting Started with Global Technology

CHAPTER 1

Getting Started with Global Technology

This chapter provides an overview of PeopleTools Global Technology and discusses how to implement PeopleTools in a multilanguage, global environment.

PeopleTools Global Technology Overview

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, 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.

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 Chapter 4, “Selecting and Configuring Character Sets and Language Input and Output,” page 39.
If required, convert your existing non-Unicode database to Unicode.	See Chapter 4, “Selecting and Configuring Character Sets and Language Input and Output,” page 39. See also the special Unicode database conversion documentation available on PeopleSoft Customer Connection.
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 Chapter 8, “Sorting in PeopleTools,” page 111.
Install the PeopleSoft-delivered translations that you have licensed, 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 Chapter 11, “Using Related Language Tables,” Installing PeopleSoft-Provided Translations, page 173. 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 licensed 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 Chapter 11, “Using Related Language Tables,” page 167.

Step	Reference
Set up time zones and generate query offsets. Note. You can also maintain some time zone settings after installation.	See Chapter 3, “Setting and Maintaining Time Zones,” page 27.
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 Chapter 5, “Controlling Currency Display Format.” Using System-Wide Multicurrency Settings, page 65. See Chapter 5, “Controlling Currency Display Format.” Resizing Currency Fields by Using the International Field Size Utility, page 67.
If you are running a Unicode database, convert any COBOL programs to a Unicode environment by running the COBOL conversion utility.	See Chapter 6, “Running COBOL in a Unicode Environment,” page 69.

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 Chapter 2, “Controlling International Preferences.” Setting Up Language Preferences in the PeopleSoft Pure Internet Architecture, page 12. See Chapter 2, “Controlling International Preferences.” Applying Browser Language Preferences, page 15. See Chapter 2, “Controlling International Preferences.” Setting Up Language Preferences for Microsoft Windows-Based PeopleTools, page 17.
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 Chapter 2, “Controlling International Preferences.” Setting Up Locale-Based Formatting for the PeopleSoft Pure Internet Architecture, page 19.

Step	Reference
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 Chapter 12, “Working With Language-Sensitive Application Data,” Editing Data in Multiple Languages, page 190.
If you use Microsoft Excel 97, select language-sensitive preferences for downloads from PeopleSoft application pages and queries to Microsoft Excel.	See Chapter 9, “Using Global Reporting and Data Analysis Tools,” International Versions of Microsoft Excel, page 159.

Translation Tasks

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

Step	Reference
Modify PeopleSoft-delivered terminology and translations to meet local requirements.	See Chapter 14, “Modifying Terminology,” page 203.

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 Chapter 15, “Adding New Languages,” page 225.
Translate PeopleTools objects into non-delivered languages or modify PeopleSoft-delivered translations of PeopleTools objects.	See Chapter 16, “Translating PeopleTools,” page 237.
Translate PeopleSoft application definitions into non-delivered languages (or modify delivered PeopleSoft translations) using the Translation Designer feature, Translation pages, and PeopleSoft Application Designer.	See Chapter 17, “Translating Application Definitions,” page 245.

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
Control time zone display.	See Chapter 3, “Setting and Maintaining Time Zones,” page 27.

Step	Reference
Validate field lengths for the character sets you will use.	See Chapter 4, “Selecting and Configuring Character Sets and Language Input and Output,” Understanding Character Sets, page 39. See Chapter 4, “Selecting and Configuring Character Sets and Language Input and Output,” Validating Field Length, page 58.
Design appropriate field formats for character sets you will use.	See Chapter 4, “Selecting and Configuring Character Sets and Language Input and Output,” Handling International Characters in PeopleTools, page 61.
Set up currency amount and currency control fields.	See Chapter 5, “Controlling Currency Display Format,” page 63.
Use appropriate sorts in the objects that you develop.	See Chapter 8, “Sorting in PeopleTools,” page 111.
Design reports and configure reporting tools for global requirements.	See Chapter 9, “Using Global Reporting and Data Analysis Tools,” page 119. 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 Chapter 11, “Using Related Language Tables,” page 167.

PeopleSoft also provide a discussion of best practices for developing easily translatable applications.

See [Chapter 10, “Developing Global Applications,” page 161.](#)

PeopleTools delivers a number of globalization-related PeopleCode functions.

See *Enterprise PeopleTools 8.49 PeopleBook: PeopleCode Language Reference*, “PeopleCode Built-in Functions”.

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 PeopleSoft Customer Connection, PeopleSoft curriculum guides, and other PeopleBooks.

See Also

Enterprise PeopleTools 8.49 PeopleBook: Getting Started with PeopleTools

PART 2

Configuring Your System for Global Implementation

Chapter 2
Controlling International Preferences

Chapter 3
Setting and Maintaining Time Zones

Chapter 4
Selecting and Configuring Character Sets and Language Input and Output

Chapter 5
Controlling Currency Display Format

Chapter 6
Running COBOL in a Unicode Environment

Chapter 7
Running COBOL in a z/OS Unicode Environment

Chapter 8
Sorting in PeopleTools

Chapter 9
Using Global Reporting and Data Analysis Tools

Chapter 10

Developing Global Applications

CHAPTER 2

Controlling International Preferences

This chapter provides an overview of international preferences and discusses how to:

- Set up language preferences in the PeopleSoft Pure Internet Architecture.
- Apply browser language preferences.
- Set up language preferences for Windows-based PeopleTools.
- Set up locale-based formatting for the PeopleSoft Pure Internet Architecture.
- Set up locale-based formatting for Windows-based PeopleTools.

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 signon 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 Microsoft Windows client PeopleTools. 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 provides an overview of the session language and discusses how to:

- Choose a session language.
- Change the session language while signed in.
- Change the session language programmatically.
- Determine the session language when the sign-in screen is bypassed.
- Control the user profile language preference.
- Determine the language of help documentation.

See Also

Chapter 11, “Using Related Language Tables,” page 167

Understanding the Session Language

In the PeopleSoft Pure Internet Architecture, the *session language* is the language selected by the user on the signon 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 code 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. If a signon page displays in a language that is not installed on the system, delete the language from the PSLANGUAGES table.

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, Utilities, Maintain 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 should always be available.

See Also

[Chapter 15, “Adding New Languages,” page 225](#)

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, Utilities, International, Preferences.
The International Preferences page appears.
2. Select a language in the Language Preference field.
3. Save the page.

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 code 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.

See Also

Enterprise PeopleTools 8.49 PeopleBook: PeopleCode Language Reference, “PeopleCode Built-in Functions,” `SetLanguage`

Enterprise PeopleTools 8.49 PeopleBook: PeopleCode Language Reference, “System Variables,” `%Language`

Enterprise PeopleTools 8.49 PeopleBook: PeopleCode Language Reference, “System Variables,” `%Language_Base`

Enterprise PeopleTools 8.49 PeopleBook: PeopleCode Language Reference, “System Variables,” `%Language_User`

Determining the Session Language When the Signon 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 code within a direct query string using the parameter `&languageCd=<target language>`.
- Single signon.

You need to select a language preference only once during the single signon.

All subsequent PeopleSoft sessions that are controlled by the single signon use the same language preference.

See Also

Enterprise PeopleTools 8.49 PeopleBook: Security Administration, “Implementing 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 Language report (SQR), 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 email 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, Security, User Profiles, 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 Code dropdown list box, select a language.
4. Save the page.

To set your user profile language preference:

1. From the menu pagelet, select My System Profile.
The General Profile Information page appears.
2. In the Personalization section, choose a language in the My preferred language for reports and email is drop-down list box.
3. Save the page.

See Also

Chapter 12, “Working With Language-Sensitive Application Data,” page 189

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 the Windows client, 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 provides an overview of browser language preferences and discusses how to:

- Set browser language preferences.
- Use browser language settings to determine 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 signon 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 Preferences dialog box. You can add new languages using the Add Language dialog box.

The representation of language codes 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.

Netscape and 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 language codes used by the browser are mapped to PeopleSoft three-letter language codes. The PeopleSoft system reads the PSLANGUAGES table to perform this mapping.

As delivered, the PeopleSoft system includes the following mappings:

PeopleSoft Language Code	Description	Corresponding ISO Locales
ARA	Arabic	ar
CFR	Canadian French	fr_CA
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

PeopleSoft Language Code	Description	Corresponding ISO Locales
KOR	Korean	ko
NOR	Norwegian (Bokmål)	no
POL	Polish	pl
POR	Portuguese	pt
RUS	Russian	ru
SVE	Swedish	sv
THA	Thai	th
TUR	Turkish	tr
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 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, 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.

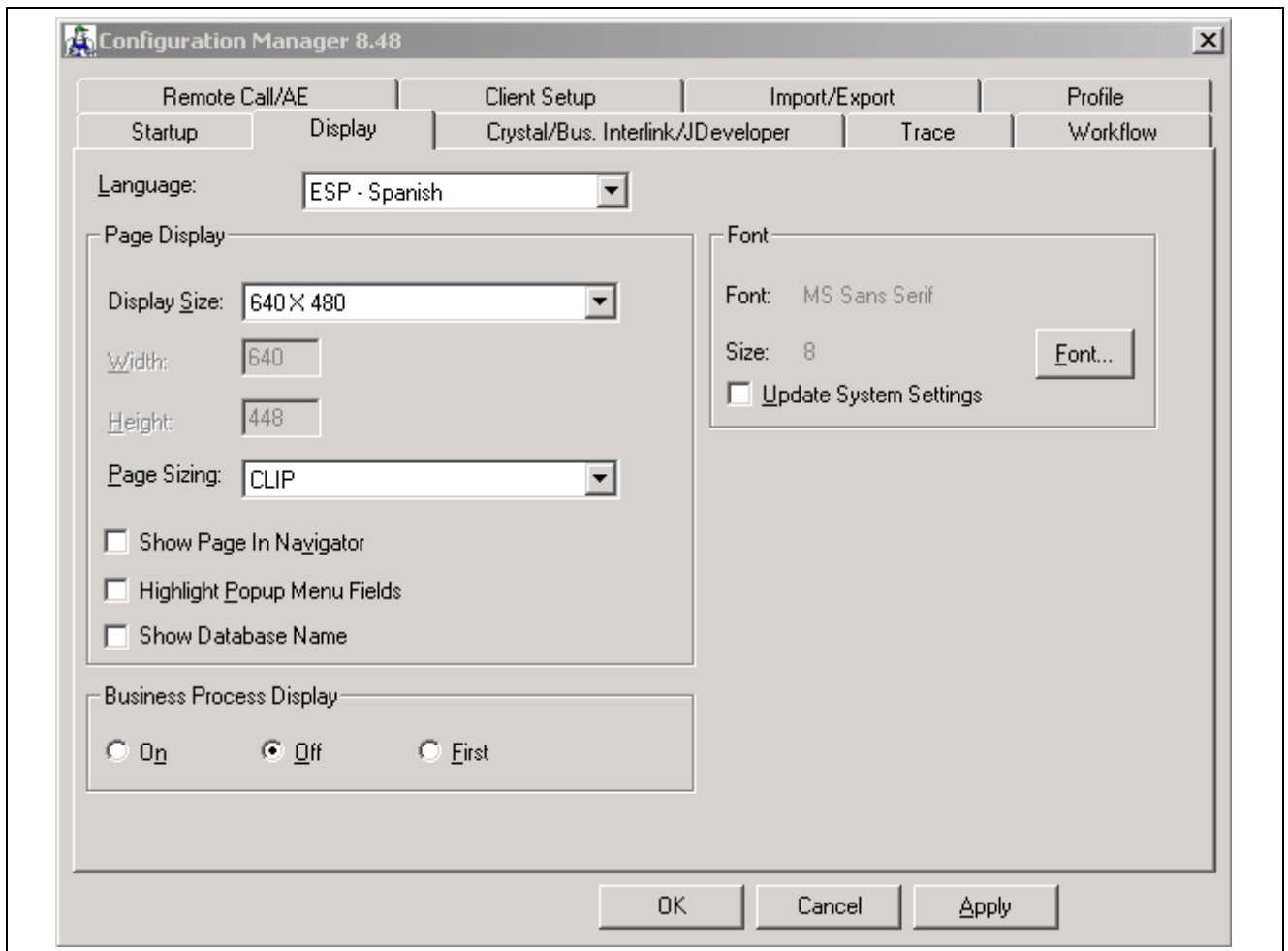
See Also

Chapter 11, “Using Related Language Tables,” page 167

Controlling PeopleSoft Configuration Manager Language Settings

The language setting in PeopleSoft Configuration Manager is maintained on each Microsoft Windows client 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*.



Configuration Manager dialog box: Display tab

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 provides an overview of locale-based formatting drivers and discusses how to:

- Define user-specific formatting.
- Define locales for locale-specific formatting.
- Define locale-specific formatting.
- Apply system-wide default formatting.

See Also

Chapter 5, “Controlling Currency Display Format,” page 63

Enterprise PeopleTools 8.49 PeopleBook: Security Administration, “Managing PeopleSoft Personalizations”

Understanding Locale-Based Formatting Drivers

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.
- Locale-specific.
- 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 inexperienced users and users who don't have 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. 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 French language. If locale-specific settings for French 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, might prefer another type of formatting for number separators. To register a user-specific preference, the user opens the My Personalizations - Personalizations 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 a custom value, regardless of language choice. All other preferences are read from the locale-specific personalizations.

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.

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

You use the following pages to manage locale-based personalization options:

Page Name	Navigation	Usage
Locale Definition	PeopleTools, Personalization, Locales	Control the locales for which you can specify defaults.
Locale Defaults	PeopleTools, Personalization, Locale Defaults	Specify Personalization Option defaults for locales appearing on the Locale Definition page.

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.

Pages Used to Set Up PeopleSoft Pure Internet Architecture Locale-Based Formatting

Page Name	Object Name	Navigation	Usage
My Personalizations - Personalizations (Option Category: Regional Settings)	PSUSERSELFPRSNL	My Personalizations. Click the Personalize Option button for <i>Regional Settings</i> .	End users access this page to view and modify their personal locale-based formatting preferences.
Locale Definition	PSLOCALEDEFN	PeopleTools, Personalization, Locales	Control the locales for which you can specify defaults.
Locale Defaults	PSLOCALEOPTNDFLTS	PeopleTools, Personalization, Locale Defaults	Specify personalization option defaults for locales appearing on the Locale Definition page. Locale-based personalizations enable PeopleTools to intelligently determine appropriate data formatting based on the language of the user logging in to PeopleTools if they have not specified individual user-based overrides for each formatting option.
Define Personalizations	PSUSEROPTNDEFN	PeopleTools, Personalization, Personalization Options	Set up system-wide locale-based formatting defaults. These defaults are applied to a user session if no locale-specific defaults or user-specific defaults have been specified for an option.

Defining User-Specific Formatting

Access the My Personalizations - Personalizations page.

Option Category: Regional Settings

Personalizations			Find	First	1-10 of 10	Last
Personalization Option	Default Value	Override Value				
Afternoon designator (PM, pm)	PM		Explain			
Decimal Separator	.		Explain			
Date Format	MMDDYY		Explain			
Date Separator	/		Explain			
Local Time Zone	Pacific Time (US), Tijuana		Explain			
Morning designator (AM, am)	AM		Explain			
Time Format	12 hour clock		Explain			
Time Separator	:		Explain			
Digit Group Separator	,		Explain			
Use Local Timezone	No		Explain			

[Restore Category Defaults](#)

My Personalizations - Personalizations page: Regional Settings

Use this 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 or all 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, simply 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 on based on your language code, 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.

See Also

Enterprise PeopleTools 8.49 PeopleBook: Security Administration, “Managing PeopleSoft Personalizations,” Working with the My Personalizations Interface

Enterprise PeopleTools 8.49 PeopleBook: Security Administration, “Managing PeopleSoft Personalizations,” Understanding Regional Settings

Defining Locales for Locale-Specific Formatting

Access the Locale Definition page.

Locale Definition		
Customize Find View 100 First 1-25 of 142 Last		
*Locale Code	*Description	
af	Afrikaans	+ -
ar	Arabic	+ -
ar-ae	Arabic (U.A.E.)	+ -
ar-bh	Arabic (Bahrain)	+ -
ar-dz	Arabic (Algeria)	+ -
ar-eg	Arabic (Egypt)	+ -
ar-iq	Arabic (Iraq)	+ -
ar-jo	Arabic (Jordan)	+ -
ar-kw	Arabic (Kuwait)	+ -
ar-lb	Arabic (Lebanon)	+ -
ar-ly	Arabic (Libya)	+ -

Local Definition page

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 even a default level of locale-sensitive data formatting.

For example, English is spoken in several countries that radically disagree on date formatting. In the U.S., dates are typically formatted as month, day, year, whereas in the United Kingdom and elsewhere, they are typically formatted as day, month, year. Therefore if PeopleTools only has a sign-in language of English, and no country information, it could not determine which of the two possible date formats to use.

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.

Defining Locale-Specific Formatting

Access the Locale Defaults page.

Locale Defaults				
Customize Find View 100 First 1-25 of 670 Last				
*Option Category Level	*User Option	*Locale Code	Description	*Override Value
Tools	Afternoon designator (PM, prr)	af	Afrikaans	nm
Tools	Afternoon designator (PM, prr)	en	English	PM
Tools	Afternoon designator (PM, prr)	en-au	English (Australia)	PM
Tools	Afternoon designator (PM, prr)	en-bz	English (Belize)	PM
Tools	Afternoon designator (PM, prr)	en-ca	English (Canada)	PM
Tools	Afternoon designator (PM, prr)	en-nz	English (New Zealand)	PM
Tools	Afternoon designator (PM, prr)	en-ph	English (Philippines)	PM
Tools	Afternoon designator (PM, prr)	en-tt	English (Trinidad)	PM

Local Defaults page

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, Personalization, Personalization Options, and click the Format tab.

See Also

Enterprise PeopleTools 8.49 PeopleBook: Security Administration, "Managing PeopleSoft 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 PeopleSoft 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.

See Also

Chapter 5, “Controlling Currency Display Format,” page 63

CHAPTER 3

Setting and Maintaining Time Zones

This chapter provides an overview of time zones and discusses how to:

- Set the base time zone.
- Maintain time zones.
- Update time zones to reflect the Energy Policy Act of 2005.
- Control the time zone display when you develop PeopleSoft application objects.

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 universal coordinated time (UTC, also known as Greenwich mean time or GMT), a time entered as 10 a.m. in U.S. eastern standard time (EST) is stored as 3 p.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 3 p.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 savings time observances. When a time zone begins or ends in daylight savings time, both the description of the zone and the times that are associated with that zone are adjusted to reflect the change. This approach ensures chronological consistency throughout an organization's geographically dispersed locations, while allowing for flexibility in how users see times.

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.

Rows become effective when midnight passes on the effective date according to the base time zone of the system.

- Internal processing in PeopleCode and all batch processes.
- Display on pages and in reports, if no other time zone information is specified or available.

User Time Zone Details

The user time zone is read from the operating system of the user's web browser in the PeopleSoft Pure Internet Architecture.

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.

Setting the Base Time Zone

Select PeopleTools, Utilities, Administration, PeopleTools Options to access the PeopleTools Options page.

PeopleTools Options

Environment Long Name: Environment Short Name:

System Type:

Language Settings

Language Code: *Sort Order Option:

☐ Translations Change Last Update

General Options

Background Disconnect Interval: 30 ☐ Multi-Company Organization ☒ Multi-Currency ☒ Use Business Unit in nVision ☒ Use Secure Rep Rqst in nVision ☐ Multiple Jobs Allowed ☒ Allow DB Optimizer Trace ☒ Grant Access ☒ Platform Compatibility Mode ☐ Allow NT batch when CCSID<>37 ☐ Save Error is Fatal ☐ Set Focus on Save Button *Case Insensitive Searching: On - CaseSensitive Default Off Style Sheet Name: PSSTYLEDEF Branding Application Package: PT_BRANDING Branding Application Class: BrandingBase	Temp Table Instances (Total): Temp Table Instances (Online): *Maximum App Message Size: 10,000,000 Base Time Zone: PST Last Help Context # Used: 100222 *Data Field Length Checking: Others *Maximum Attachment Chunk Size: 28,000 Upgrade Project Commit Limit: 50 *Enable Switch User: All

PeopleTools Options page

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 [Chapter 3, “Setting and Maintaining Time Zones,” Maintaining Time Zones, page 30](#).
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.

Maintaining Time Zones

This section provides an overview of time zone maintenance and discusses how to:

- View time zone definitions.
- Generate time zone offsets.
- View daylight savings start and end dates.

Understanding Time Zone Maintenance

PeopleSoft delivers an initial list of time zones; including information about daylight savings time observances. The PeopleSoft system does not maintain time zone data. You are responsible to ensure the time zone data is kept updated. Use the Time Zone component to manage time zone data.

Viewing Time Zone Definitions

Select PeopleTools, Utilities, International, Time Zones to access the Time Zone Data page.

Time Zone Data

Daylight Savings

Generate Query Offsets...

Time Zone Data

Customize | Find | View All |  First  1-10 of 18  Last

Time Zone

Daylight Savings

*Time Zone	Time Zone Description	Time Zone Standard Label	Time Zone Daylight Label	Universal Time Coord Offset		
ACST	Australian Central Standard Time	ACST	ACDT	570		
AEST	Australian Eastern Standard Time	AEST	AEDT	600		
AKST	Alaska Time (US)	AKST	AKDT	-540		
AST	Atlantic Time (Canada)	AST	ADT	-240		
AWST	Australian Western Standard Time	AWST	AWDT	480		
CET	Central European Time	CET	CEST	60		
CST	Central Time (US)	CST	CDT	-360		
EET	Eastern European Time	EET	EEST	120		
EST	Eastern Time (US)	EST	EDT	-300		
GMT	Greenwich Mean Time	GMT	BST			

 Save

 Notify

Time Zone Data | [Daylight Savings](#)

Time Zone Data: Time Zone tab

Time Zone Tab

Time Zone Standard Label Displays the label that appears for the time zone during standard time.

Time Zone Daylight Label Displays the label that appears for the time zone during daylight savings time.

Universal Time Coord Offset Displays the number of minutes the time is offset from UTC.

(universal coordinated time offset) 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.

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 Data page to populate the PSTZOFFSET table with offsets for all the time zones and their daylight savings time periods for a specified period of time. This makes the time zone information available in a format that can be easily accessed with SQL. This may be useful for 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 Savings Tab

Select the Daylight Savings tab to view daylight savings offset information.

Time Zone Data: Daylight Savings

Generate Query Offsets...

*Time Zone	Time Zone Description	Observes Daylight Savings Time	Daylight Savings Offset	Daylight Saving Time Start	Daylight Savings Time End
ACST	Australian Central Standard Time	☒	60	2LastSunOct	2LastSunMar
AEST	Australian Eastern Standard Time	☒	60	2LastSunOct	2LastSunMar
AKST	Alaska Time (US)	☒	60	2FirstSunApr	2LastSunOct
AST	Atlantic Time (Canada)	☒	60	2FirstSunApr	2LastSunOct
AWST	Australian Western Standard Time	☐			
CET	Central European Time	☒	60	2LastSunMar	3LastSunOct
CST	Central Time (US)	☒	60	2FirstSunApr	2LastSunOct
EET	Eastern European Time	☒	60	2LastSunMar	3LastSunOct
EST	Eastern Time (US)	☒	60	2FirstSunApr	2LastSunOct
GMT	Greenwich Mean Time	☒	60	2FirstSunApr	2LastSunOct

Save Notify

Time Zone Data | Daylight Savings

Time Zone Data: Daylight Savings tab

Daylight Savings Offset

Displays the number of minutes by which the time is offset during daylight savings time.

Daylight Saving Time Start

Displays the day that daylight savings time begins. This field prompts against values from the Daylight Savings page.

Daylight Savings Time End

Displays the day that daylight savings time ends. This field prompts against values from the Daylight Savings page.

Generating Time Zone Offsets

Click the Generate Query Offsets button on the Time Zone Data page to access the Time Zone Offset Generation 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.

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

TIMEZONE	STARTDATETIME	ENDDATETIME	BASEOFFSET	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

PSTZOFFSET table

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 savings time, but PST hasn't changed yet. The label for EST during this period is *EDT*, for eastern daylight time.

Viewing Daylight Savings Start and End Dates

Select PeopleTools, Utilities, International, Time Zones, Daylight Savings to access the Daylight Savings page.

Time Zone Data
Daylight Savings

Daylight Savings Detail
Customize | Find | View All |
First 1-10 of 17 Last

*Daylight Savings Time ID	Absolute	Month	Day	Day of the Week	Hour	Minute	Description		
0LastSunMar	☐	Mar	5	Sunday	0	0	Last Sunday in March, midnight	+	-
1LastSunSep	☐	Sep	5	Sunday	1	0	Last Sunday in Sept, 1:00am	+	-
22ndSunFeb	☐	Feb	2	Sunday	2	0	Second Sunday in Feb, 2:00am	+	-
24thTueSep	☐	Sep	4	Tuesday	2	0	Fourth Tuesday in Sept, 2:00am	+	-
2FirstFriMay	☐	May	1	Thursday	2	0	First Friday in May, 2:00am	+	-
2FirstSunApr	☐	Apr	1	Sunday	2	0	First Sunday in April, 2:00am	+	-
2FirstSunMar	☐	Mar	1	Sunday	2	0	First Sunday in March, 2:00am	+	-
2FirstSunOct	☐	Oct	1	Sunday	2	0	First Sunday in Oct, 2:00am	+	-
2LastSunMar	☐	Mar	5	Sunday	2	0	Last Sunday in March, 2:00am	+	-
2LastSunOct	☐	Oct	5	Sunday	2	0	Last Sunday in October, 2:00am	+	-

Daylight Savings page

Daylight Savings Time ID Displays a unique identifier for the date.

Absolute Clear this check box to indicate a relative date, such as *Last Sunday in October*.

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.

The number 5 for a relative date indicates the last occurrence of the specified day of the week, even if that day of the week occurs only four times in the month.

Complying with the Energy Policy Act of 2005

As a result of the Energy Policy Act of 2005, Daylight Saving Time will start earlier and end later, beginning in 2007. In the US and Canada, Daylight Saving Time will start the second Sunday in March and end the first Sunday in November.

To comply with this change, you need to add two new Daylight Saving Time IDs, set your affected time zones to use the new Daylight Saving Time IDs, and regenerate your offset data. Depending on your business needs, you may also want to define new time zones and associate them with Daylight Saving Time 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 Daylight Savings Time IDs

To add new Daylight Savings Time IDs:

1. Go to PeopleTools, Utilities, International, Time Zones, Daylight Savings
2. Click the Add button and add the following information:

Option	Value
Daylight Savings Time ID	22ndSunMar
Absolute	Unchecked
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:

Option	Value
Daylight Savings Time ID	2FirstSunNov
Absolute	Unchecked
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. Go to PeopleTools, Utilities, International, Time Zones, Time Zone Data, Daylight Savings
2. Choose the Daylight Saving Time Start and Daylight Saving Time End IDs for your affected time zones.

In the US and Canada, the affected time zones that PeopleTools delivers are the following:

- AKST Alaska Time (US)
- AST Atlantic Time (Canada)
- CST Central Time (US)
- EST Eastern Time (US)
- MST Mountain Time (US)

- NST Newfoundland Time (Canada)
 - PST Pacific Time (US)
3. Enter start and end dates for the time range to generate, for instance, 1/1/2007 to 1/1/2008, 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.

Note. To generate offsets earlier than 2007, set the time zone data accordingly first, then update the time zone data to the new values for offsets later than 2007. Do not generate offsets for a Start Date before 2007 and an End Date after 2007 in one run, since the offset data would only be correct for one time range or the other.

See the additional time zone documentation available on PeopleSoft Customer Connection.

Define New Time Zones

You may want to define new time zones. For instance, Canada has decided to follow the US 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 Daylight Saving Time ID, to distinguish it from the PST (for PST US and Canada) with the new Daylight Saving Time ID.

After defining your new time zones, run Generate Query Offsets again so that the offsets include the new time zones.

Controlling Time Zone Display When Developing PeopleSoft Application Objects

This section provides an overview of time zone display settings and discusses how to:

- Set time zone options in record field properties.
- Set time zone options in page field properties.
- Use PeopleCode time functions.

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.

- 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 the Microsoft Windows client, 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 Microsoft Windows client time and datetime fields appear in the base time zone, but each user can override this option by selecting View, Time Display, Local Time or View, Time Display, Database Time.

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 PeopleSoft 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, 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.

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 savings. 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 savings 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 savings 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 the record, 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, unless a user selects View, Time Display in the Microsoft Windows client to display in the user's local time zone.

If you select the Specified Time Zone check box, the system converts 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

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 effect 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 savings 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.

See Also

Enterprise PeopleTools 8.49 PeopleBook: PeopleCode Language Reference, "PeopleCode Built-in Functions"

Enterprise PeopleTools 8.49 PeopleBook: PeopleCode Language Reference, "System Variables"

CHAPTER 4

Selecting and Configuring Character Sets and Language Input and Output

This chapter provides overviews of character sets and selecting character sets and discusses how to:

- Select character sets.
- Validate field length.
- Handle international characters in PeopleTools.
- Detect and convert characters.

Understanding Character Sets

This section discusses:

- Character sets.
- Common character sets.
- The Unicode standard.
- Non-Unicode character sets.
- Character sets in the PeopleSoft Pure Internet Architecture.

Character Sets

Before you install your PeopleSoft system you must select an appropriate character set for PeopleSoft client workstations, application servers, and database servers.

A *character set*, also known as a *codepage*, is an ordered set of characters in which each character is mapped to a numeric index, called a *codepoint*. This codepoint stores character data in a computer system. Many hundreds of character sets exist. Some are international standards, sanctioned by the International Organization for Standardization (ISO), some are country-specific standards, and others are not standardized at all, but are specific to a particular computer system vendor. Given the number of separate computers that are involved in a typical PeopleSoft installation, it is likely that your system 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 standardized 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.

Note. To display extended Japanese characters install JDK 1.4.2_11+.

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:

Character Set	Description and Comments	Type	PeopleSoft and Structured Query Report (SQR) Name	Oracle database management system (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 <i>oe</i> ligature.	ISO	LATIN1 or ISO_8859-1	WE8ISO8859P1	CP28591

Character Set	Description and Comments	Type	PeopleSoft and Structured Query Report (SQR) Name	Oracle database management system (DBMS) Name	Microsoft Windows Name
Microsoft Codepage 1252	Microsoft Codepage 1252 - Western European. Very similar to ISO 8859-1, except for the inclusion of additional characters. Includes the euro symbol, trademark (TM) symbol, and <i>oe</i> 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
ISO 8859-15	Western European extended Latin-9. Similar to ISO 8859-1, but contains the euro symbol, <i>oe</i> ligature, and several characters that are required for Icelandic.	ISO	LATIN9 or ISO_8859-15	WE8ISO8859P15	CP28605

Character Set	Description and Comments	Type	PeopleSoft and Structured Query Report (SQR) Name	Oracle database management system (DBMS) Name	Microsoft Windows Name
Shift-JIS	Most common Japanese character set. Defines thousands of characters for writing Japanese.	Country (Japan)	SJIS	JA16SJIS	CP932
IBM CCSID 37	IBM Coded Character Set ID 37. Western European Multilingual EBCDIC-based character set.	Vendor (IBM)	EBCDIC	WE8EBCDIC37	CP1140

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:

- Character sets that are used by the database system include 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 required in Japanese. However, it does not include the accented Latin characters that are needed for French and German, so it is not a suitable character set for implementations that encompass Western European countries.

Given the sample list of 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 type of situation, an industry consortium of vendors devised a universal character set: the Unicode standard. This internationally recognized character set encodes every character that is required to write virtually every written language. The repertoire of characters is jointly maintained by the Unicode Consortium and ISO, and it is synchronized with ISO standard 10646: Universal Multiple-Octet Coded Character Set (UCS).

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.

Unicode Encodings

Unicode provides space for more than one million characters. To manage such a large repertoire of characters, Unicode defines 16 planes. Each plane contains 65,533 character positions. Plane 0, known as the Basic Multilingual Plane (BMP), is generally sufficient for everyday business use. The other planes are intended to encode extended or rarely used ideographic characters (such as Japanese, Chinese, and Korean ideographs), archaic scripts, and other rarely used characters, such as braille and advanced mathematical symbols.

Note. PeopleTools 8.4 supports the use of characters from the BMP only, otherwise known as plane 0. Supplementary characters are not supported in PeopleTools 8.4.

Based on the concept of 16 planes, each Unicode character has a unique, four-byte code. Using four bytes of storage per character may seem wasteful, especially considering that most applications today use a small collection of characters and that the vast majority of characters in business applications are from the BMP. Therefore, several different encoding forms for Unicode characters have been standardized and adopted by vendors who are implementing the standard.

Four encodings of Unicode are widely used. All of them are fully compatible with each other and share the same repertoire of characters. They differ, however, in how each character is represented at the byte level. These encodings are:

- Universal Transformation Form - 32-bit (UTF-32).
- Universal Transformation Form - 16-bit (UTF-16).
- Universal Transformation Form - 8-bit (UTF-8).
- Universal Character Set - 2-byte (UCS-2).

The PeopleSoft system currently supports UCS-2 and UTF-8 encodings; however, following are brief overviews of all four encodings.

Note. The PeopleSoft system supports only UTF-8 and several Asian character sets for outgoing email messages from PeopleTools application servers.

Encoding	Description
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 (a) character is represented in hexadecimal as 0x00000061. UTF-32 was formerly called UCS-4. PeopleSoft systems do not use UTF-32.

Encoding	Description
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 to be represented: a low surrogate and a high surrogate. When no supplementary characters are present, UTF-16 is identical to UCS-2.
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 1 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. PeopleTools uses UTF-8 for serving HTML pages in the PeopleSoft Pure Internet Architecture. The PeopleSoft system also uses UTF-8 in Oracle Unicode databases.
UCS-2	A 16-bit (two-byte) representation of each Unicode character. As such, it can reference only 65,535 codepoints and is limited to characters on the BMP. The PeopleSoft system uses UCS-2 in memory on the Microsoft Windows development tools, the application server, and in Unicode Microsoft SQL Server and IBM DB2 UDB for Linux, Unix, and Microsoft Windows databases.

The following table summarizes the four Unicode encodings and their uses in the PeopleSoft system:

Unicode Encoding	Minimum Bytes per Character	Maximum Bytes per Character	PeopleSoft System Usage
UTF-32	4	4	None.
UTF-16	2	4	None.

Unicode Encoding	Minimum Bytes per Character	Maximum Bytes per Character	PeopleSoft System Usage
UTF-8	1	4	PeopleSoft Pure Internet Architecture HTML pages, inbound and outbound XML, and Oracle databases.
UCS-2	2	2	In-memory Microsoft Windows client; application server; Microsoft SQL Server and IBM DB2 UDB for Linux, Unix, and Microsoft Windows databases Unicode databases.

Encoding Examples

This section includes Unicode encoding examples for the following characters:

a	Latin small letter <i>a</i> .
ñ	Latin small letter <i>n</i> .
か	Hiragana letter <i>ka</i> .

The following table shows examples of these characters in each of the Unicode encodings:

Unicode Encoding	Latin Small Letter <i>a</i>	Latin Small Letter <i>n</i> with Tilde	Hiragana Letter <i>ka</i>
UTF-32	0x00000061	0x000000f1	0x0000304b
UTF-16	0x0061	0x00f1	0x304b
UTF-8	0x61	0xc3b1	0xe3818b
UCS-2	0x0061	0x00f1	0x304b

See Also

The Unicode Standard Version 3.0, The Unicode Consortium, Reading, MA, USA, Addison-Wesley Developers Press, 2000. ISBN 0-201-61633-5

<http://www.unicode.org/unicode/uni2book/u2.html>

Chapter 4, “Selecting and Configuring Character Sets and Language Input and Output,” Selecting Email Character Sets, page 57

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:

- SBCSs.
- 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. In the context of this book, ANSI character sets refer to non-Unicode character sets, which can be any international, national, or vendor standard character set, such as those that are discussed at the beginning of this chapter.

SBCSs

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.

DBCSs

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.

The PeopleSoft system supports two types of DBCS:

- Nonshifting
- Shifting

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

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 codepage, 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.

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.

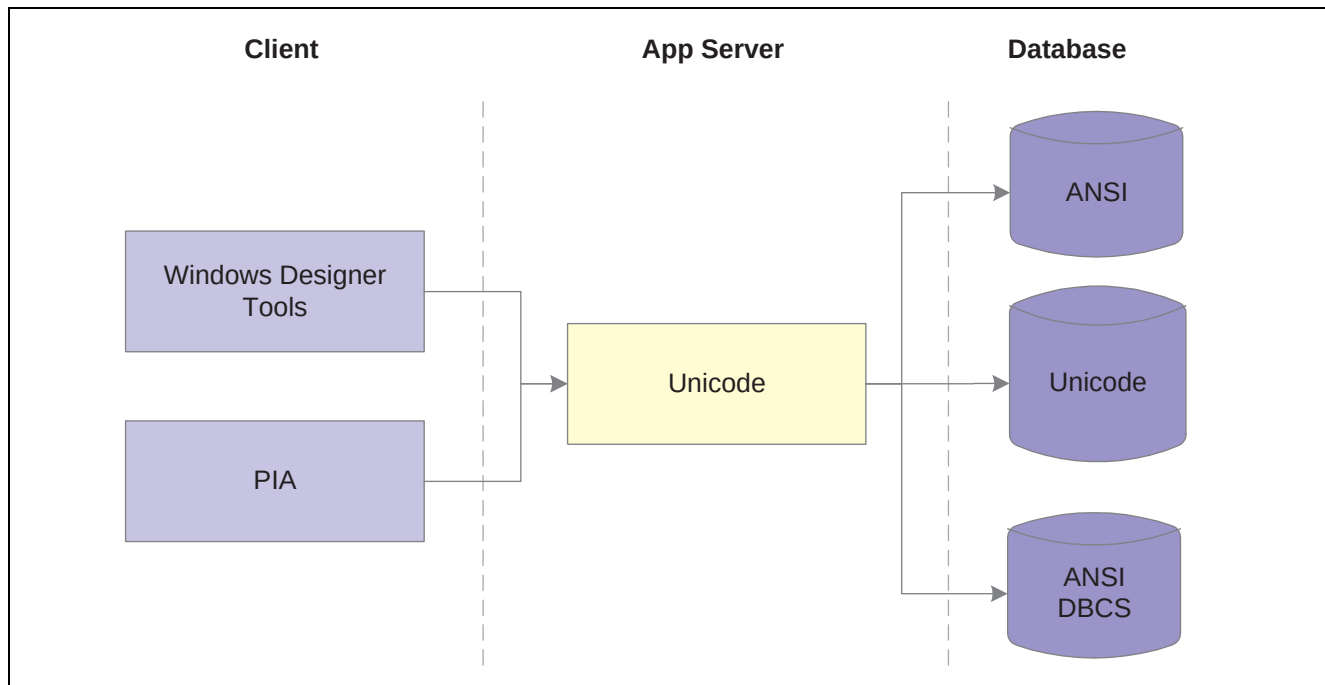
Instead of reserving a range of bytes as lead bytes, shifting DBCSs always keep track of whether they are in double-byte or single-byte mode. In double-byte mode, every two bytes form a character. In single-byte mode, every byte is a character in itself. To track what mode the character set is in, the system uses *shifting* characters. By default, the character set is expecting single-byte data. As soon as a double-byte character needs to be represented, a shift-in byte is added to the string. From this point on, all characters are expected to be two bytes per character. This continues until a shift-out byte is detected, which indicates that the character set should go back to a single byte per character.

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 requires more space to store in a shifting character set because you need to include the shift-in and shift-out bytes, as well as the data itself.

Character Sets in the PeopleSoft Pure Internet Architecture

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

This diagram illustrates the PeopleSoft Unicode architecture:



PeopleSoft Unicode architecture

PeopleSoft clients and application servers 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, so you may not be able to run all parts of your system in Unicode. Therefore, PeopleTools enables you to configure these system components to use other character sets:

- The PeopleSoft database.

Not all database platforms support Unicode data storage. Even when Unicode storage is available, not all PeopleSoft implementations require the flexibility in language choices that are available with Unicode, so you can use a non-Unicode encoding for the database.

- 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.

- File input and output.

All file operations in PeopleTools, including file layout objects, trace and log files, and file operations from SQRs 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.

- 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. Some of the products that do not yet support Unicode data are Hyperion Essbase and Cognos PowerPlay.

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 chapter.

The following table lists the character sets that the PeopleSoft system supports and the names by which they may be referred to in PeopleSoft applications. Use these character set names:

- In PeopleCode programs for manipulating file layout objects.
- 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.

A limited number of the character sets in this table are supported as database character sets. Refer to your hardware and software requirements guide for details about the character sets that are supported for your database platform.

This PeopleBook also contains information about supported character set encodings for SQL for PeopleSoft globalization.

See [Chapter 9, “Using Global Reporting and Data Analysis Tools,” SQL for PeopleSoft-Supported Character Set Encodings, page 142.](#)

See *Enterprise PeopleTools 8.49 Hardware and Software Requirements Guide*.

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. With Microsoft Windows NT, the ANSI codepage is dependent on the language version of the Microsoft Windows operating system. For example, Japanese Windows NT uses CP932 as its ANSI codepage. English Windows NT uses CP1252 as its ANSI codepage. Microsoft Windows 2000 and XP enable the system administrator to select the ANSI codepage independently of the language release of Microsoft Windows.	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	IBM EBCDIC 290 (Katakana)	SBCS
CCSID300	IBM EBCDIC 300 (Kanji)	Nonshifting DBCS

Character Set Name	Description and Comments	Character Set Type
CCSID930	IBM EBCDIC 930 (Kana-Kanji)	Shifting DBCS
CCSID935	IBM EBCDIC 935 (Simplified Chinese)	Shifting DBCS
CCSID937	IBM EBCDIC 937 (Traditional Chinese)	Shifting DBCS
CCSID939	IBM EBCDIC 939 (Latin-Kanji)	Shifting DBCS
CCSID942	IBM EBCDIC 942 (Japanese PC)	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 (US)	SBCS
CP500	Windows 500 (EBCDIC 500V1)	SBCS
CP708	Windows 708 (Arabic - ASMO708)	SBCS

Character Set Name	Description and Comments	Character Set Type
CP720	Windows 720 (Arabic - ASMO)	SBCS
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

Character Set Name	Description and Comments	Character Set Type
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	ISO-2022-JP Japanese	Shifting DBCS
ISO-2022-KR	ISO-2022-JP Korean	Shifting DBCS
ISO_8859-1	ISO 8859-1 (Latin1)	SBCS
ISO_8859-10	ISO 8859-10 (Latin6)	SBCS
ISO_8859-11	ISO 8859-11 (Thai)	SBCS
ISO_8859-14	ISO 8859-14 (Latin8)	SBCS
ISO_8859-15	ISO 8859-15 (Latin9/Latin0)	SBCS

Character Set Name	Description and Comments	Character Set Type
ISO_8859-2	ISO 8859-2 (Latin2)	SBCS
ISO_8859-3	ISO 8859-3 (Latin3)	SBCS
ISO_8859-4	ISO 8859-4 (Latin4)	SBCS
ISO_8859-5	ISO 8859-5 (Cyrillic)	SBCS
ISO_8859-6	ISO 8859-6 (Arabic)	SBCS
ISO_8859-7	ISO 8859-7 (Greek)	SBCS
ISO_8859-8	ISO 8859-8 (Hebrew)	SBCS
ISO_8859-9	ISO 8859-9 (Latin5)	SBCS
JIS_X0201	Japanese Half-width Katakana	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
UTF7	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

See Also

For more information and code charts for Microsoft codepages, visit <http://www.microsoft.com/globaldev/reference/cphome.asp>

For more information and code charts for Unicode, visit <http://www.unicode.org>

Selecting Character Sets

This section discusses how to:

- Select database character sets.
- Select application server character sets.
- Select and manage client workstation character sets.
- Select email character sets.

See Also

Chapter 4, “Selecting and Configuring Character Sets and Language Input and Output,” Understanding Character Sets, page 39

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 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.

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.

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.

See Also

Enterprise PeopleTools 8.49 Hardware and Software Requirements Guide

Enterprise PeopleTools 8.49 Installation Guide for Your Database Platform

Your operating system and database guides

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 non-Unicode character set. 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 2000 and XP enable 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 2000 or XP installation. To change the system default locale (and therefore the character set), on Microsoft Windows 2000 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.

See Also

Chapter 4, "Selecting and Configuring Character Sets and Language Input and Output," Character Sets in the PeopleSoft Pure Internet Architecture, page 47

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.
- Fonts and the Microsoft Windows client.
- 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. Most modern browsers can support the use of UTF-8 encoded HTML pages, including all browsers that are supported by PeopleTools.

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, and you may need alternative 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 Microsoft Windows client.

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 PeopleSoft 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.

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 2000 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 PeopleSoft Application Designer style sheet definitions or by following the browser-specific instructions in this section.

- Several public domain fonts exist that contain a large character repertoire for use in web browsers. For a list of such fonts, see <http://www.hclrss.demon.co.uk/unicode/fonts.html>.
- Several font foundries license fonts for individual or corporate use.

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

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

Fonts and the Microsoft Windows Client

The Microsoft Windows client enables you to specify the font that is used for all graphical components in PeopleSoft Application Designer, the Microsoft Windows client panel processor, and all other PeopleTools modules for Microsoft Windows. Three separate areas enable you to specify a font:

- 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 [Chapter 16, "Translating PeopleTools," page 237](#).

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.

Several fonts that are shipped with Microsoft Windows 2000 and Microsoft Office 2000, including Arial Unicode MS and Microsoft Sans Serif, contain a large number of glyphs covering most of the languages that are supported by Unicode. Microsoft Windows 2000 and Microsoft Windows XP 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 PeopleSoft Application Designer also enables you to select a font for character display in the editor's window itself. This is useful if the PeopleCode programs that you are working on contain Unicode characters. To set the font in PeopleSoft Application Designer, select Tools, Options and then select the PeopleCode tab.

Input Methods

If users will enter foreign language data by using PeopleSoft Pure Internet Architecture or the Microsoft Windows client, 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.

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, because many email products in Asian markets support only traditional character sets, the PeopleSoft system supports several additional character sets for outgoing email.

PeopleSoft application servers support the following character sets for outgoing email:

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

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>
```

SMTP settings

Writing Custom SMTP Encoding Modules

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

Validating Field Length

This section provides overviews of PeopleSoft 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 PeopleSoft Application Designer.

If you create a Unicode database, the field length, as shown in PeopleSoft 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, the UTF-8 character set is used by Oracle with byte semantics; therefore, 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 PeopleSoft 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 PeopleSoft 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 more than one byte storage per character.

The following tables show some of the possible database encodings for database platforms that the PeopleSoft system supports in Unicode and DBCS and their effects on database column sizes.

For Oracle with byte semantics (PeopleTools 8.48 or higher):

Database Character Set	Database Representation of a Character Field with a Length of 10 in PeopleSoft Application Designer	Number of Characters (Worst Case) Allowed in a Character Field with a Length of 10 in PeopleSoft Application Designer
Unicode (UTF-8)	VARCHAR2(30)	10
Any SBCS	VARCHAR2(10)	10
Shift-JIS (JA16SJIS)	VARCHAR2(10)	5

For Oracle with character semantics:

Database Character Set	Database Representation of a Character Field with a Length of 10 in PeopleSoft Application Designer	Number of Characters (Worst Case) Allowed in a Character Field with a Length of 10 in PeopleSoft Application Designer
Unicode (UTF-8)	VARCHAR2(20)	10
Any SBCS	VARCHAR2(10)	10
Shift-JIS (JA16SJIS)	VARCHAR2(10)	10

For Microsoft SQL Server with Varchar (PeopleTools 8.48 or higher):

Database Character Set	Database Representation of a Character Field with a Length of 10 in PeopleSoft Application Designer	Number of Characters (Worst Case) Allowed in a Character Field with a Length of 10 in PeopleSoft Application Designer
Unicode (UCS-2)	NVARCHAR(10)	10
Any SBCS	VARCHAR(10)	10
Shift-JIS (CP932)	VARCHAR(10)	5

For Microsoft SQL Server:

Database Character Set	Database Representation of a Character Field with a Length of 10 in PeopleSoft Application Designer	Number of Characters (Worst Case) Allowed in a Character Field with a Length of 10 in PeopleSoft Application Designer
Unicode (UCS-2)	NCHAR(10)	10
Any SBCS	CHAR(10)	10
Shift-JIS (CP932)	CHAR(10)	5

For DB2 UDB for OS/390 and z/OS:

Database Character Set	Database Representation of a Character Field with a Length of 10 in PeopleSoft Application Designer	Number of Characters (Worst Case) Allowed in a Character Field with a Length of 10 in PeopleSoft Application Designer
Any SBCS	CHAR(10)	10
Shifting DBCS (CCSID 930/939)	CHAR(10)	4 (4 x 2 byte characters, plus shift-in & shift-out bytes)

For all others:

Database Character Set	Database Representation of a Character Field with a Length of 10 in PeopleSoft Application Designer	Number of Characters (Worst Case) Allowed in a Character Field with a Length of 10 in PeopleSoft Application Designer
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.

For graphically sizing page fields, PeopleTools uses the Unicode length of the field as defined in PeopleSoft Application Designer. For example, if a field is defined in PeopleSoft 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 or IBM CCSID 930/939, 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 PeopleSoft 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, Utilities, Administration, 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:

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.
DB2 MBCS	Select if you are running a Japanese database on the DB2 UDB for Linux, Unix, and Microsoft Windows platform. This options enables field length checking based on a shifting DBCS.
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.

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.

3. Save the page.

Handling International Characters in PeopleTools

PeopleTools contains several features that enable you to manipulate character data based on the language, script, or type of character being processed. Some field formats, such as the Name format, are dependent on the language of the data being maintained. Several PeopleCode functions enable you to manipulate language-dependent characters.

This section discusses standard name formats for Chinese, Japanese, and Korean characters.

Standard Name Formats for Chinese, Japanese, and Korean Characters

The PeopleSoft standard name format is:

```
[lastname] [suffix],[prefix] [firstname] [middle name/initial]
```

However, if the name contains any of the following types of characters, you must separate the first and last names by a space instead of a comma:

- Chinese, Japanese, and Korean Unified Ideographs (Chinese Hanzi, Japanese Kanji, and Korean Hanja).
- Japanese half-width or full-width Katakana.
- Japanese Hiragana.
- Korean Hangul.

The following example shows the PeopleSoft standard name format when any one of the previous types of characters appears:

```
[lastname] [suffix] [prefix] [firstname] [middle name/initial]
```

This format allows for Chinese, Japanese, and Korean names, which are traditionally separated by spaces, and which would appear incorrectly if separated by a comma.

Detecting and Converting Characters

PeopleTools also provides PeopleCode string functions that recognize and convert between different characters within the Japanese character set. This enables you to detect, convert, and enforce the types of characters that you can enter in any PeopleSoft field. The PeopleSoft system uses these functions in the development of Alternate Character Architecture in some PeopleSoft applications. 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 recognize and convert between different characters within the Japanese character set:

- CharType
- ContainsCharType
- ContainsOnlyCharType
- ConvertChar

See Also

Enterprise PeopleTools 8.49 PeopleBook: PeopleCode Language Reference, “PeopleCode Built-in Functions,” CharType

CHAPTER 5

Controlling Currency Display Format

This chapter provides an overview of currency-specific settings and discusses how to:

- Set up currency amount fields when developing applications.
- Set currency field display properties when developing applications.
- Use system-wide multicurrency settings.
- Resize currency fields by using the international field size utility.

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 currency symbol is controlled by the CUR_SYMBOL field. The decimal precision for a currency is controlled by the DECIMAL_POSITIONS field.

Note. Starting with PeopleTools 8.48, the Currency Code table (CURRENCY_CD_TBL) contains sample data.

Note. The PeopleSoft Currency Code table is based on ISO Standard 4217, “Codes for the representation of currencies,” and is maintained in the Currency Code pages. To access these pages, select PeopleTools, Set Up Application, Common Definitions, Currency. Consult your application PeopleBook for more information on maintaining the Currency Code table.

The navigation to the page where you maintain currency code information depends on which application you are using. Please see the PeopleBooks for your application for navigation information.

See *Enterprise PeopleTools 8.49 Installation Guide for Your Database Platform*.

Setting Up Currency Amount Fields When Developing Applications

To set up currency amount fields:

1. Access PeopleSoft 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 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 PeopleSoft 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.

See Also

Chapter 2, “Controlling International Preferences,” Setting Up Locale-Based Formatting for the PeopleSoft Pure Internet Architecture, page 19

Chapter 5, “Controlling Currency Display Format,” Using System-Wide Multicurrency Settings, page 65

Using System-Wide Multicurrency Settings

This section provides an overview of multicurrency settings and discusses how to:

- Connect a currency control field to the Multi-Currency check box.
- Enable or disable system-wide currency 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 PeopleSoft 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.

Enabling or Disabling System-Wide Currency Settings

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

1. Select PeopleTools, Utilities, Administration, 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 PeopleSoft 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.

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, Utilities, International, Process Field Size 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
```


CHAPTER 6

Running COBOL in a Unicode Environment

This chapter provides an overview of COBOL in a Unicode environment and discusses how to:

- Run the COBOL conversion utility.
- Fine-tune COBOL programs.

See Also

[Chapter 4, “Selecting and Configuring Character Sets and Language Input and Output,” Understanding Character Sets, page 39](#)

Understanding COBOL in a Unicode Environment

This section discusses:

- Unicode encodings in PeopleSoft COBOL.
- Expanded storage space requirements.
- Special logic for parsing Unicode strings.
- COBOL sorting.
- Unicode-specific error messages.

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.

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, 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, which is the encoding that the PeopleSoft system uses in COBOL.

UTF-8 is a variable format of one to four bytes per character. Currently, the PeopleSoft system supports only one to three-byte characters. Four-byte UTF-8 characters are required only to represent characters that are outside Unicode’s Basic Multilingual Plane (BMP). In UTF-8, the actual number of bytes to store a character can be determined by the first three bits of the first, or 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 store the character:

Bit Pattern of First Byte	Character Length
0xxxxxxx	One byte
110xxxxx	Two bytes
111xxxxx	Three bytes

For example, the following graphic shows the different UCS-2 and UTF-8 encodings for two sample characters:

Character	UCS-2	UTF-8
a	= ,0x00,0x61,	= ,0x61,
学	= ,0x5B,0x66,	= ,0x5B,0x66,0x22,

UCS-2 and UTF-8 encodings

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.

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.

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.

See Also

Chapter 6, “Running COBOL in a Unicode Environment,” Running the COBOL Conversion Utility, page 72

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.

See Also

[Chapter 6, “Running COBOL in a Unicode Environment,” Identifying Unicode and Non-Unicode Data, page 83](#)

[Chapter 6, “Running COBOL in a Unicode Environment,” Defining Single Character Arrays, page 85](#)

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 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.

See Also

Enterprise PeopleTools 8.49 PeopleBook: PeopleCode Language Reference, “Meta-SQL”

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 provides an overview of the COBOL conversion utility and discusses how to:

- Run the conversion utility.
- Identify converted COBOL programs.
- Understand what is expanded.
- Use utility directives.
- View error logs.

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.

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 Chapter 6, “Running COBOL in a Unicode Environment,” Viewing Error Logs, page 81 . -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 NT or Microsoft Windows 2000

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.

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:

```
***** Converted for Unicode: <date converted>␣
***** DO NOT MODIFY THIS FILE *****␣
**** ALL SOURCE CODE CHANGES MUST BE DONE TO THE ANSI VERSION ****␣
***** FOUND IN %PS_HOME%/SRC/CBL *****
```

COBOL conversion utility comment

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 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 'OPPPPP' .
          10  FILLER          PIC X(4)     VALUE ALL 'I' .
          10  FILLER          PIC X         VALUE 'H' .
          10  FILLER          PIC X(18)    VALUE ALL 'C' .
          10  FILLER          PIC X(4)     VALUE ALL 'I' .
          10  FILLER          PIC X(4)     VALUE ALL 'N' .
          10  FILLER          PIC X(30)    VALUE ALL 'H' .
          10  FILLER          PIC X(30)    VALUE ALL 'C' .
          10  FILLER          PIC X(30)    VALUE ALL 'H' .
          10  FILLER          PIC X(10)    VALUE ALL 'C' .
          10  FILLER          PIC X(6)     VALUE 'OPPPPP' .
          10  FILLER          PIC X(8)     VALUE 'OPPPPPPP' .
          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 'OPPPPP' .
        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 'OPPPPP' .
        10 FILLER        PIC X(8)    VALUE 'OPPPPPPP' .
        10 FILLER        PIC X          VALUE 'Z' .

    05 BIND-DATA.
        10 TSE-JOBID      PIC X(30)   VALUE SPACES.
        10 TSE-PROC-INSTANCE PIC 9(10) VALUE ZERO  COMP-3.
        10 TSE-SET-NBR    PIC 9(6)    VALUE ZERO  COMP.
        10 TSE-EDIT-TYPE  PIC X(3)    VALUE SPACES.
        10 TSE-FIELDNAME  PIC X(54)   VALUE SPACES.
        10 MESSAGE-SET-NBR PIC 9(5)    VALUE ZERO  COMP.
        10 MESSAGE-NBR    PIC 9(5)    VALUE ZERO  COMP.
        10 MESSAGE-PARM   PIC X(90)   VALUE SPACES.
        10 MESSAGE-PARM2  PIC X(90)   VALUE SPACES.
        10 MESSAGE-PARM3  PIC X(90)   VALUE SPACES.
        10 BUSINESS-UNIT  PIC X(30)   VALUE SPACES.
        10 TRANSACTION-NBR PIC 9(10)   VALUE ZERO  COMP-3.
        10 SEQ-NBR        PIC 9(15)   VALUE ZERO  COMP-3.
        10 FILLER         PIC X          VALUE 'Z' .

```


Exception 2: 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.

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.

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' .
              88 PAY-PERIODS-ALL-BIWEEKLY        VALUE 'YYNN' .
              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' .
      88 PAY-PERIOD-NO          VALUE 'N' .
NOCEND
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) .
NOCLEN
      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.
      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:

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.

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:

- Identify Unicode and non-Unicode data.
- Specify column lengths in dynamic SQL.

- Define single character arrays.

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' .
```

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.

This example shows the PeopleSoft COBOL trace file, which provides information about the encoding scheme that the COBOL API used:

PeopleSoft Batch Timings Report: cobsql_0416165133.trc								
Encoding Scheme Used: Unicode								
PeopleSoft Batch Statistics (All timings in seconds)								
Statement	Retrieve Count	Time	Compile Count	Time	Execute Count	Time	Fetch Count	Time
APIBNN	0	0.00	0	0.00	266043	4.71	0	0.00
APISSB	0	0.00	0	0.00	1315	0.00	0	0.00
COMMIT	0	0.00	0	0.00	10	0.22	0	0.00
CONNECT	0	0.00	0	0.00	75	0.92	0	0.00
DISCONNECT	0	0.00	0	0.00	75	1.91	0	0.00

Sample trace file

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 hardcoded 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.
      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

```


See Also

[Chapter 6, “Running COBOL in a Unicode Environment,” Identifying Unicode and Non-Unicode Data, page 83](#)

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.
08  FIELD-DELIM      VALUE '*' .

. . .
. . .
SET CHAR-IDX  TO  1
SEARCH CHAR-ARRAY
      WHEN FIELD-DELIM(CHAR-IDX)
          SET W-OFFSET  TO  CHAR-IDX
          DISPLAY 'FIELD DELIMITER FOUND AT POSITION ' W-OFFSET
END-SEARCH
```

The intent of the code 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 is possible for the second or third byte to match the bit pattern of the delimiter character, thus falsely passing the test and ending the search loop.

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 character length of a character database type field:

- 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 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
    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 character for which the program is searching is one byte in length.

Consider the following code fragment, where the 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
    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
    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

This table lists the input parameters:

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 PTPSTRFN COBOL subroutine provides an option to map the byte length of an entire character string. The following sample COBOL code provides an example of how this feature is used. This example shows the Unicode table from which the code reads all fields:

LANGUAGE	UNITEXT
Catalan	Quan el món vol conversar, parla Unicode
Chinese (Simplified)	当世界需要沟通时, 请用Unicode !
Chinese (Traditional)	當世界需要溝通時, 請用統一碼(Unicode)
Danish	Når verden vil tale, taler den Unicode
Dutch	Als de wereld wil praten, spreekt hij Unicode
English	When the world wants to talk, it speaks Unicode
Esperanto	Kiam la mondo volas paroli, ĝi parolas Unicode
Finnish	Kun maailma haluaa puhua, se puhuu Unicodea
French	Quand le monde veut communiquer, il parle en Unicode

Sample Unicode table

For each row 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 COBOL code analyzes the text as follows:

```

01  W-WORK.
    02  LANGUAGE                      PIC X(20) .
    02  UNICODE-TEXT                  PIC X(300) .
    02  UNICODE-TEXT-MAP              PIC X(300) .
    02  DATA-LEN                     PIC 9(3)      COMP.
    02  BYTE-POS-MAX                  PIC 9(4)      COMP.
    02  COUNTERS .
        05  COUNT-1BYTE-CHAR          PIC 9(02)     VALUE ZEROS.
        05  COUNT-2BYTE-CHAR          PIC 9(02)     VALUE ZEROS.
        05  COUNT-3BYTE-CHAR          PIC 9(02)     VALUE ZEROS.
01  BYTE-ARRAY.
    02  BYTE-POS                      PIC X      OCCURS 300 TIMES
                                     INDEXED BY BYTE-IDX.
        88  ONE-BYTE-CHAR              VALUE '1' .
        88  TWO-BYTES-CHAR             VALUE '2' .
        88  THREE-BYTES-CHAR           VALUE '3' .
        88  BYTE-STRING-END            VALUE SPACE.
01  STR-FUNC                          COPY 'PTPSTRFN' .
. . .
. . .
    <Code to retrieve the text from the database and assign to the appropriate⇒
fields>
. . .
. . .
* Initialize the string map before calling the function
  MOVE SPACES      TO UNICODE-TEXT-MAP
  CALL 'PTPSTRFN'   USING ACTION-STRMAP

```

```

                                STR-FUNC
                                UNICODE-TEXT
                                UNICODE-TEXT-MAP
IF NOT STRFN-RC-OF OF STR-FUNC
    <ABEND PROGRAM>
END-IF
SET BYTE-POS-MAX TO 300
MOVE 300 TO DATA-LEN OF W-WORK
MOVE UNICODE-TEXT-MAP TO BYTE-ARRAY
PERFORM VARYING BYTE-IDX FROM 300 BY -1
    UNTIL BYTE-IDX <= 1
    OR NOT BYTE-STRING-END(BYTE-IDX)
    SUBTRACT 1 FROM DATA-LEN OF W-WORK
END-PERFORM
* Initialize counters
MOVE ZEROS TO COUNT-1BYTE-CHAR
MOVE ZEROS TO COUNT-2BYTE-CHAR
MOVE ZEROS TO COUNT-3BYTE-CHAR
PERFORM UNTIL BYTE-IDX > DATA-LEN OF W-WORK
    EVALUATE TRUE
        WHEN ONE-BYTE-CHAR
            ADD 1 TO COUNT-1BYTE-CHAR
        WHEN TWO-BYTES-CHAR
            ADD 1 TO COUNT-2BYTE-CHAR
        WHEN THREE-BYTES-CHAR
            ADD 1 TO COUNT-3BYTE-CHAR
    END-EVALUATE
    SET BYTE-IDX UP BY 1
END-PERFORM
DISPLAY ' LANGUAGE = ' LANGUAGE
DISPLAY ' UTF8 TEXT: (LENGTH = ' DATA-LEN ') '
DISPLAY ' ' UNICODE-TEXT
DISPLAY ' '
DISPLAY ' UTF8 BYTE MAPPING: '
DISPLAY ' ' UNICODE-TEXT-MAP
DISPLAY ' '
DISPLAY ' TALLY: '
DISPLAY ' NUMBER OF ONE-BYTE CHAR: '
COUNT-1BYTE-CHAR
DISPLAY ' NUMBER OF TWO-BYTES CHAR: '
COUNT-2BYTE-CHAR
DISPLAY ' NUMBER OF THREE-BYTES CHAR: '
COUNT-3BYTE-CHAR
DISPLAY ' '
DISPLAY ' '

```

The output of this program appears as follows. Note that certain strings appear to be garbled. This is because there is no font to display the characters properly.

```

LANGUAGE = Catalan
UTF8 TEXT: (LENGTH = 0041)

```

Copyright © 1988-2007, Oracle. All rights reserved.


```

        When the world wants to talk, it speaks Unicode
UTF8 BYTE MAPPING:
        1111111111111111111111111111111111111111111111111111111
TALLY:
        NUMBER OF ONE-BYTE CHAR:      047
        NUMBER OF TWO-BYTES CHAR:     000
        NUMBER OF THREE-BYTES CHAR:   000

LANGUAGE = Esperanto
UTF8 TEXT: (LENGTH = 0047)
        Kiam la mondo volas paroli, Ĉi parolas Unicode
UTF8 BYTE MAPPING:
        111111111111111111111111111111111112X111111111111111111111111
TALLY:
        NUMBER OF ONE-BYTE CHAR:      045
        NUMBER OF TWO-BYTES CHAR:     001
        NUMBER OF THREE-BYTES CHAR:   000

LANGUAGE = Finnish
UTF8 TEXT: (LENGTH = 0043)
        Kun maailma haluaa puhua, se puhuu Unicodea
UTF8 BYTE MAPPING:
        1111111111111111111111111111111111111111111111111111111
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:
        1111111111111111111111111111111111111111111111111111111
TALLY:
        NUMBER OF ONE-BYTE CHAR:      052
        NUMBER OF TWO-BYTES CHAR:     000
        NUMBER OF THREE-BYTES CHAR:   000

```


CHAPTER 7

Running COBOL in a z/OS Unicode Environment

This chapter discusses how to:

- Use the PeopleTools COBOL Unicode Conversion Utility for z/OS
- Identify converted COBOL programs.
- Understand converted data.
- Use error, exception and summary logging.
- Fine-tune converted COBOL programs.

Understanding Running COBOL in a z/OS Unicode Environment

This section discusses:

- Unicode encodings in PeopleSoft COBOL on z/OS.
- Data usage for Unicode storage.
- Input/output in the z/OS Unicode environment.

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 chapter, 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 NT servers.

On Unix and Microsoft Windows NT 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.

See Also

Chapter 4, “Selecting and Configuring Character Sets and Language Input and Output,” The Unicode Standard, page 42

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 NT, there is no need to explicitly expand storage size for Unicode processing in the z/OS Unicode environment. To store ten characters on DB2 UDB for OS/390 and 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.

See Also

Chapter 7, “Running COBOL in a z/OS Unicode Environment,” Running the COBOL Unicode Conversion Utility for z/OS, page 98

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 UDB for OS/390 and 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.

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 Crystal Reports or SQR 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 PeopleSoft 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.

See Also

Chapter 7, “Running COBOL in a z/OS Unicode Environment,” Running the COBOL Unicode Conversion Utility for z/OS, page 98

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 and discusses how to:

- Automatically run the COBOL Unicode Conversion Utility for z/OS.
- Manually run the COBOL Unicode Conversion Utility for z/OS.

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\psclubcvtz.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 <code>-psfs</code> flag to process all the PeopleSoft COBOL source subdirectories at once. Example: <code>-s:d:\PT8\SRC\CBL</code> 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 <code>-psfs</code> flag. Example: <code>-t:d:\PT8\SRC\CBLUNICODEZ</code>
-r: or -rd: <Log Directory>	-r generates only the summary log file; -rd generates all of the log files. <u>See Chapter 7, “Running COBOL in a z/OS Unicode Environment,” Using Error, Exception and Summary Logging, page 106.</u>
-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.

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 NT 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 NT.

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 *****
```

COBOL Unicode Conversion Utility for z/OS comments

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 NT 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.

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 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) .
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.

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 EBCDIC 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().

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 SIZE
    N'%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
```

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 NT 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 how to:

- Set logging parameters.
- View messages from the exception log.
- View messages from the summary log.

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.

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 from 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.
<Field 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.

<pre> Conversion Summary: Source: c:\ps\cbl\fs Target: c:\temp\fschl Number of Copy Libraries Read: 576 Modified: 513 Not Modified: 63 Number of Programs Read: 572 Modified: 564 Not Modified: 8 </pre>	

Summary Log Example

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 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:

- Identifying Unicode data for z/OS, Unicode data for Microsoft Windows NT/Unix and non-Unicode data.
- Character fields and byte size.
- Character set conversion.

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 NT/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 NT/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 NT/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 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 NT/Unix from Unicode processing on z/OS. Unicode processing on Microsoft Windows NT/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.

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).

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 may be 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: <pre>move <PIC X field or alphanumeric string literal>=> to <PIC N field></pre> Note. Using the statement <code>move <PIC N field or national literal> to <PIC X></code> causes a compiler error. When converting from Unicode data item to non-Unicode, you must insert the <code>DISPLAY-OF()</code> function to convert the Unicode data to non-Unicode first.
DISPLAY-OF()	This function converts data from Unicode to a non-Unicode character. The <code>DISPLAY-OF()</code> function has two forms: - The following converts from Unicode to system EBCDIC character set. <pre>FUNCTION DISPLAY-OF(<PIC N field or national=> literal>)</pre> - The following converts from Unicode to a character set specified in the <code>CCSID</code> parameter. <pre>FUNCTION DISPLAY-OF(<PIC N field or national=> literal>, <CCSID>)</pre>
NATIONAL-OF()	This function converts data from a non-Unicode character set to Unicode. <code>NATIONAL-OF()</code> function has two forms: - The following converts from the system EBCDIC character set to Unicode. <pre>FUNCTION NATIONAL-OF(<PIC X field or => alphanumeric literal>)</pre> - The following converts from a character set specified in the <code>CCSID</code> parameter to Unicode. <pre>FUNCTION NATIONAL-OF(<PIC X field or => alphanumeric literal>, <CCSID>)</pre>

CHAPTER 8

Sorting in PeopleTools

This chapter provides an overview of sort orders and discusses how to:

- Set the sort order.
- Force a binary sort in Structured Query Language (SQL).
- Sort in COBOL.
- Sort in Structured Query Reports (SQR).

Understanding Sort Orders

This section discusses:

- Sorting overview.
- PeopleTools sorts.
- Database-level SQL ORDER BY sorts.
- PeopleTools in-memory sorts.
- Binary sorts.

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 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 [Chapter 8, “Sorting in PeopleTools,” PeopleTools In-Memory Sorts, page 114.](#)

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, Crystal, 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 *Enterprise PeopleTools 8.49 Installation Guide for your database platform*.

See Also

Documentation 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 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 [Chapter 8, “Sorting in PeopleTools,” Setting the Sort Order, page 115](#).

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.

Setting the Sort Order

As noted previously in this chapter, 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, Utilities, Administration, 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 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 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.

See Also

Enterprise PeopleTools 8.49 PeopleBook: PeopleCode Language Reference, “Meta-SQL,” %BINARYSORT

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 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

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 from SQR. 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.

CHAPTER 9

Using Global Reporting and Data Analysis Tools

This chapter discusses:

- Language-sensitive queries.
- PeopleSoft Cube Manager for global implementations.
- The Strings table and language-sensitive text in reports.
- SQR for PeopleSoft globalization.
- PS/nVision reporting for global implementations.
- International versions of Microsoft Excel.

See Also

Enterprise PeopleTools 8.49 PeopleBook: Crystal Reports for PeopleSoft, “Working with Multiple Languages”

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 though 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.

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 Process Scheduler processes.

See Also

Enterprise PeopleTools 8.49 PeopleBook: PeopleSoft Query

Using PeopleSoft Cube Manager for Global Implementations

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

This section discusses:

- Cognos PowerPlay local-language cubes.
- Hyperion Essbase multiple-language cubes.

See Also

[Chapter 17, "Translating Application Definitions," Translating Trees, page 266](#)

[Chapter 12, "Working With Language-Sensitive Application Data," page 189](#)

Enterprise PeopleTools 8.49 PeopleBook: PeopleSoft Cube Manager

Using Cognos PowerPlay Local-Language Cubes

Cognos PowerPlay cubes are not language-sensitive. You must build a separate cube in each language that your implementation supports. Users must then access the language version of the cube that matches their language preferences.

When you build a PowerPlay cube instance, you can override the default language in which the cube is built. If you do not override the language, PeopleSoft Cube Manager builds the PowerPlay cube using the database's base language.

Each PowerPlay cube instance is built using a single language. This language can be the database's base language or any non-base language that your implementation supports. If the cube is built in the base language, then all language-sensitive elements appear in the base language. If the cube is built in a non-base language, then the cube's language-sensitive elements appear in that language using translations of language-sensitive query output and tree nodes. If the language-sensitive fields for the query output and tree nodes have not been translated into the cube's language, the cube displays the base-language descriptions.

To override the language of a Cognos PowerPlay cube:

1. Select PeopleTools, Cube Manager, Cube Instances.
Use the standard search method to access the Cube Inst Def (cube instance definition) page.
2. Confirm that the Analysis Platform field is set to *Cognos PowerPlay*.
3. Click the Platform Options link.
The PowerPlay Options page appears.
4. Select the desired language in the Language Override field.
5. Click OK to close the page.
6. Save the component.

Using Hyperion Essbase Multiple-Language Cubes

When building a Hyperion Essbase cube instance, you specify the languages that the cube supports. These can be the database base languages and any other languages supported by the implementation. When users access the cube in Essbase Administrator (or in the Essbase Excel Client), they can choose to view aliases for the cube members (stored on Essbase alias tables) in any of the supported languages. The translations of the aliases are derived from language-sensitive query output and tree nodes, so the translated aliases appear only if the tree nodes have been translated and the query data is language-sensitive and has been translated. Specific elements that have not been translated appear in the database's base language.

See the Essbase documentation for information about using Essbase alias tables.

Specify the languages that supported by an Essbase cube when you define the Essbase cube instance. If no languages are specified, no local-language Essbase alias tables are built, and the cube member descriptions appear in the database's base language.

To specify the languages of an Essbase cube instance:

1. Select PeopleTools, Cube Manager, Cube Instances.
Use the standard search method to access the Cube Inst Def (cube instance definition) page.
2. Confirm that the Analysis Platform field value is *Hyperion Essbase*.
3. Click the Platform Options link.
The Essbase Advanced Options page appears.
4. From the Language Code drop-down list box, select the cube's supported languages.
To set the value of a prompt field, click in the field and then choose from the list of supported languages. To insert a new language, click the Add button. To remove a language from the list of languages that are supported by the cube, click the Delete button.
5. Click OK to close the page.
6. Save the component.

Using the Strings Table for Language-Sensitive Text in Reports

The PeopleTools Strings table (STRINGS_TBL) stores textual strings used for language-sensitive labels and other text in PS/nVision and SQRs to avoid hard-coding labels into the report files themselves. The use of strings rather than hardcoded 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.

Each row in the Strings table keys each string to a STRING_ID, which is associated with one of two different string types:

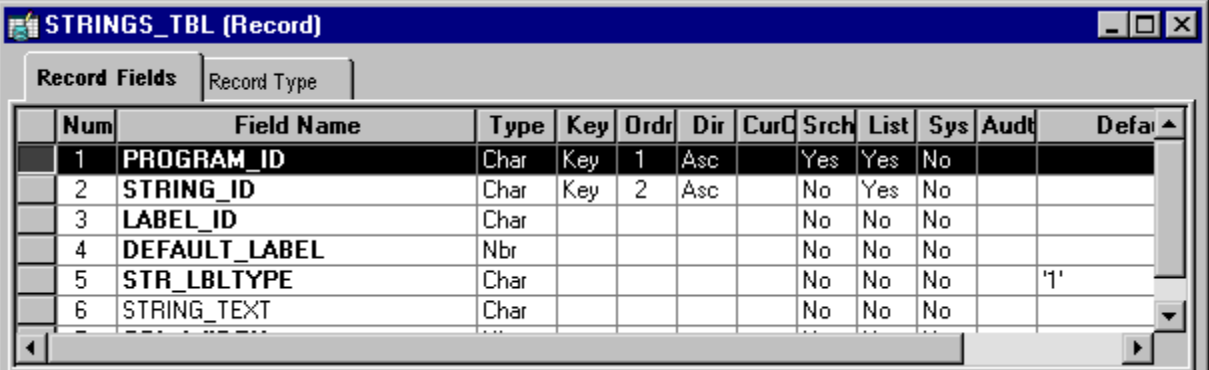
- The short (RFT Short) or long (RFT Long) field description of the field.

The STRING_ID must equal the field name.

- A free-form text string.

In either case, when a report requests a string in a particular language, the system returns the translated string if a translation is available. If no translation exists, the base language string is returned.

The Strings table is also keyed by a PROGRAM_ID field, which enables you to classify strings into groups that are used in similar reports. PROGRAM_ID can refer to a specific SQR or PS/nVision report name, or it can be a mnemonic for a group of common strings that are shared between reports.



The screenshot shows a window titled "STRINGS_TBL (Record)" with a "Record Fields" tab selected. Below the tab is a table with the following columns: Num, Field Name, Type, Key, Order, Dir, CurC, Srch, List, Sys, Audit, and Default. The table contains six rows of data:

Num	Field Name	Type	Key	Order	Dir	CurC	Srch	List	Sys	Audit	Default
1	PROGRAM_ID	Char	Key	1	Asc		Yes	Yes	No		
2	STRING_ID	Char	Key	2	Asc		No	Yes	No		
3	LABEL_ID	Char					No	No	No		
4	DEFAULT_LABEL	Nbr					No	No	No		
5	STR_LBLTYPE	Char					No	No	No		'1'
6	STRING_TEXT	Char					No	No	No		

STRINGS_TBL record definition

See Also

[Chapter 9, "Using Global Reporting and Data Analysis Tools," Using SQR for PeopleSoft Globalization, page 124](#)

[Chapter 17, "Translating Application Definitions," Translating Strings, page 270](#)

Evaluation of PS/nVision String Variables

The system uses the string definitions in the Strings table to evaluate PS/nVision variables based on three factors:

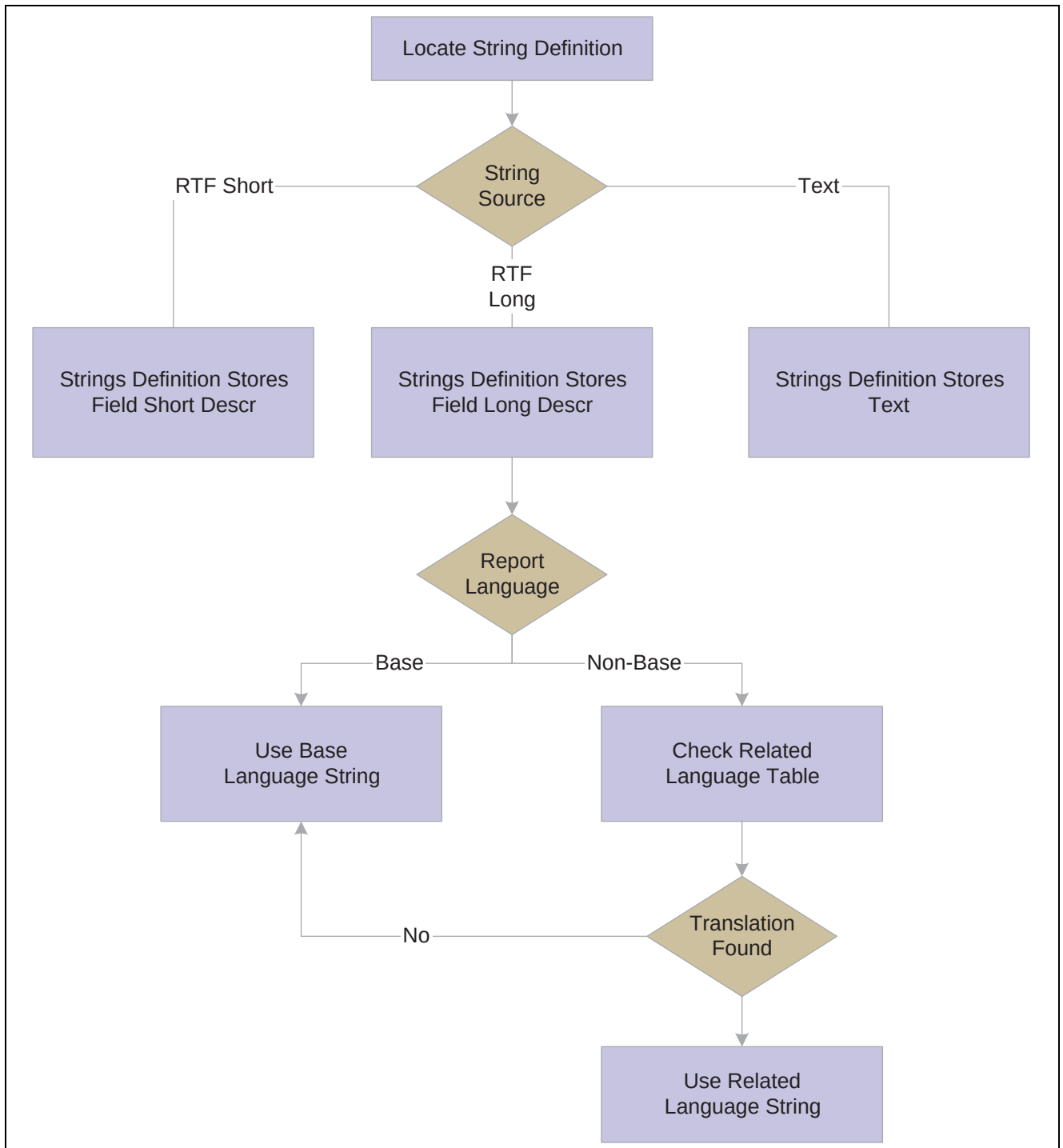
- A program ID, which identifies a set of strings used in similar reports.
- A string ID, which uniquely identifies the string definition within a program ID group.

- The language of the report request. If no language is specified, the system assumes the preferred language of the current user.

The syntax of a PS/nVision string variable includes the string ID and the program ID:

```
% .StringID, ProgramID%
```

The following flowchart shows how the system evaluates language-sensitive string variables:



Evaluation of SQR and PS/nVision string variables

The evaluation follows this sequence:

1. The system locates a string definition in the Strings table using values for the program ID and string ID.
The text string stored in the Strings table (or its related language table) can be a short or long field description (validated against the PSDBFIELD or PSDBFIELDLANG table) or a free-form text string that is not associated with a field.
2. The system determines the language of the report request. If the value is unspecified, the system uses the user's current language preference setting.
3. If the language is the system's base language, the system retrieves the string values from the base-language Strings table (STRINGS_TBL).
4. If the system is a non-base language, the system looks for a string with the correct language code in the related-language table (STRINGS_LNG_TBL).
5. If no translation exists in the related-language table, the system uses the string in the base-language table (Default).

See Also

Chapter 17, "Translating Application Definitions," Translating Strings, page 270

Using SQR for PeopleSoft Globalization

SQR for PeopleSoft provides a range of features to allow not only for multilanguage reports, 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:

- Printing for A4 paper.
- Currency precision.
- Date and time formatting.
- Report translation
- PSSQR.INI and PSSQR.UNX files.
- SQR configuration for processing international text.
- SQR configuration for printing international text.
- International text in SQR for PeopleSoft programs.
- SQR for PeopleSoft supported character set encodings.

See Also

Enterprise PeopleTools 8.49 PeopleBook: SQR Language Reference for PeopleSoft, "SQR Command Reference"

Printing for A4 Paper

PeopleTools supports printing A4, Legal, and US 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.

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 the 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

SQR Date Format	Description
{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:

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, 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
- Add_Report_Translation

Init_Report_Translation

Call the Init_Report_Translation procedure from your SQR before using any of the String table information. Typically, you should call Init_Report_Translation in the Init-Report section of your SQR. Init_Report_Translation takes two parameters:

Parameter Name	Description
\$Report_ID	\$Report_ID is normally the name of the SQR. This parameter is used as the program ID when looking up strings in the Strings table.
\$Report_Language	\$Report_Language indicates the preferred language for the strings that are being retrieved. Init_Report_Translation 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, 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. 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.

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 SQRs.

`Append_Report_Translation` takes a single argument, `$Report_ID`. It assumes the same language that was specified in `Init_Report_Translation`. It must be called after `Init_Report_Translation`.

Add_Report_Translation

The `Add_Report_Translation` procedure calls `Init_Report_Translation` or `Append_Report_Translation`, depending on whether the Strings table has been initialized. It takes the same arguments as `Init_Report_Translation`. If `Init_Report_Translation` has not yet been called during the processing of this SQR, this function 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

The following sample code demonstrates how to use the Strings table to retrieve string values in SQR, using the procedures described in the preceding sections.

```
!*****
!  SAMP001:  Report on database's base language *
!*****
#include 'setenv.sqc'      !Set environment
begin-report
  do Init-Report
  do Process-Main
  do Reset
end-report
begin-heading 6
do Get_Field_Information ('SAMP001',      'REPORT_TITLE', $REPORTTITLE,      #DWRT)
do Get_Field_Information ('SAMP001',      'EXPLAIN_TEXT', $EXPTTEXT,          #DWET)
PRINT $REPORTTITLE          (1)      CENTER
PRINT $EXPTTEXT             (+2,1)
end-heading
begin-procedure Init-Report
move 'SAMP001' to $ReportID
move 'ENG' to $Language_cd
do Init_Report_Translation ($ReportID, $Language_cd)
do Append_Report_Translation ('GEN')
end-procedure
begin-procedure Process-Main
do Get_Field_Information ('GEN',          'BASELANGUAGE', $BASELANGUAGE,      #DWBL)
do Get_Field_Information ('GEN',          'ENDOFREPORT',  $ENDOFREPORT,        #DWER)
begin-SELECT
LANGUAGE_CD &Base_Language
  let $langlabel = $BASELANGUAGE || ':'
```

```

print $langlabel (+1,1)
let #fieldpos = #DWBL + 3
print &Base_Language (0,#fieldpos)
FROM PSOPTIONS
end-SELECT
print $ENDOFREPORT (+4,1)
end-procedure
#Include 'reset.sqc'      !Reset printer procedure
#Include 'sqrtrans.sqc'   !SQR Strings Table procedures

```

See Also

[Chapter 9, “Using Global Reporting and Data Analysis Tools,” Using the Strings Table for Language-Sensitive Text in Reports, page 122](#)

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, 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. 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 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 Process Scheduler on z/OS, you must set up multiple Process Schedulers.

See Also

[Chapter 9, “Using Global Reporting and Data Analysis Tools,” SQR Configuration for Processing International Text, page 129](#)

[Chapter 9, “Using Global Reporting and Data Analysis Tools,” SQR Configuration for Printing International Text, page 131](#)

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 [Chapter 9, “Using Global Reporting and Data Analysis Tools,” The PSSQR.INI and PSSQR.UNX Files, page 129.](#)

Advanced PSSQR.INI/UNIX 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.
- 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 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 <http://www.microsoft.com/globaldev> for a list of Microsoft Windows OEM codepages. 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.

See Also

Chapter 9, “Using Global Reporting and Data Analysis Tools,” The PSSQR.INI and PSSQR.UNX Files, page 129

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.
- 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
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

Font Number	Japanese	Simplified Chinese	Traditional Chinese	Thai	Korean
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 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 *Enterprise PeopleTools 8.49 Installation Guide for Your Database Platform*.

See [Chapter 15, "Adding New Languages," page 225](#).

See *Enterprise PeopleTools 8.49 PeopleBook: 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.

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, the 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.

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

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=37058
  symbol-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

Language	init-string
Traditional Chinese	<27>E<27>(18T<27>&10O<27>&16C<27>&10E<27>&195F<27>(s16.66H<27>&k2G<27>(s1p7.25v0s0b33269T
Simplified Chinese	<27>E<27>(18C<27>&10O<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 from %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.

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
```

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	HYSMyeongJo-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 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).

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*

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 PeopleSoft Application Designer) are character-based, the **Length()** and **Substr()** functions are useful for calculating the length of the string as it is stored in the database.

For example, the following code determines whether string *&abc* will fit into a database field that is 10 characters long. If the string 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 SQRs, 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 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 SQRs 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 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.

See Also

Chapter 4, “Selecting and Configuring Character Sets and Language Input and Output,” page 39

Enterprise PeopleTools 8.49 PeopleBook: SQR Language Reference for PeopleSoft, “SQR Command Reference,” 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:

PCL	<i>PCL</i> denotes PCL printing format. Values in parentheses next to PCL, for example <i>PCL (8V)</i>, 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	<i>PS</i> denotes a PostScript printer.
CSV	CSV denotes comma-separated value format. <i>CSV (*)</i> 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	<i>Flat file</i> 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.

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

Encoding Parameter	Description	Output Support
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:

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

Encoding Parameter	Description	Output Support
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:

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	Ukrainian — 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

Encoding Parameter	Description	Output Support
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
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

Encoding Parameter	Description	Output Support
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
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

Encoding Parameter	Description	Output Support
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

Encoding Parameter	Description	Output Support
CP860	MS-DOS Portuguese	Flat file
CP20261	T.61	Flat file
hp-roman8	HP Roman8	PCL (8U), Flat file

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 (US English)

The following table lists the supported encodings for the Latin (US English) character set:

Encoding Parameter	Description	Output Support
CP20105	IA5 IRV International Alphabet No.5	Flat file
EBCDIC, CP037	IBM EBCDIC - US/Canada	PS, CSV (*), Flat file

Encoding Parameter	Description	Output Support
CP037	IBM EBCDIC - US/Canada	CSV (*), Flat file
CP437	MS-DOS US	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:

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
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 signon 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 are used as for Crystal Reports described previously. Each PS/nVision layout and drill-down directory can contain a subdirectory for each language. PS/nVision searches the 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:

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.

See Also

Enterprise PeopleTools 8.49 PeopleBook: 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:

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.
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:

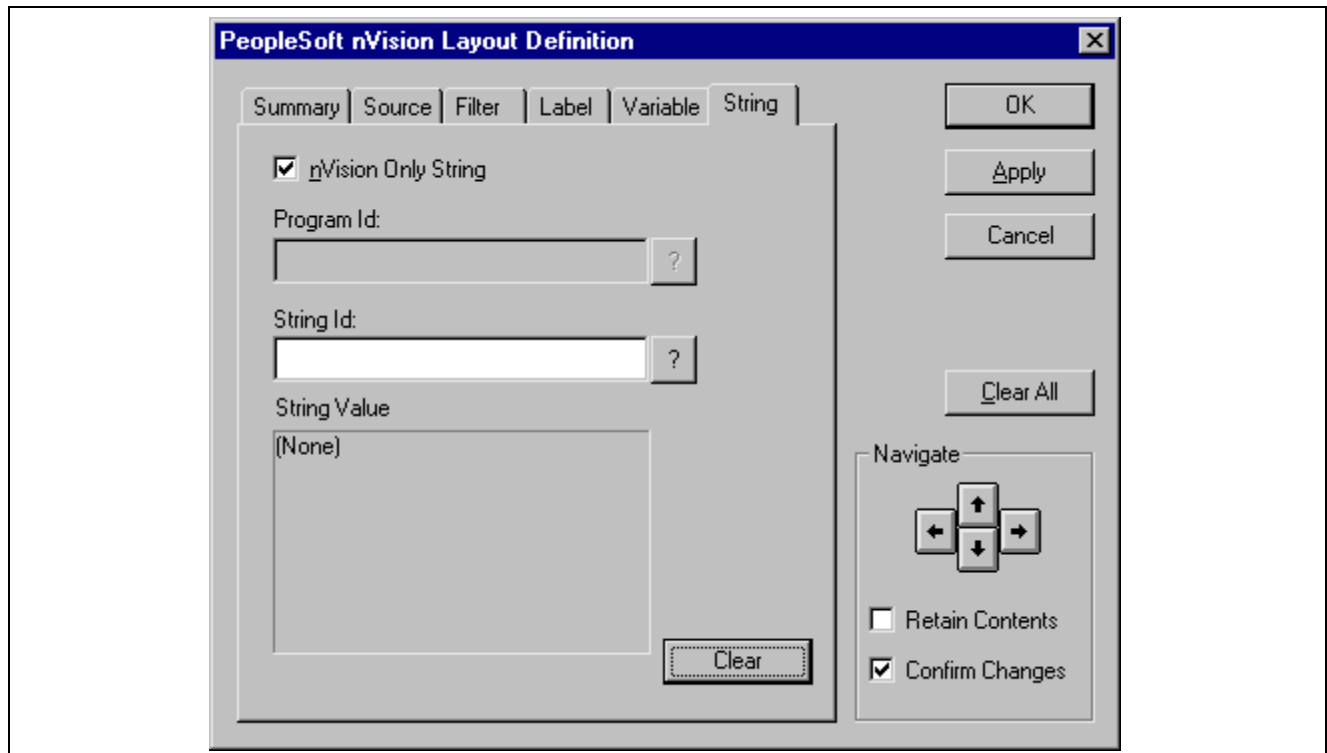
```
%.name,program%
```

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 SQRs. 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:



PeopleSoft nVision Layout Definition dialog box: String tab

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.

See Also

[Chapter 9, “Using Global Reporting and Data Analysis Tools,” Using the Strings Table for Language-Sensitive Text in Reports, page 122](#)

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 Microsoft Windows client and the Web. In the Microsoft Windows client, 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 NVUSER 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}%
```

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 PeopleSoft Application Designer, to get labels in the user's language.

PS/nVision Variables

When retrieving a descriptive value for a PS/nVision variable, such as *Business Unit Description*, PS/nVision determines whether the table being queried has a related language record and if the field being retrieved is on that record. If so, and if the user does not use the base language, PS/nVision retrieves the value from the related language record or, if the row doesn't exist on the related language record, from the base record.

See Also

Enterprise PeopleTools 8.49 PeopleBook: PS/nVision, "Running PS/nVision Reports in the Microsoft Windows Client," Creating Requests

Enterprise PeopleTools 8.49 PeopleBook: PS/nVision, "Defining Report Scopes"

International Versions of Microsoft Excel

Microsoft Excel is available in numerous languages. PeopleTools is designed to work with all international editions of Microsoft Excel as long as the base version of Microsoft Excel is supported by that version of PeopleTools. For example, if your current PeopleTools version supports Microsoft Excel 2000, then all language editions of Microsoft Excel 2000, such as Japanese Microsoft Excel 2000, are supported by the PeopleSoft system.

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.

Note. If you plan to download grids from PeopleSoft pages to Microsoft Excel 97 and you want to use the character set defined in the user language—that is, you want to use a character set other than the default UTF-8 character set—then you must select the *EXCEL97* option on the Personalizations page and set its option value to *Y*. This option is only recommended for non-English users who use Microsoft Excel 97. It is not recommended for Microsoft Excel 2000 or later versions.

See Also

Enterprise PeopleTools 8.49 Hardware and Software Requirements Guide

CHAPTER 10

Developing Global Applications

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

This chapter discusses how to:

- Develop easily translatable applications.
- Design global-ready pages.

Developing Easily Translatable Applications

When designing application objects, especially pages, follow these guidelines to ensure easy translation:

- Avoid using page-based 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.

- 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 15 percent extra space between a label and its field, and 50 percent extra space for abbreviations.

- Complete page layout early in the development cycle.
- Because the positions of fields and other objects on a page are stored individually for each language, you might need to reapply page layout changes after the page has been translated.

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.

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 the 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 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. Reduce or 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 PeopleSoft 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, on the page.

- Associate group boxes with record fields.

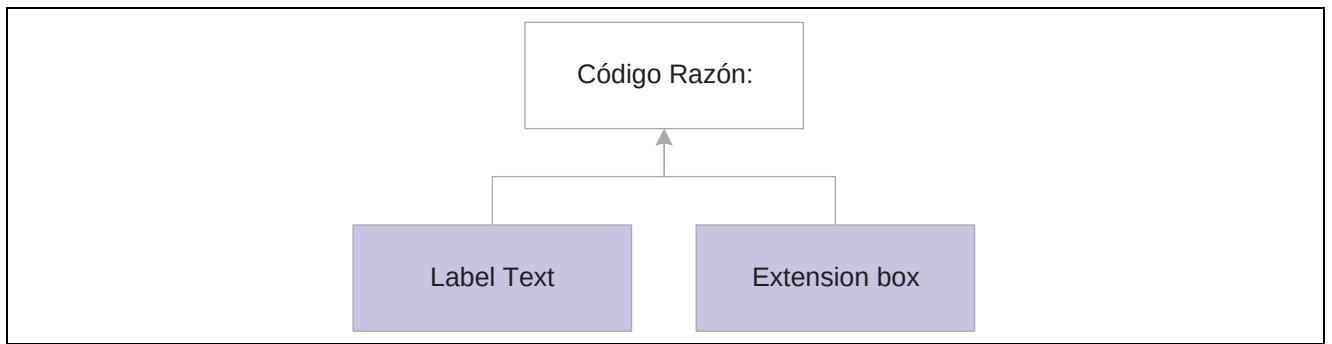
Use the Page Field Properties dialog box in PeopleSoft Application Designer to associate group boxes with record fields. In this way, the label for the group box can be derived from field's RFT Short or RFT Long value. The only effect of this association is the label derivation. It has no other effect on the page's operation.

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

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.

- 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, the dotted boundary box has two components—the surround box for the label, and an extension box to the right of the label. This extension box size is typically 10 percent of the label size and is a useful guide as to the minimum amount of space that you should leave between a field and its label to allow for expansion during translation. Ensure that there is enough space between the field and the label, so that the extension box doesn't overlap the field or other page controls. The following diagram shows the label text and extension box:



Page control label and extension box

PART 3

Working With PeopleSoft Applications in Multiple Languages

Chapter 11
Using Related Language Tables

Chapter 12
Working With Language-Sensitive Application Data

Chapter 13
Implementing Bidirectional Language Support

CHAPTER 11

Using Related Language Tables

This chapter provides an overview of Related Language tables and discusses how to:

- Install PeopleSoft-provided translations.
- Swap the base language.
- Create related language tables.
- Create related language views.

Understanding Related Language Tables

This section discusses:

- Related language tables overview.
- Related language table structure.
- How related language tables store transactions.
- How related language tables are used.

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.

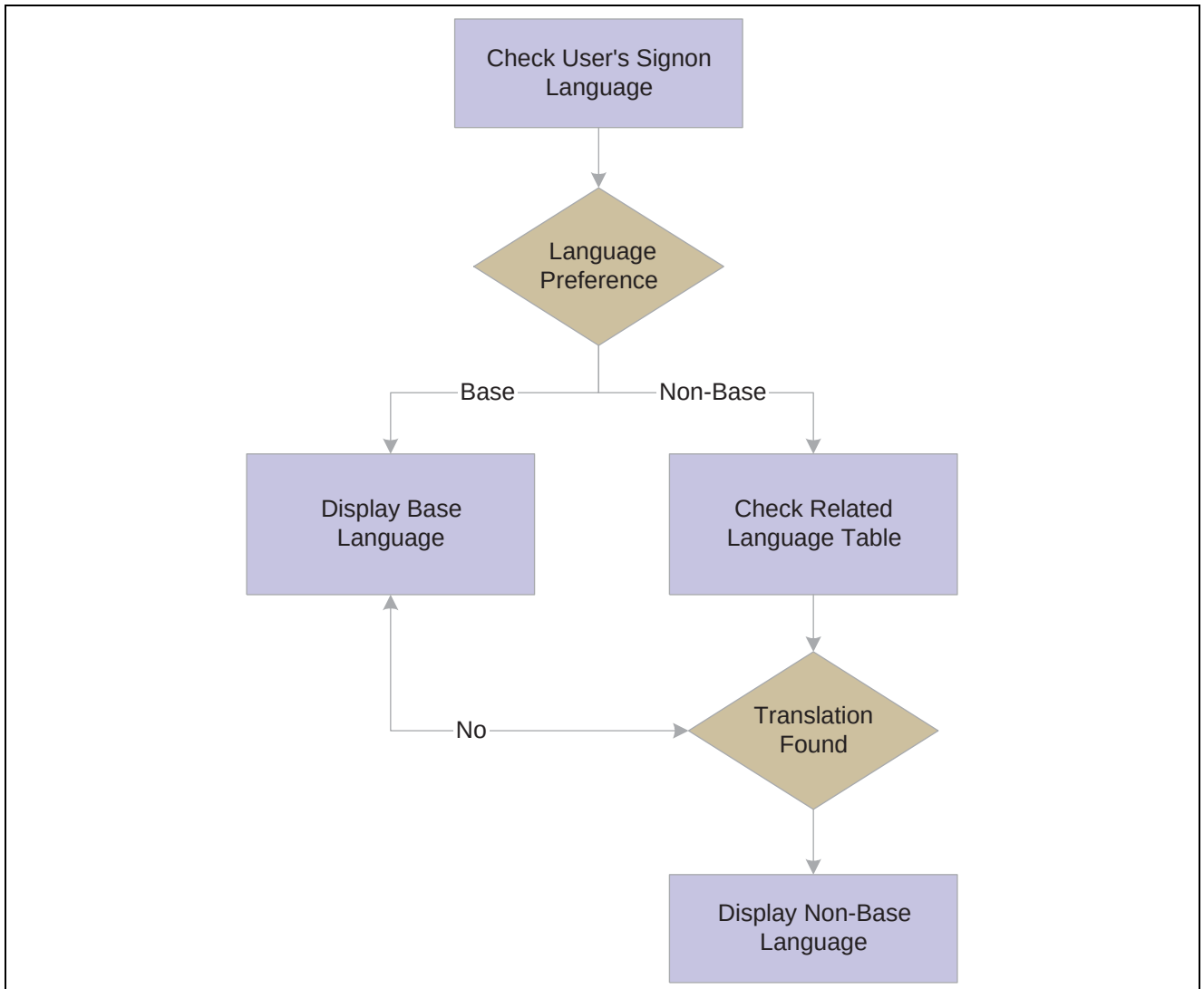
In this case, you would not need a related language table for your Journal Lines table, as you would be maintaining Journal Line descriptions in a single language only, and this would be in the base table, but you would need a related language table for each of your ChartField tables.

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* refers either to the PeopleSoft Pure Internet Architecture sign-in language or, in the case of PeopleSoft Application Designer, to the language preference as 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. The following flowchart shows this flow of execution:



System display of base and non-base 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 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 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.

See Also

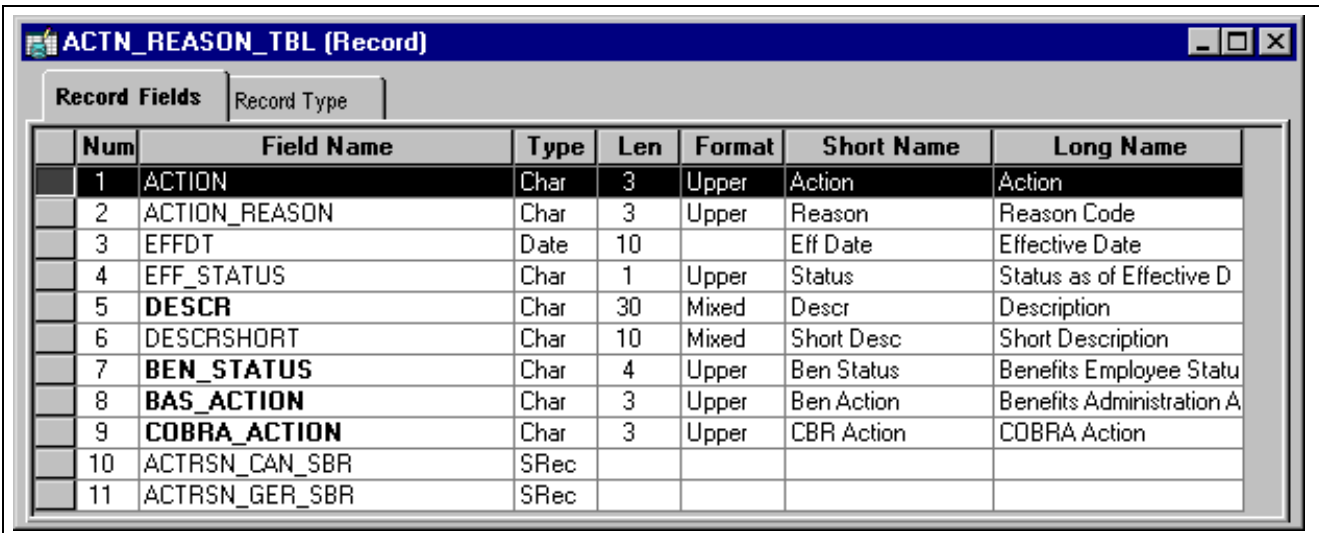
Chapter 2, “Controlling International Preferences,” page 11

Chapter 12, “Working With Language-Sensitive Application Data,” Editing Data in Multiple Languages, page 190

Related Language Table Structure

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

Each table in your database that is not a related language table is, by definition, a base language table. The Action Reason table, ACTN_REASON_TBL, is one example of a base table. Note that it has DESCR and DESCRSHORT fields which would typically be language-sensitive. All other fields in the table are translate value or prompt table codes you can edit and are therefore not language sensitive. In the following example, the developer has decided that it is likely that each user would need to see the description of the Action Reason in their own language, and so a related language table is needed to store translations of DESCR and DESCRSHORT.



The screenshot shows a window titled "ACTN_REASON_TBL (Record)". Inside, there is a table with 7 columns: Num, Field Name, Type, Len, Format, Short Name, and Long Name. The table contains 11 rows of field definitions.

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

ACT_REASON_TBL record

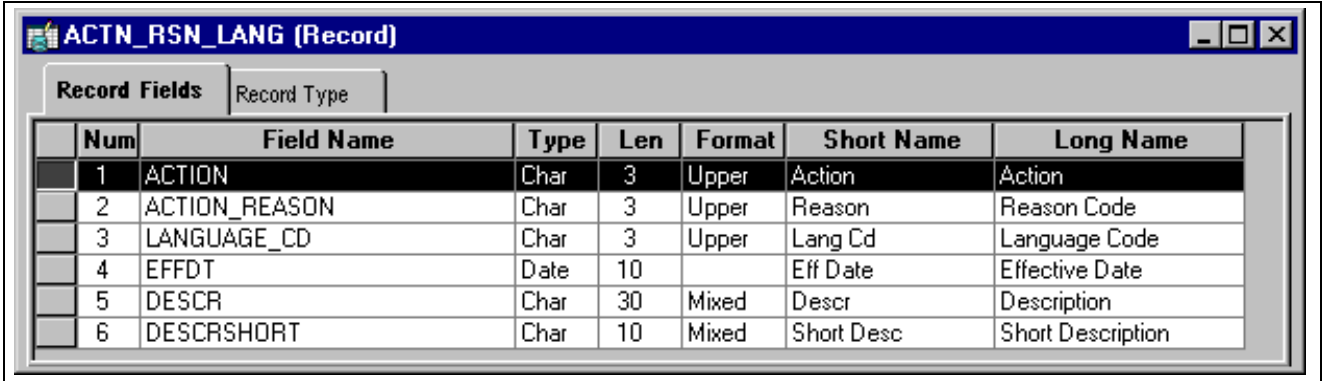
The developer has therefore created a new record definition, ACTN_RSN_LANG, to act as the related language table for ACTN_REASON_TBL. Related Language records are defined in PeopleSoft Application Designer just like any other record definitions. However, as they often contain many of the same fields 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 on 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 and DESCRSHORT, or LONGNAME and SHORTNAME.
- Not contain any fields that are not present on the base table, except the LANGUAGE_CD key field.

- Not contain any non-key, non-language-sensitive fields from the base table.

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



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

ACTN_RSN_LANG record

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.
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.

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 ACTN_RSN_TBL table.

Action	Action_Reason	Effdt	Descr	DescrLong
PAY	PRO	01/01/98	Promotion	Promotion

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

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

How Related Language Tables Are Used

Related language tables are the primary means of storing language translations in the PeopleSoft system. They store translations of descriptive text for 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 page appears when the user's language preference is Spanish (ESP). The Country field (País) prompts against the Country table (COUNTRY_TBL). The translated description (Francia) comes from the Country table's related language table (COUNTRY_TBL_LANG).

Page with translated descriptive text

Many PeopleTools object definitions, including fields, records, and components, also have associated related language tables. This makes maintenance of multiple language 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.

In the following example all the fields on the ADDRESS2_SBP subpage are translated into Spanish:

Field	Base Long Name	Tgt Long Name	Base Short Name	Tgt Short Name
Label	Base Long Name	Tgt Long Name	Base Short Name	Tgt Short Name
Page Text	Base Label Text	Tgt Label Text		
Xlat Value	Base Long Name	Tgt Long Name	Base Short Name	Tgt Short Name
Xlat Eff Date				
ADDRESS2_SBP				
ADDRESS1				
LB1 - ADDRESS1	Address Line 1	Dirección1	Address 1	Dirección1
ADDRESS2				
LB1 - ADDRESS2	Address Line 2	Dirección2	Address 2	Dirección2
ADDRESS3				
LB1 - ADDRESS3	Address Line 3	Dirección3	Address 3	Dirección3
CITY				
LB1 - CITY	City	Ciudad	City	Ciudad
COUNTRY				
LB1 - COUNTRY	País	País	País	País
VIEW_RESUME				
ZIP				
LB1 - ZIP	Postal Code	Cd.Postal	Zip	Cd.Postal

Translation of fields in ADDRESS2_SBP

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.

See Also

[Chapter 12, “Working With Language-Sensitive Application Data,” page 189](#)

[Chapter 17, “Translating Application Definitions,” page 245](#)

Installing PeopleSoft-Provided Translations

This section provides an overview of translation installation and discusses how to:

- Add translations to an existing database.
- Upgrade customizations into a multilingual database.

Understanding Translation Installation

In many cases, PeopleSoft provides translations of the user interface and key application data for many of our application products. Except for PeopleTools, these translations are shipped on a separate CD-ROM for each product line; this CD-ROM is known as the "Global Multi-Language CD-ROM." If you did not receive this CD-ROM in your kit, you may not have licensed the appropriate translations from PeopleSoft. Contact Customer Care to obtain the CD. Translations for PeopleTools itself are shipped on the PeopleTools CD; no Global Multi-Language CD-ROM is required for PeopleTools.

Typically, you would install these translations when you first install your database. However, if you did not install the PeopleSoft-provided translations during your initial install, this section describes the steps to follow to add translations that you have licensed from PeopleSoft to an already-existing database.

Note. You should install the PeopleSoft Global Multi-Language CD-ROM at the same time that you install the English release.

To add translations licensed from PeopleSoft, you must determine how you will perform your upgrade. Choose the most appropriate option:

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

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

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

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

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.

- Upgrade your customizations to a multilingual database.

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

Adding Translations to Existing Databases

This section explains how to load translations by adding them to an existing database. This method of loading translations may not be available to you, depending on the customizations you have performed on your existing database.

Prerequisites for Adding Translations to Existing Databases

Before you add translations to an existing database:

- Read the section “Understanding Translation Installation” presented previously in this chapter.

See [Chapter 11, “Using Related Language Tables,” Understanding Related Language Tables, page 167](#).

- Back up your database.

Adding Translations to an Existing Database

To add translations to an existing database:

1. Load your PeopleSoft Global Multi-Language CD-ROM.

See *Enterprise PeopleTools 8.49 Installation for your database platform*, “Setting Up the File Server.”

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

If you are running a Microsoft Windows NT or Microsoft Windows 2000 process scheduler (*PSNT*), then you must rerun the steps for “Setting Up the Batch Environment” documented in the *Enterprise PeopleTools 8.49 installation guide for your database platform*.

Because the PeopleSoft Global Multi-Language CD-ROM contains translations for your Crystal Reports and PS/nVision layouts, you must transfer those translations to your Microsoft Windows NT or Microsoft Windows 2000 process scheduler.

If you run only non-Microsoft Windows NT or Microsoft Windows 2000 process scheduler programs, you do not need to complete this step. This step also does not need to be performed for Unix batch servers because the only objects that don't contain translations in the database are Crystal and PS/nVision reports, both of which are supported on only Microsoft Windows NT and Microsoft Windows 2000 batch servers.

3. Generate database setup from PeopleSoft Data Mover.

Run PeopleSoft Data Mover in bootstrap mode by logging in with your database's access ID. From PeopleSoft Data Mover, select File, Database Setup, and follow the wizard step by step to generate the database setup script. Be sure to select "Add New Language" in the "Database Type" selection box in the second step of the Wizard. The generated database setup script appears in the PeopleSoft Data Mover input view. Be sure to enter the same database parameters that you entered the first time you ran the database setup program to install your English database; do not select any language that already exists in your database.

If you choose a new base language, then the generated script will include the command to swap the base language.

4. Run the PeopleSoft Data Mover script.

Because you have already created your database and set up the appropriate database-specific objects, you must perform only the "Run Data Mover Import Scripts" task.

5. Copy the *PPLTLSCURML* PeopleSoft Application Designer project. (No English or COMMON)

This project loads onto your system the current definitions and PeopleTools objects. When you run the project, deselect English language and COMMON objects. Choose only the language that you need.

6. Copy the *PPLDELETE* PeopleSoft Application Designer project.

This project removes definitions and objects that are not applicable to the current release.

7. Copy the *PATCH84xML* PeopleSoft Application Designer project. (No English or COMMON).

When you run the project, deselect English language and COMMON objects. Choose only the language that you need.

8. Run the *PT84xTLSxxx* data mover script in User mode.

The *PT84xTLSxxx* data mover script loads system data for the release in the target language. It also copies sample data onto your system, such as time zone sample data, currency sample data, and so on.

Note that the *xxx* in the data mover script name denotes the language code. For example, *PT84xTLSFRA.DMS* is the French version of the data mover script.

9. Run the *MSGTLxxx* data mover script in User mode.

The *MSGTLxxx* data mover script loads message catalog entries onto the system in the target language.

Note that the *xxx* in the data mover script name denotes the language code. For example, *MSGTLFRA.DMS* is the French version of the data mover script.

10. Run the VERSION Application Engine program to update your PeopleTools version numbers.

Run the PeopleTools VERSION Application Engine program against your database. This synchronizes the PeopleTools version numbers, taking the new translations into account.

11. Rerun your database audits.

Your database should now be complete. To ensure that it is complete, rerun the database audits. Refer to the section about creating and checking a database in your installation and administration documentation for more information.

See *Enterprise PeopleTools 8.49 Installation for your database platform*.

Upgrading Customizations into Multilingual Databases

This section explains how to load translations by upgrading your customizations into a multilingual database.

Prerequisites for Upgrading Customizations into Multilingual Databases

Before you upgrade customizations into multilingual databases:

- Read the section “Understanding Translation Installation” presented previously in this chapter.

See [Chapter 11, “Using Related Language Tables,” Understanding Related Language Tables, page 167](#).

- Back up your database.

Upgrading Customizations into a Multilingual Database

To upgrade your customizations into a multilingual database:

1. Follow steps one and two in the previous section entitled Adding Translations to an Existing Database.
2. Create the new multilingual database.

Create and load your new PeopleSoft Global Multi-Language database, which contains all of the languages you’ve licensed, and apply patches and bundles to bring it to the same level as your source database. Refer to the section on creating a database in your installation and administration documentation for more information.

Keep in mind that you must load the English objects provided by PeopleSoft even if you don’t plan to run your system in English. Follow all of the applicable steps in the documentation about creating a database except the Swap Base Language step. You should not swap the base language of your new database until you have copied your customizations into it.

3. Copy the customizations into the new multilingual database.

Use the PeopleSoft Application Designer Upgrade Compare process to compare your customized database against a newly installed database in order to identify your changes. Once you have identified the objects you have changed, using the Compare process, use Upgrade Copy to copy the objects into your new Multi-Language database.

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

4. Create a project with all objects.

This project will be used in the next step to upgrade customizations and translations into the test or production database.

5. Load changes in the test database.

- a. Using a test database (a copy of the production database) as a target, copy the project from the previous step to the database, selecting only the new language.
- b. Using PeopleSoft Data Mover in User mode, import the language tools DAT files using the data mover script `ppltls84Xxxx.dms`.

- c. In the PeopleSoft Pure Internet Architecture, log in as English and ensure that the new language is marked as *Enabled* in the Maintain Installed Languages pages.
- d. Run the *VERSION* Application Engine program to update your PeopleTools version numbers to synchronize the PeopleTools version numbers, taking the new translations into account.
- e. (Optional) Swap your base language. If you have not already swapped your base language and you decide to run your database in a non-English base language, swap the base language of your database.

6. After testing, repeat the previous steps to load changes in the production database

Your database should now be complete. To ensure that it is complete, rerun the database audits, as documented in the Enterprise PeopleTools Installation guide, in the chapter on creating a database.

See Also

[Chapter 11, “Using Related Language Tables,” Swapping the Base Language, page 177](#)

Enterprise PeopleTools 8.49 Installation for your database platform

Swapping the Base Language

This section provides an overview of base language swapping, lists prerequisites, and discusses how to:

- Run the swap audit report.
- Check space and resources.
- Run the SWAP_BASE_LANGUAGE Data Mover command.

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 licensed and 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.

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 code by looking in the Options page. To navigate to the Options page, select PeopleTools, Utilities, Use, Options. You can't change the Base Language Code 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.
- Sufficient space and resources to run the swapping process.

Running the Swap Audit Report

The SQR process SWPAUDIT.SQR ensures that there are no data integrity problems that will cause errors during the swap process.

This SQR does not have a run control and, therefore, cannot be run with PeopleSoft Process Scheduler. For information on how to run SQRs manually in your environment, refer to the section on creating a database in your PeopleTools installation documentation.

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

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

(SWAP-1) Related Language Records That are not Valid Records

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

To fix this problem, do one of the following:

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

(SWAP-2) The Field LANGUAGE_CD is not a Key in the Following Related Language Record(s)

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

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

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

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

To fix this problem, correct the key structure to ensure that it conforms to this requirement. Then recreate the view in the database.

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

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

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

(SWAP-5) The Following Related Language Table(s) Have the Wrong Structure Defined

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

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

(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.

To fix this problem, use your platform's query tools to select against the related language table, using the fields listed in the report, where the values do not match a row on the base table. For every row on the report there is an orphan row on the 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. Here is sample SQL code for a Microsoft SQL database.

```
SELECT * FROM PMSGCATLANG WHERE NOT EXISTS
(SELECT 'X' FROM PMSGCATDEFN A
WHERE PMSGCATLANG.MESSAGE_SET_NBR = A.MESSAGE_SET_NBR
AND PMSGCATLANG.MESSAGE_NBR = A.MESSAGE_NBR)
AND LANGUAGE_CD = 'xxx'

DELETE FROM PMSGCATLANG WHERE NOT EXISTS
(SELECT 'X' FROM PMSGCATDEFN A
WHERE PMSGCATLANG.MESSAGE_SET_NBR = A.MESSAGE_SET_NBR
AND PMSGCATLANG.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> = B.<Field Name> ...
for each field name listed)
AND LANGUAGE_CD = 'xxx'
```

Where 'xxx' is equal to the language code to which you are swapping.

(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.

To fix this problem, use PeopleSoft Application Designer to open the base record listed in the report and select a field that you want to make a key field. Right mouse click on the field and select Record Field Properties. In the dialog that pops up, select the Key check box and save the record. Then click on the Build bar item, select Current Definition, click Key, select Execute SQL Now, and click the Build button.

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.

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.

See Also

Enterprise PeopleTools 8.49 Installation for your database platform

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 exists between the base record and the related language record. In addition, both base and related language records must 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 PeopleSoft Application Designer.

Note. If a field is defined as a Long (DESCRLONG) on the related language table, 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). This is needed to support DB2 UDB for OS/390 and z/OS restrictions on bufferpool sizes.

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 removed the order key(s). 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 and DESCRSHORT.
If the base table has no fields that are language sensitive, then it does not need a related language record.
2. Design the related language record.

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.

5. Save the record.

Creating Related Language Views

This section provides an overview of related language views and discusses how to:

- Create views for one base table and one related language table.
- Create views for two base tables and one related language table.
- Create views for two base tables and two related language tables.

Understanding Related Language Views

Just as records with language-sensitive fields require related language records, views with language-sensitive fields require related language views. This is necessary because the view is created at the database-level outside of PeopleTools and cannot recognize when one or more of the tables it selects from has a related language table. If you create a view over a table that has a related language record and don't create the corresponding related language view, any data selected from your view will appear only in the base language of the database.

Related language views work the same way as related language records and must adhere to the same rules:

- You can create a related language view for any view that selects language-sensitive fields from a table with a related language record.

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 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: append a *LANG* suffix to the view name.

Note. The related language record must match the base record type, either both tables or both views

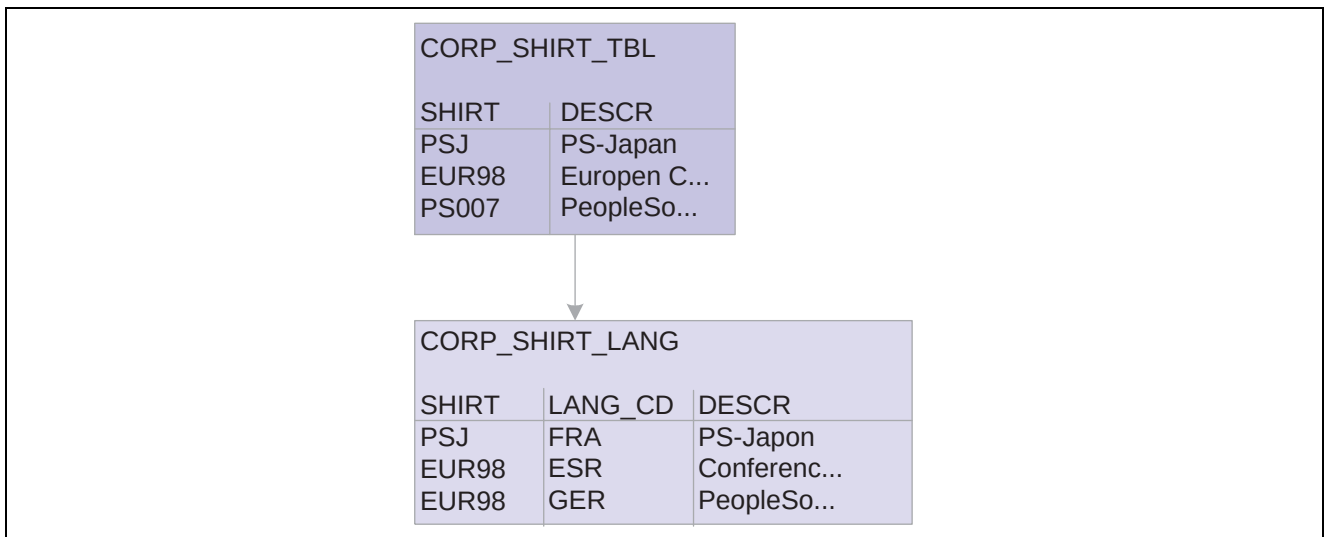
Related language views have one additional issue: the SELECT statement from the original view must be modified to select any language-sensitive fields from the appropriate related language tables.

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.

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 language-sensitive view is straightforward: select the data from the related language table, making sure to also select LANGUAGE_CD.

This example shows a view on the CORP_SHIRT_TBL, a table keyed by SHIRT and with one language-sensitive field (DESCR):



Example of one base table, one related language table

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

Following is the SELECT statement for the base view:

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

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

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

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

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

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

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.

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

DEPT_TBL			PERSONAL_DATA	
DEPT_ID	DESCR	ACCOUNT	EMPLID	NAME
09	Development	20200	20200	Jones
10	Sales	10900	10900	Smith
11	Marketing	12020	12020	Stuart

DEPT_TBL_LANG		
DEPT_ID	LANG_CD	DESCR
09	FRA	LeDvlp
10	FRA	LeSales
10	GER	Verkauf
11	GER	Markt

Example of two base tables, one related language table

The following SELECT statement is for the base view that selects the department ID and description and the name of the manager for each department:

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

The following SELECT statement is for the language-sensitive view:

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

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

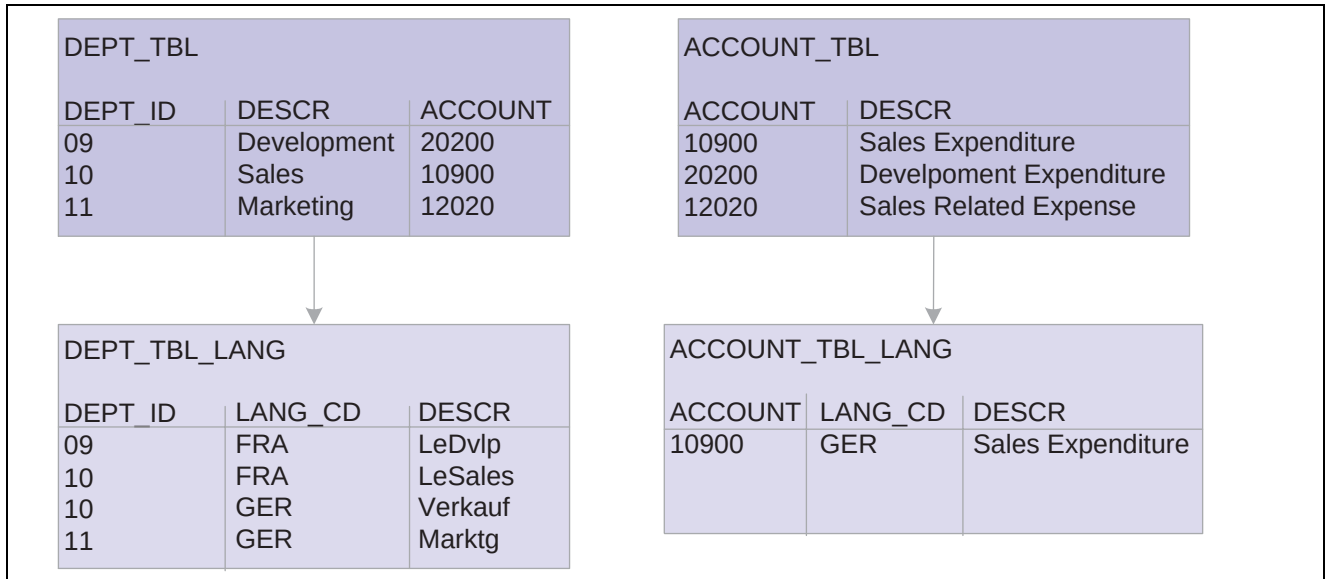
To retrieve the `MANAGER` field, `PS_DEPT_TBL_LANG` needs to be joined to `PS_DEPT_TBL`.

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

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.

This example shows a view that joins `DEPT_TBL` with `ACCOUNT_TBL`:



Example of two base tables, two related language tables

In this example, the base view joins `DEPT_TBL` with `ACCOUNT_TBL` to retrieve the description of the account. The following `SELECT` statement is for the base view:

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

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 `ACCOUNT_TBL_LANG`.

The related language view needs to take into account the following scenarios:

- A translated department may reference a translated account.
- A translated department may reference an untranslated account.
- An untranslated department may reference a translated account.

In this scenario, there is no row in `DEPT_TBL_LANG`, but there is a row in `ACCOUNT_TBL_LANG`. If you join `DEPT_TBL_LANG` with `ACCOUNT_TBL_LANG` in the view, no translation is retrieved by the related language view because the department doesn't have a translation—even though the account did.

- An untranslated department may reference an untranslated account.

The related language view doesn't need to address this situation because no translation exists in this case. It's correct for the related language view to 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 language-sensitive view:

```
SELECT A.DEPT_ID, A.DESCR, C.ACCOUNT, C.DESCR, A.LANGUAGE_CD
FROM PS_DEPT_TBL_LANG A,
     PS_DEPT_TBL B,
     PS_ACCOUNT_TBL_LANG C
WHERE A.DEPT_ID = B.DEPT_ID
     AND B.ACCOUNT = C.ACCOUNT
     AND A.LANGUAGE_CD = C.LANGUAGE_CD
UNION
SELECT E.DEPT_ID, E.DESCR, G.ACCOUNT, G.DESCR, E.LANGUAGE_CD
FROM PS_DEPT_TBL_LANG E,
     PS_DEPT_TBL F,
     PS_ACCOUNT_TBL G
WHERE E.DEPT_ID = F.DEPT_ID
     AND F.ACCOUNT = G.ACCOUNT
     AND NOT EXISTS (SELECT 'X'
                     FROM PS_ACCOUNT_TBL_LANG H
                     WHERE H.ACCOUNT = G.ACCOUNT
                     AND H.LANGUAGE_CD = E.LANGUAGE_CD)
UNION
SELECT I.DEPT_ID, I.DESCR, J.ACCOUNT, J.DESCR, J.LANGUAGE_CD
FROM PS_DEPT_TBL I,
     PS_ACCOUNT_TBL_LANG J
WHERE I.ACCOUNT = J.ACCOUNT
     AND NOT EXISTS (SELECT 'X'
                     FROM PS_DEPT_TBL_LANG K
                     WHERE K.DEPT_ID = I.DEPT_ID
                     AND K.LANGUAGE_CD = J.LANGUAGE_CD)
```

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

Statement One

The following statement addresses scenario one. It retrieves the rows where translations for both the department and the account exist.

```
SELECT A.DEPT_ID, A.DESCR, C.ACCOUNT, C.DESCR, A.LANGUAGE_CD
FROM PS_DEPT_TBL_LANG A,
     PS_DEPT_TBL B,
     PS_ACCOUNT_TBL_LANG C
WHERE A.DEPT_ID = B.DEPT_ID
     AND B.ACCOUNT = C.ACCOUNT
     AND A.LANGUAGE_CD = C.LANGUAGE_CD
```

Statement Two

The following statement addresses scenario two; it addresses instances where the department translation exists, but the account translation does not exist.

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

```
SELECT E.DEPT_ID, E.DESCR, G.ACCOUNT, G.DESCR, E.LANGUAGE_CD
FROM PS_DEPT_TBL_LANG E,
     PS_DEPT_TBL F,
     PS_ACCOUNT_TBL G
WHERE E.DEPT_ID = F.DEPT_ID
      AND F.ACCOUNT = G.ACCOUNT
      AND NOT EXISTS (SELECT 'X'
                      FROM PS_ACCOUNT_TBL_LANG H
                      WHERE H.ACCOUNT = G.ACCOUNT
                            AND H.LANGUAGE_CD = E.LANGUAGE_CD)
```

Statement Three

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

```
SELECT I.DEPT_ID, I.DESCR, J.ACCOUNT, J.DESCR, J.LANGUAGE_CD
FROM PS_DEPT_TBL I,
     PS_ACCOUNT_TBL_LANG J
WHERE I.ACCOUNT = J.ACCOUNT
      AND NOT EXISTS (SELECT 'X'
                      FROM PS_DEPT_TBL_LANG K
                      WHERE K.DEPT_ID = I.DEPT_ID
                            AND K.LANGUAGE_CD = J.LANGUAGE_CD)
```


CHAPTER 12

Working With Language-Sensitive Application Data

This chapter provides an overview of data editing in related language tables and base tables and discusses how to edit data in multiple languages.

See Also

[Chapter 17, “Translating Application Definitions,” page 245](#)

[Chapter 9, “Using Global Reporting and Data Analysis Tools,” page 119](#)

[Chapter 11, “Using Related Language Tables,” page 167](#)

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	Signon Language	Result
Translate a language-sensitive field. (The base language field has never been translated.)	Non-base	Adds a new row to the related language table that is keyed to the current 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.

Action	Signon Language	Result
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.
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.

Editing Data in Multiple Languages

This section provides an overview of multi-language entry and discusses how to:

- Enable multi-language entry.

- 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.

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 both of the following 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.

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.

Option Category: General Options

Personalizations			Find	First	1-5 of 5	Last
Personalization Option	Default Value	Override Value				
Accessibility Features	Accessibility features off					Explain
Excel 97 grid download	N					Explain
Time page held in cache	900					Explain
Multi Language Entry	No	Yes				Explain
Spell Check Dictionary	Use session language					Explain

[Restore Category Defaults](#)

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.

3. Change the value in the Multi Language Entry Override Value drop-down list box to *Yes*.
4. Click OK.

Entering Data in Multiple Languages

After you have enabled Multi Language Entry in My Personalizations, the page display changes in two important ways:

Data Language: English [New Window](#) | [Help](#)

Generic Template Definition Blackberry Email Responses

Template: SYSTEMDEFAULT

***Description:** System-wide default template

Instructional Text: Type names or email addresses in the To, CC, or BCC fields, using a semi-colon as a separator. Click LOOKUP RECIPIENT to search for a name. Click DELIVERY OPTIONS to view or change the method of the

Priority:

***Sender:** User **Email ID:**

Subject: <Enter Subject here>

Message Text: Workflow Notification
Priority: %NotificationPriority
Date Sent: %Date
Sent To: %NotificationToList

Below is the list of available variables for this template.
You can use template variables within your subject or message text.
The following variables can also be used:
%Date, %DateTime, %Time, %ServerTimeZone, %EmailAddress, %NotificationPriority,
%NotificationToList, %NotificationCCList

Template Variables	
*Value	*Description
%1	URL of the source transaction

Generic Template Definition page with Multi Language Entry enabled

1. A Data Language drop-down list box appears at the top of each page.

The Data Language drop-down list box 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 from the Data Language drop-down list box. 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 the new language selected.

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 function.

See [Chapter 12, "Working With Language-Sensitive Application Data," Enabling Multi-Language Entry, page 191](#).

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. From the Languages menu, 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, save the page.

The system updates the base and related-language tables for the page.

CHAPTER 13

Implementing Bidirectional Language Support

This chapter provides an overview of bidirectional language support and discusses how to:

- Access bidirectional PeopleSoft Pure Internet Architecture pages.
- Navigate in bidirectional PeopleSoft Pure Internet Architecture pages.
- Enable the Hijri lunar calendar display.
- Customize PeopleSoft Pure Internet Architecture for bidirectionality.

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 provided
- Platforms supported

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, DB2 z/OS and DB2 Windows/Unix/Linux.

Translations are provided for PeopleTools-owned components only. No translations of application product lines are delivered; however, you may customize applications to support bidirectionality.

Platforms Supported

Bidirectional language support is supported on the following browsers: Microsoft Internet Explorer 6.0 on Windows, Mozilla 1.6 on Windows, Netscape 7.1 on Windows, and Safari 1.2 on Apple Macintosh.

Accessing Bidirectional PeopleSoft Pure Internet Architecture Pages

As delivered, bidirectional PeopleSoft Pure Internet Architecture pages are enabled when you select Arabic as the signon language:

ORACLE®
PEOPLESOFT ENTERPRISE

حدد إحدى اللغات:

[Español](#) [English](#)
[Deutsch](#) [Dansk](#)
[Français du Canada](#) [Français](#)
[Magyar](#) [Italiano](#)
[Norsk](#) [Nederlands](#)
[Português](#) [Polski](#)
[Svenska](#) [Suomi](#)
[日本語](#) [Čeština](#)
[Русский](#) [한국어](#)
[简体中文](#) [ไทย](#)
[العربية](#) [繁體中文](#)

معرّف المستخدم: QEDMO
 كلمة المرور:

تسجيل الدخول

لتحديد إشارات تتبع، انقر هنا

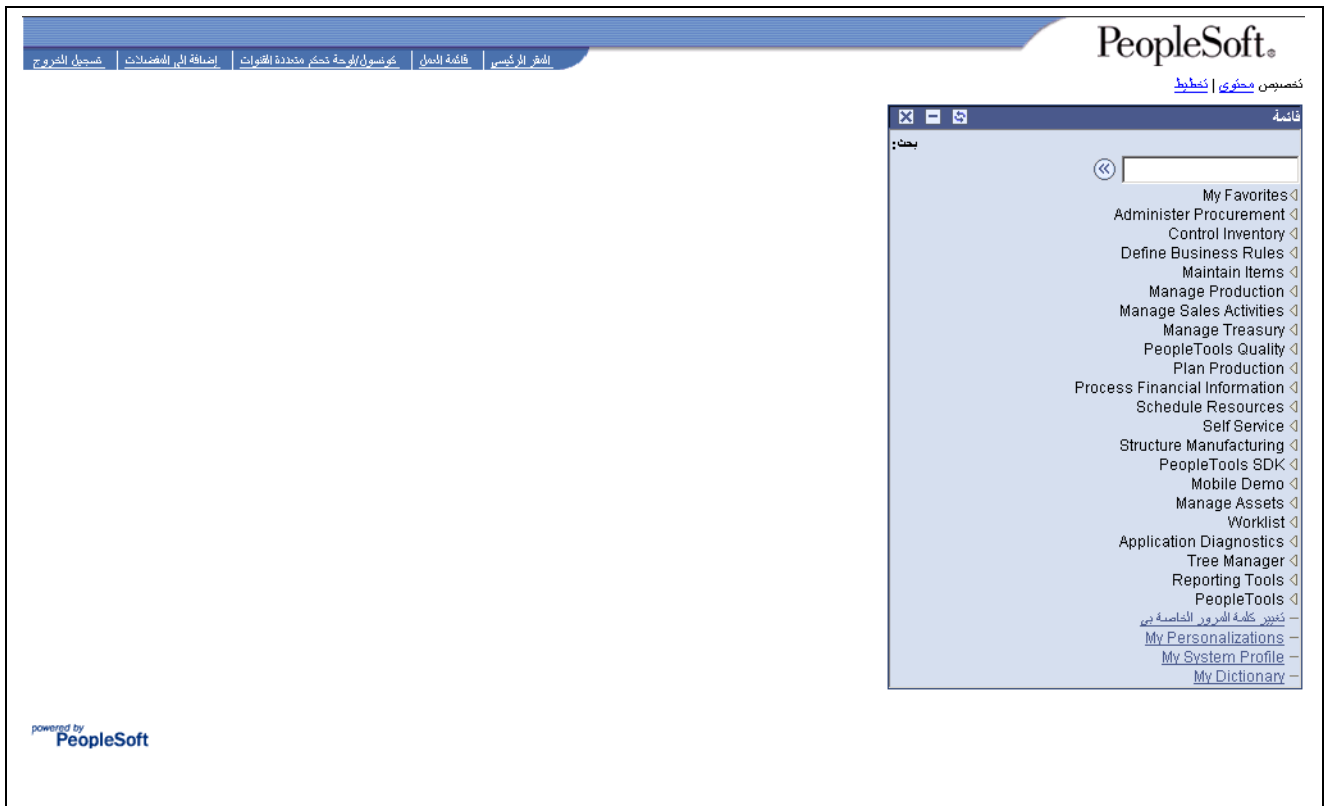
حقوق النشر © 2000، 2006، Oracle. كافة الحقوق محفوظة. إن PeopleSoft هي علامة تجارية مسجلة لشركة Oracle Corporation ولجميع الفروع التابعة لها. من الممكن أن تظهر أسماء أخرى علامات تجارية لعلامة تجارية لغيرها.

Bidirectional PeopleSoft Pure Internet Architecture signon page in Arabic

To sign in to the PeopleSoft Pure Internet Architecture page in Arabic:

1. Access the PeopleSoft Pure Internet Architecture.
2. Click *Arabic* from the list of signon languages.
A bidirectional signon page appears.
3. Enter a User ID and Password.
4. Click the Sign In button.

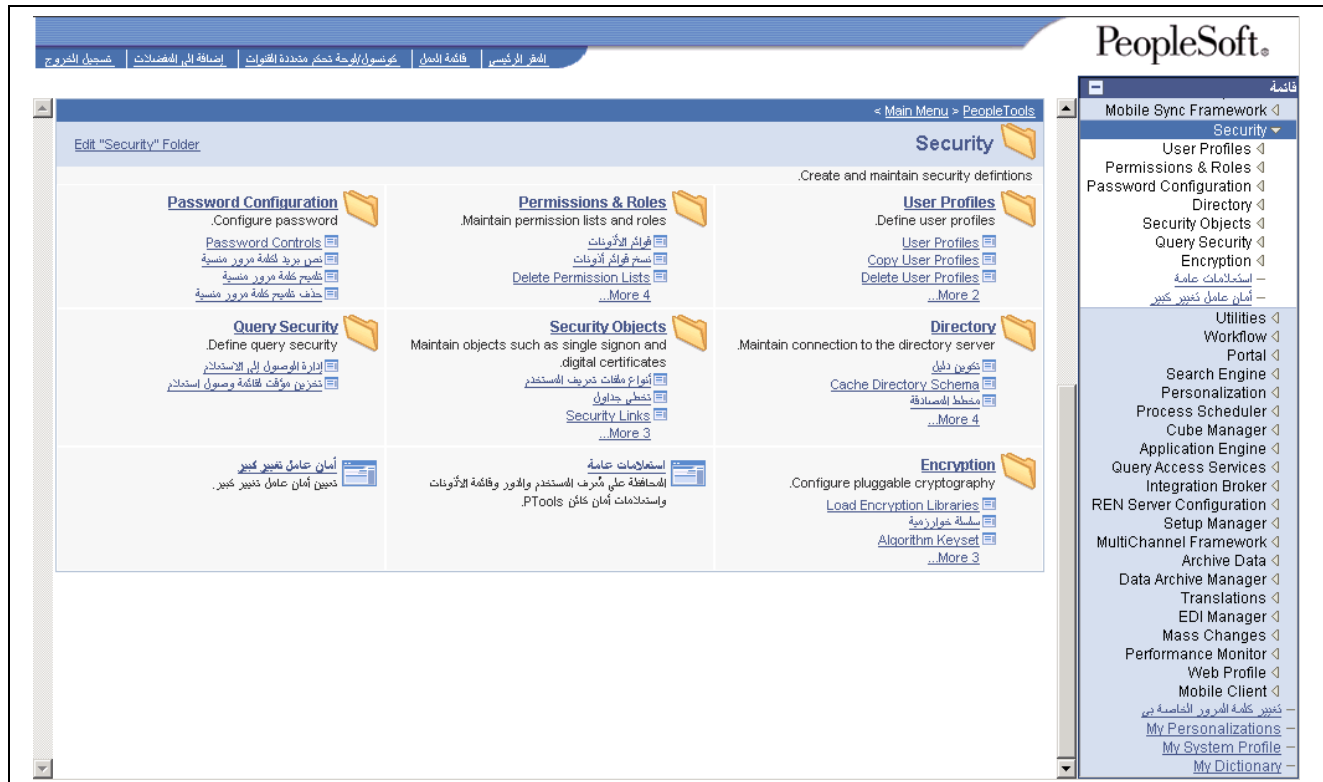
After signing in to the PeopleSoft Pure Internet Architecture, the bidirectional PeopleTools default template appears:



Bidirectional PeopleTools default template

Navigating in Bidirectional PeopleSoft Pure Internet Architecture Pages

Once signed on as an Arabic or other bidirectional user, text, fields, scroll bars, and images are right-aligned. The following graphic shows the default functional area navigation (DFAN) page for the PeopleTools Security component with bidirectionality enabled:



Bidirectional DFAN page for PeopleTools Security component

Enabling the Hijri Calendar

The Hijri (Umm al-Qura) calendar is a lunar calendar display option available as a personalizations regional setting.

When using the Hijri calendar, date field data entry and display will use Hijri dates. If auto-recognize Gregorian dates is set to *Yes* (the default) and the calendar is set to Hijri, any dates entered in date fields which fall in the range of the Gregorian calendar will be assumed to be Gregorian and will be converted to Hijri dates.

To enable the Hijri (Umm al-Qura) calendar:

1. Select PeopleTools, My Personalizations, Regions Settings, Calendar.
2. Select Hijri (Umm al Qura).

Customizing PeopleSoft Pure Internet Architecture Pages for Bidirectionality

This section discusses how to:

- Develop HTML areas.
- Work with images.

Note. You can set bidirectionality explicitly for an object or inherit it from parent objects.

Developing HTML Areas

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
<i>%AlignStart</i>	Use this system or meta-HTML variable instead of hardcoded <i>left</i> to align page elements.
<i>%AlignEnd</i>	Use this system or meta-HTML variable instead of hardcoded <i>right</i> to align text.
<i>%AlignAmount</i>	Use this system or meta-HTML variable instead of hardcoded <i>right</i> for numeric amounts
<i>%Direction</i>	Use this system or meta-HTML variable instead of hardcoded <i>ltr</i> or <i>rtl</i> 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* variables in 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 PeopleSoft 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.

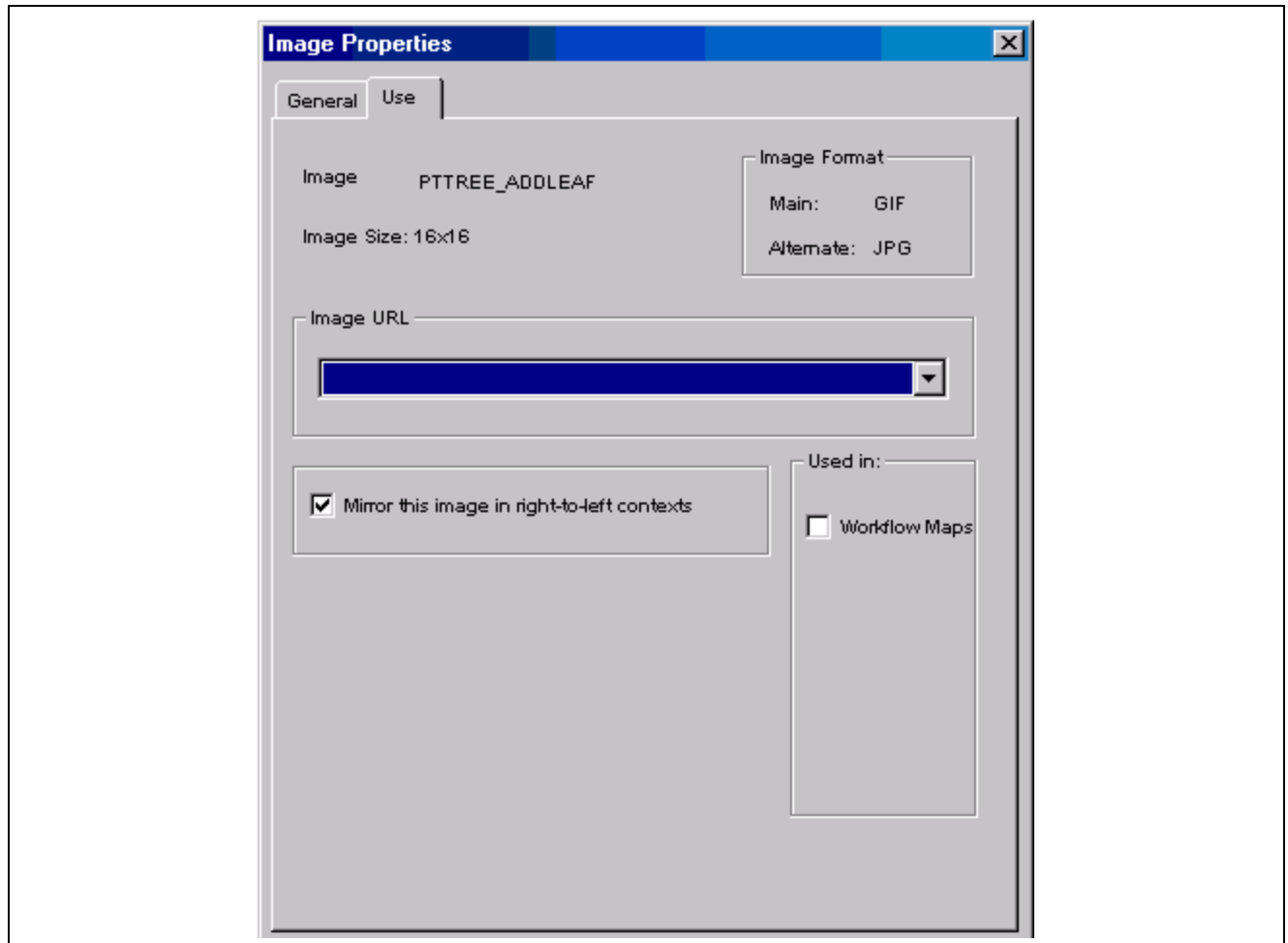


Image Properties - Use tab

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.

PART 4

Translating PeopleSoft Applications

Chapter 14
Modifying Terminology

Chapter 15
Adding New Languages

Chapter 16
Translating PeopleTools

Chapter 17
Translating Application Definitions

CHAPTER 14

Modifying Terminology

This chapter provides an overview of terminology management, lists prerequisites, and discusses how to:

- Grant terminology search privileges.
- Search terminology.
- View and replace terminology search results.
- View and undo terminology replacement results.
- Review terminology search information.
- Select terminology search objects.
- Review upgrade considerations.

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 chapter 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 [Chapter 15, “Adding New Languages,” page 225](#).

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
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 Search Criteria 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.

Step	Description
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 Replace Results/Undo page. If you don't want to replace the text, you can mark the process as complete. If Replace processing is disabled, review the results on the Search Only Results 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 Search Results/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.
Reviewing the Replace.	Go to the Replace Results/Undo page to see the list of results, along with any replacement text. This page displays all results from the original search and indicates whether you replaced text. 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, 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 Replace Results/Undo page, specify whether to undo any of the replacements.
Processing the Undo.	Using PeopleSoft Process Scheduler, run the Undo process.
Reviewing the Undo.	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.

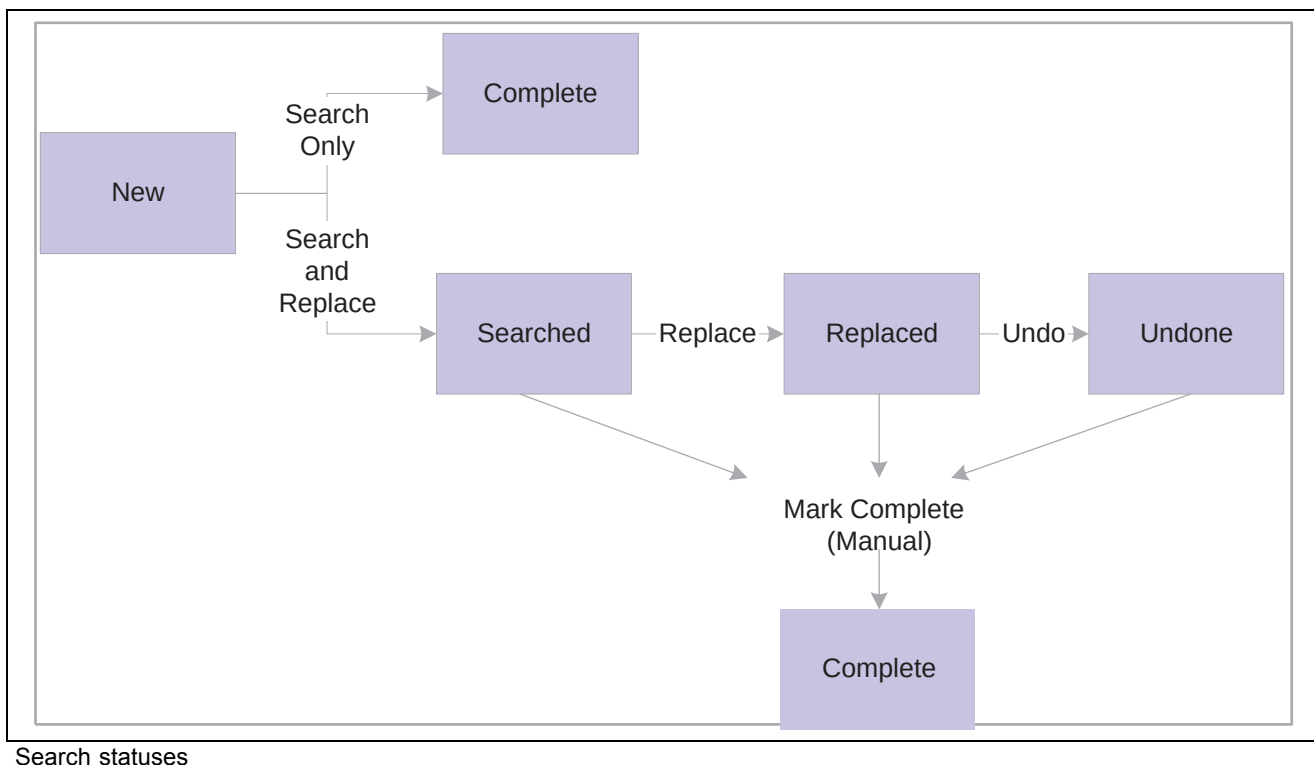
The default status for all new searches is *New*.

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*.

If you don't want to run the full Search, Replace, and Undo processes, manually mark the status *Complete* after searching or replacing. Once the status is *Complete*, you cannot change your mind. If you decide to make further terminology changes, you must recreate the search criteria using the Copy Search Criteria feature on the Search Criteria Inquiry page.

This diagram shows all the possible search statuses and their relationships:



Search statuses

Prerequisites

Before you begin modifying PeopleTools terminology, ensure that you have installed the required files and software tools.

See [Chapter 16, “Translating PeopleTools,” Prerequisites, page 237](#).

Granting Terminology Search Privileges

Use the User Search Options page to grant search and view access to your users

To set user search options:

1. Select PeopleTools, Translations, Modify Terminology, User Search Options.
2. Search for an existing user ID or add a new one.

Adding a user ID does not mean that you are creating new users; it means that you are adding a user to the list of those who have permission to use the search and replace functionality.

Add a user ID to grant access to someone who does not already have access. Search for an existing user ID to modify permissions for someone who already has access.

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

3. Set user search permissions.

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. Save the page.

Searching Terminology

This section discusses how to:

- Define search criteria.
- Clone existing search criteria.
- Run 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, Translations, Modify Terminology, Define Search Criteria.

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.

Search Profile

User ID: QEDMO

Identifier: ARTICULO

*Status: New

Base Lang: English

Search Lang: Spanish

☐ Base Language
☐ Non-Base Language
☒ Both

☒ Search Ready
☐ Search Only (No Replace)
☒ Search Long Fields

Search Objects

☒ Records, Fields, Xlats
☒ Business Processes
☒ Queries and Strings
☒ Pages

☒ Menus
☒ Exclude & from Menu Search
☒ Other System Objects
☒ Application Objects

☒ Messages
☐ Application Messages
☐ PeopleTools Messages
☒ All Messages

Search / Replacement Text

Find | View All First 1-2 of 2 Last

Base Search Text	Non-Base Search Text	Replacement Text
ABC abc Abc PRODUCT Full Word Match ☒ Match Case?:	PRODUCTO Full Word Match ☒ Match Case?:	ARTICULO Preserve Original Case
ABC abc Abc Product Full Word Match ☒ Match Case?:	Producto Full Word Match ☒ Match Case?:	Articulo Preserve Original Case

Define Search Criteria - Search Profile page

Note. The Search Profile page displays only searches where the status is *New*. This means that you can edit the search criteria only while the status is *New*. You cannot edit the search criteria after running the search.

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.

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:

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 <i>and</i> 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.

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 PeopleSoft Application Designer objects, the translatable text shows up only in PeopleSoft Application Designer dialog boxes and on property sheets. For example, you see page descriptions in the PeopleSoft Application Designer Open dialog box, but end users don't see the descriptions when they look at the actual page. The important exception to this situation are fields, translate values, page text, and menus. The descriptions for these four types of objects are visible to end users.

Select one or more of these groups:

Records, Fields, Xlats	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: <i>Administer <u>W</u>orkforce</i> is actually stored in the system as <i>Administer &Workforce</i>. 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.
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.

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 available unless you selected the Search Only (No Replace) check box.

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.

Use the three abc buttons to automatically put the text in the specified case.

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:

Exact Text Match	Returns any instance where the text constitutes the entire contents of the field. For example, if you perform an exact text search on <i>Department</i> , your results won't include fields that contain <i>Department Code</i> .
Full Word Match	Returns all occurrences of the text string where the string is not embedded in another word. In this case, searching for <i>Department</i> returns occurrences of <i>Department Code</i> , but will not return occurrences of <i>Departments</i> or <i>Departmentalize</i> .
Like Text Match	Returns all occurrences of the text string

See Also

[Chapter 9, “Using Global Reporting and Data Analysis Tools,” The PSSQR.INI and PSSQR.UNX Files, page 129](#)

[Chapter 14, “Modifying Terminology,” Reviewing Search Criteria, page 221](#)

[Chapter 14, “Modifying Terminology,” Modifying Searchable Objects, page 223](#)

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, Translations, Modify Terminology, Review 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 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:

- Searching 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 for base language text with existing translations.

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 where *Product* has been translated into Spanish. You might do this as a preliminary effort, to see the different ways in which the word *Product* has been translated.

- Searching non-base language text.

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 see the number of occurrences in the database. The result shows all occurrences where English text has a translation with the word *Producto*.

- 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*.

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, Translations, Modify Terminology, Search.
2. Search for an existing run control ID or add a new one.

Use the standard search or add method to enter your run control ID and access the Search page.

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 *Enterprise PeopleTools 8.49 PeopleBook: PeopleSoft Process Scheduler*, "Submitting and Scheduling Process Requests".

6. Click OK.

The search runs. When the Search process is complete, you can navigate to the Search Results/Replace page to see the results.

Viewing and Replacing Terminology Search Results

After a search is completed, select PeopleTools, Translations, Modify Terminology, Search Results/Replace to review the search results.

This section discusses how to:

- Replace non-message text.
- Replace message text.
- Run the Replace process.

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.

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. See Chapter 14, “Modifying Terminology,” Viewing and Replacing Terminology Search Results, page 215.
2. Look to see which pages display the search text.	
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 (hardcoded) 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.

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 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.

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.

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.

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.

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.
3. Save the page.

See Also

Chapter 14, “Modifying Terminology,” Replacing Message Text, page 217

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. Save the page.

Running the Replace Process

After you have finalized your replacement text and settings in the Search Results/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 [Chapter 14, “Modifying Terminology,” Replacing Message Text, page 217](#).

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, Translations, Modify Terminology, Replace.
2. Search for an existing run control ID or add a new one.

Use the standard search or add method to enter your run control ID and access the Replace page.

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 Search Results/Replace page and clear the Replace Ready check box. If a search is missing, go back to the Search Results/Replace page and select 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 *Enterprise PeopleTools 8.49 PeopleBook: PeopleSoft Process Scheduler*, “Submitting and Scheduling Process Requests”.

6. Click OK.

The Replace process runs. Once the replace is complete, you can navigate to the Replace Results/Undo page to view the results.

Viewing and Undoing Terminology Replacement Results

This section discusses how to:

- Undo data replacements.
- Undo text replacements.
- Run the Undo process.

Undoing Data Replacements

After you run the Replace process, select PeopleTools, Translations, Modify Terminology, Replace Results/Undo 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 Replace Results/Undo page shows the same information as the Replace Data page.

The Base Text and, if applicable, the Non-Base Text show the original text.

You can see how many pages display the text, and you can click the View Text Search Details button to see details about these pages.

The Replacement Text field shows what was—or would have been—substituted for the original search text.

The Replaced check box indicates whether the original search text was replaced. At this point, the original text has been replaced, but the change has not been finalized.

The Field Name, Length, Record and RLang Rec columns provide details about where the text is stored in the database.

2. Selectively enable and disable undo processing.

You can undo some of the replacements without undoing all of them. 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. Save the page.

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 [Chapter 14, “Modifying Terminology,” Replacing Message Text, page 217](#).

3. Save the page.

Undoing Text Replacements

The Undo Messages page displays the search results in the Message Catalog. If you edited the messages during the replace step, the text reflects those changes.

The Undo Messages page is identical to the Replace Messages page, and you can review and update messages using the same procedures.

To access the Undo Messages page, select PeopleTools, Translations, Modify Terminology, 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.

See Also

[Chapter 14, “Modifying Terminology,” Replacing Message Text, page 217](#)

Running the Undo Process

To run the Undo process:

1. Select PeopleTools, Translations, Modify Terminology, Undo.
2. Search for an existing run control ID or add a new one.

Use the standard search or add method to enter your run control ID and access the Undo page.

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 Replace Results/Undo page and clear the Undo Ready check box. If a search is missing, go back to the Replace Results/Undo page and select the Undo Ready check box.

Click the View Search Criteria button to display a read-only version of the search criteria. This is useful when you want to confirm which set of criteria you’re processing.

4. Click the Run button.

The Process Scheduler Request page appears.

5. Complete the Process Scheduler Request page.

See *Enterprise PeopleTools 8.49 PeopleBook: PeopleSoft Process Scheduler*, “Submitting and Scheduling Process Requests”.

6. Click OK.

The Undo process runs. Once the undo is complete, navigate to the All 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:

- Review search criteria.
- View results for search-only searches.
- View search results for all searches.

Reviewing Search Criteria

To see the search criteria for any search that's been defined, select PeopleTools, Translations, Modify Terminology, Review Search Criteria.

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 Search Results/Replace pages. The following table explains which component you see, based on the status of the search:

Status	Component	Navigation
New	Define Search Criteria	PeopleTools, Translations, Modify Terminology, Define Search Criteria
Searched	Search Results/Replace	PeopleTools, Translations, Modify Terminology, Search Results/Replace
Replaced/Undone	Replace Results/Undo	PeopleTools, Translations, Modify Terminology, Replace Results/Undo Replace
Complete/Undone/Replaced	All Search Results	PeopleTools, Translations, Modify Terminology, Review Search Results
Complete/Undone/Replaced	Search Only Results	PeopleTools, Translations, Modify Terminology, Review Search Only Results

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 Search Results/Replace pages. Instead, you must access the result using one of the Inquiry pages.

To see the search results for a completed Search Only (No Replace) search, select PeopleTools, Translations, Modify Terminology, Review Search Only Results.

This page is a display-only version of the Search Results/Replace page.

See Also

[Chapter 14, “Modifying Terminology,” Viewing and Replacing Terminology Search Results, page 215](#)

Viewing Search Results for All Searches

To see the search results for any search, select PeopleTools, Translations, Modify Terminology, Review Search Results.

This page is a display-only version of the Replace Results/Undo page.

See Also

[Chapter 14, “Modifying Terminology,” Viewing and Undoing Terminology Replacement Results, page 219](#)

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.

The screenshot shows a dialog box titled "Search Objects". It contains a list of object groups, each with an "i" icon (View Objects button) and a checkbox. The groups and their selection status are as follows:

Object Group	Selection Status
Records, Fields, Xlats	Selected (checked)
Business Processes	Selected (checked)
Queries and Strings	Selected (checked)
Pages	Selected (checked)
Menus	Selected (checked)
Exclude & from Menu Search	Selected (checked)
Other System Objects	Selected (checked)
Application Objects	Selected (checked)
Messages	Selected (checked)
Application Messages	Not Selected (unchecked)
PeopleTools Messages	Not Selected (unchecked)
All Messages	Not Selected (unchecked)

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:

1. Select PeopleTools, Translations, Modify Terminology, 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:

- 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.

CHAPTER 15

Adding New Languages

This chapter provides an overview of the addition of new language definitions to the PeopleSoft database and discusses how to add new language codes to the system.

Understanding the Addition of New Language Definitions

PeopleSoft provides translations in several languages for all end-user objects, and these translations can be licensed from PeopleSoft. 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.

Adding New Language Codes to the System

This section discusses how to:

- Determine PeopleSoft and ISO codes for your language.
- Determine appropriate non-Unicode character sets.
- Add new translate values to the LANGUAGE_CD field.
- Add new languages to the PSLANGUAGES table.
- Modify Configuration Manager Windows resources to include the new language in the Configuration Manager language selection drop-down list box.
- Modify the PeopleSoft Pure Internet Architecture signon page to include a selection for the new language.
- Add text. and error.properties files for languages.
- Add associated images in the new language.
- Adjust truncated toolbar buttons.

See Also

Enterprise PeopleTools 8.49 Installation for your database platform

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
CFR	Canadian French
CZE	Czech
DAN	Danish
DUT	Dutch
ENG	English
ESP	Spanish
FIN	Finnish
FRA	French
GER	German
GRK	Greek
HEB	Hebrew
HUN	Hungarian

Code	Language
JPN	Japanese
KOR	Korean
MAY	Malay
NOR	Norwegian (Bokmål)
POL	Polish
POR	Portuguese
RUS	Russian
SVE	Swedish
THA	Thai
TUR	Turkish
ZHT	Traditional Chinese
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. For the complete list, query the PSCHARSETS table in your SQL tool or in PeopleSoft Query.

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)
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)

Character Set	Description
ISO_8859-6	ISO 8859-6 (Arabic)
ISO_8859-7	ISO 8859-7 (Greek)
ISO_8859-8	ISO 8859-8 (Hebrew)
ISO_8859-9	ISO 8859-9 (Latin5)
Shift_JIS	Shift-JIS (Japanese)

Adding New Translate Values to the LANGUAGE_CD Field

Once you have determined the PeopleSoft three-letter 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.

Add Translate Table Value dialog box

To add the language code to the Translate table:

1. In PeopleSoft Application Designer, open the field named *LANGUAGE_CD*.
2. Select File, 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.

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*.

Adding New Languages to the PSLANGUAGES Table

You can update and add the languages used by the PeopleSoft system to the PSLANGUAGES table using the Manage Installed Languages page. To access this page, select PeopleTools, Utilities, International, Languages.

Note. When adding a new language, make sure there are entries for the language in langlist.cfg file so Verity can identify the text. The location of the langlist.cfg file is <PS_HOME>\common\langid\langlist.cfg.

Use the Manage Installed Languages page to specify:

- Languages in use.

The PSLANGUAGES table is used by many PeopleTools utilities to determine which languages defined in the database are actually in use, rather than being defined but unused.

- Default non-Unicode character sets.

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.

- Verity locale for search indexing.
- Spell check languages for language codes.

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.

See *Enterprise PeopleTools 8.49 PeopleBook: Using PeopleSoft Applications*.

- Character sets for encoding comma separated format (CSV) query results.

The PeopleSoft system downloads query results to a file in CSV. You can use the PSLANGUAGES table to select a charset value for a given language code to encode the results.

Manage Installed Languages

Language									
Customize Find First 1-26 of 26 Last									
'Language Code	Enabled	'ISO Locale	'Default Character Set	'Verity Locale Mapping	Spell Check Language	'Windows Character Set	'Verity Character Set		
1 ARA Arabic	☐	ar	ISO_8859-6	uni	US and UK English	CP1256	UTF8		
2 CFR Canadian French	☐	fr-ca	ISO_8859-1	frenchx	French	CP1252	CP1252		
3 CZE Czech	☐	cs	ISO_8859-2	uni	Czech	CP1250	UTF8		
4 DAN Danish	☐	da	ISO_8859-1	danishx	Danish	CP1252	CP1252		
5 DUT Dutch	☒	nl	ISO_8859-1	dutchx	Dutch	CP1252	CP1252		
6 ENG English	☒	en	ISO_8859-1	englishx	US and UK English	CP1252	CP1252		
7 ESP Spanish	☐	es	ISO_8859-1	spanishx	Spanish	CP1252	CP1252		
8 FIN Finnish	☐	fi	ISO_8859-1	finnishx	Finnish	CP1252	CP1252		
9 FRA French	☐	fr	ISO_8859-1	frenchx	French	CP1252	CP1252		
10 GER German	☐	de	ISO_8859-1	germanx	German (new)	CP1252	CP1252		
11 GRK Greek	☐	el	ISO_8859-7	uni	Greek	CP1253	UTF-8		
12 HEB Hebrew	☐	he	ISO_8859-8	uni	US and UK English	CP1255	UTF8		
13 HUN Hungarian	☐	hu	ISO_8859-2	uni	Hungarian	CP1250	UTF8		
14 ITA Italian	☐	it	ISO_8859-1	italianx	Italian	CP1252	CP1252		
15 JPN Japanese	☐	ja	Shift_JIS	japanb	US and UK English	CP932	Shift_JIS		
16 KOR Korean	☐	ko	CP949	koreab	US and UK English	CP949	CP949		
17 MAY Bahasa Malay	☐	ms	ISO_8859-1	uni	US and UK English	CP1252	UTF-8		
18 NOR Norwegian	☐	no	ISO_8859-1	bokmalx	Norwegian	CP1252	CP1252		
19 POL Polish	☐	pl	ISO_8859-2	uni	Polish	CP1250	UTF-8		
20 POR Portuguese	☐	pt	ISO_8859-1	portugx	Portuguese (Brazilian)	CP1252	CP1252		
21 RUS Russian	☐	ru	ISO_8859-5	uni	Russian	CP1250	UTF8		
22 SVE Swedish	☐	sv	ISO_8859-1	swedishx	Swedish	CP1252	CP1252		
23 THA Thai	☐	th	ISO_8859-11	uni	US and UK English	CP874	UTF8		
24 TUR Turkish	☐	tr	ISO_8859-9	uni	US and UK English	CP1254	UTF8		
25 ZHS Simplified Chinese	☐	zh-cn	GB2312	simpcb	US and UK English	CP936	GB2312		
26 ZHT Traditional Chinese	☐	zh-tw	Big5	tradcb	US and UK English	CP950	Big5		

Managed Installed Languages page

Language Code

Enter the three-letter language code from the PSXLATITEM table.

Enabled

Select the check box to indicate the language that is in use. When this check box is selected, login with the language is allowed. Normally, this check box is selected automatically during the process of loading translations into the database.

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.

Verity Locale Mapping

Use the drop-down list box to select which Verity locale should be used when indexing data for search in this language. If the language is not in the drop-down list for Verity Locale Mapping, select *englishx*. This mapping

	determines the locale to use for building search collections and searching for data.
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 codepage to use to encode CSV formatted query results for the corresponding language code. The charset must be valid for Java encoding and mime-type, or downloads will fail with an error. The default value for all language codes is <i>CP1252</i> .
Verity Character Set	Select the character set for the Verity search engine to use for its internal encoding in the given language. Normally you should not modify the value in this field.

Note. You can override the information specified on the Manage Installed Languages page using the Look and Feel tab on the Web Configuration page.

See Also

Enterprise PeopleTools 8.49 PeopleBook: Internet Technology

Modifying Configuration Manager Windows Resources

PeopleSoft Configuration Manager provides a list of languages that are available to users for their PeopleTools Windows client and Application Designer 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.

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 PeopleSoft three-letter 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 Signon Page

The PeopleSoft Pure Internet Architecture enables users to select their sign-in language by selecting the language from the Signon 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. Note the Japanese entry demonstrating how to reference an image that contains the language name for a non-Latin language:

```
<tr>
  <td>&nbsp;&nbsp;&nbsp;</td>
  <td><a HREF="%=servletLoc%>?cmd=login&languageCd=JPN"></a>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</td>
  <td class="PSHYPERLINK"><a HREF="%=servletLoc%>?cmd=login&languageCd=POR">
    Portugu&ecirc;s</a>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</td>
  <td class="PSHYPERLINK"><a HREF="%=servletLoc%>?cmd=login&languageCd=DUT">
    Nederlands</a>&nbsp;&nbsp;&nbsp;</td><td class="PSHYPERLINK"><a HREF=
    "%=servletLoc%>?cmd=login&languageCd=QUE">Quechua</a>&nbsp;&nbsp;&nbsp;</td>
</tr>
```

Adding Text and Error Properties Files for Languages

Copy the text.properties and errors.properties to two files named after your three-letter language code in this format: filename_XXX.properties, where XXX is the three-letter language code. For example, for Quechua, the filenames would be text_QUE.properties and errors_QUE.properties.

Note. The text in these files can be translated into your target language.

Note. After you have added the text and error properties files, you must restart the web server.

Text Property File Variables

You must set the following properties in the text.properties file:

Note. Do not translate any of the values you enter for these properties.

Variable	Description
<i>active</i>	Set the ISO language code. Preface the <i>active</i> variable with the two character ISO code for the language and set the variable equal to -s. For example: qu_active=-s
<i>direction</i>	Set the direction that text is read. The valid options are: • <i>ltr</i>. Left to right. • <i>rtl</i>. Right to left.
<i>alignstart</i>	Set the starting point. The valid options are: • <i>right</i> • <i>left</i> The value you select must be consistent with the directionality that you specify. For example, if you set the <i>direction</i> variable to <i>ltr</i>, then you must set the <i>alignstart</i> variable <i>left</i>. If you set the <i>direction</i> variable to <i>rtl</i>, then you must set the <i>alignstart</i> variable to <i>right</i>.
<i>alignend</i>	Set the ending point. The valid options are: • <i>right</i> • <i>left</i> The value you select must be consistent with the directionality that you specify. For example, if you set the <i>direction</i> variable to <i>ltr</i>, then you must set the <i>alignend</i> variable <i>right</i>. If you set the <i>direction</i> variable to <i>rtl</i>, then you must set the <i>alignend</i> variable to <i>left</i>.

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.

native2ascii

The `native2ascii` utility, a part of the Sun Microsystems 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> <destination file>
```

Note. Use the destination properties files that have the Java-style encoding.

Note. For more information about the `native2ascii` utility, refer to the Sun Microsystems online documentation.

Adding Associated Images in the New Language

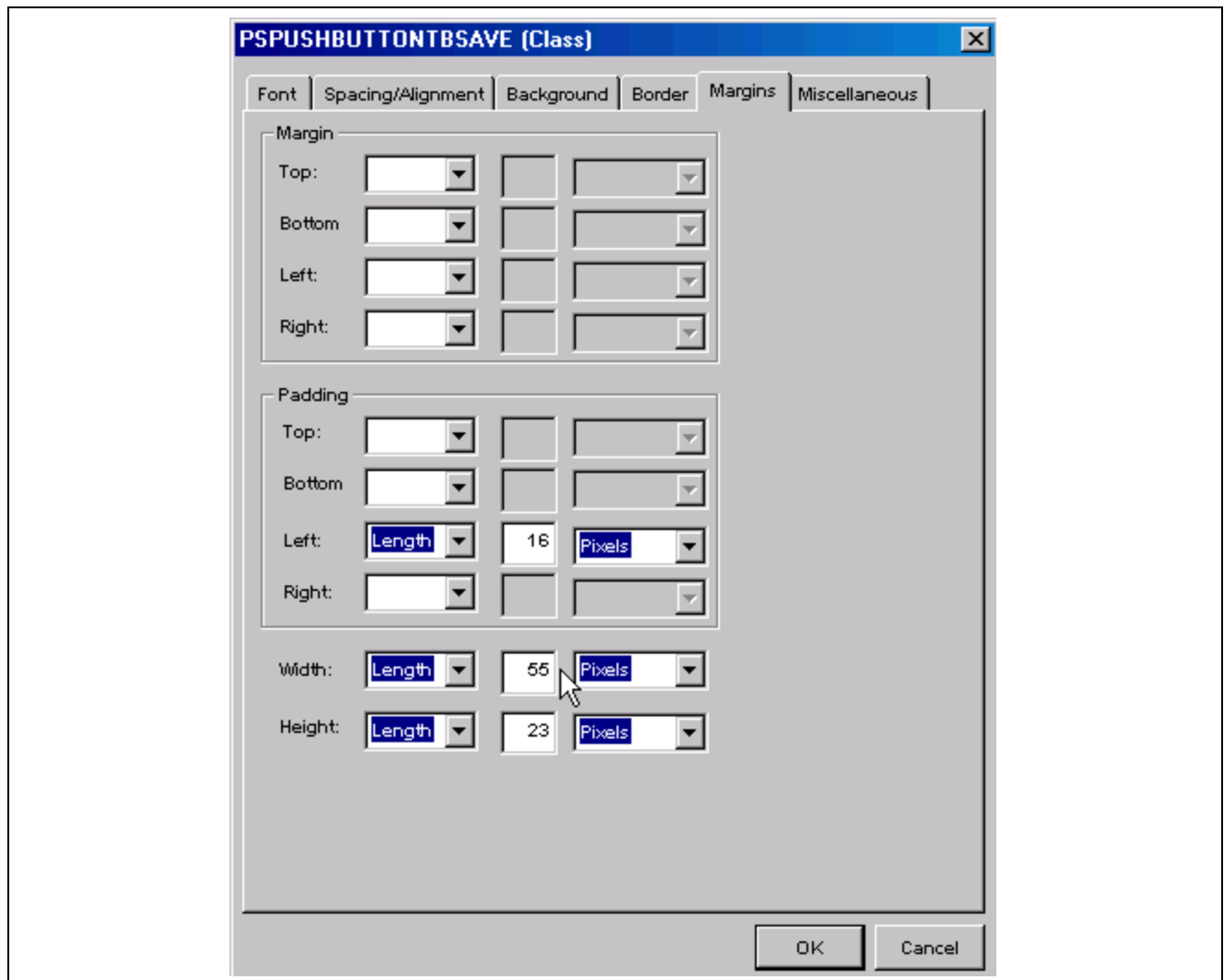
To add the new images in the new language:

1. Create a new directory named after your three-letter language code in the `PORTAL\ps\images` directory.
2. Copy the following three files from the images directory into the `\images\<language code>` directory:
 - `customerEmployeeSupplier.gif`
 - `PT_PROCESSING.gif`
 - `tagLine.gif`
3. Change the filename to `_<language code>` before the `.gif` extension.

For example, your `\images\QUE` directory would contain these files: `customerEmployeeSupplier_QUE.gif`, `PT_PROCESSING_QUE.gif`, and `tagLine_QUE.gif`.

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.



Margins tab for the PSPUSHBUTTONTBSAVE styleclass

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.

CHAPTER 16

Translating PeopleTools

This chapter provides an overview of PeopleTools translation, lists prerequisites, and discusses how to:

- Use alternate language dynamic-link libraries (DLLs).
- Locate resource directories.
- Translate resource files.
- Translate objects.

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.

This chapter explains the different translatable PeopleTools components, and how you can customize the translations delivered by the PeopleSoft system or translate PeopleTools into new languages.

See Also

[Chapter 9, “Using Global Reporting and Data Analysis Tools,” page 119](#)

Prerequisites

Before you begin modifying or creating new translations of PeopleTools, ensure that you have the required files and software tools.

This section discusses:

- Required files.
- Required tools.

To modify translations of PeopleTools or to translate PeopleTools into a new language, you must have installed the PeopleTools Language Development Kit. When you install PeopleTools, you have the option to install the PeopleTools Language Development Kit when you install the file server. If you have installed this kit, you will find a directory <PS_HOME>\SRC\RES on your file server. You can install the PeopleTools Language Development Kit at any time – if this directory does not exist on the file server where you are planning to perform the translation, rerun the PeopleTools setup program and select only this option.

The PeopleSoft system provides source code to the English language resource files and selected local language resources. It also supports header files that you need in order to produce alternate language DLLs. The PeopleSoft system also provides batch files to automate the resource compilation process.

In addition to the source code to the resource files that are delivered in the PeopleTools Language Development Kit, the PeopleSoft system delivers precompiled alternate language DLLs for each language provided by the PeopleSoft system. These are installed by selecting the PeopleTools Language Pack from the PeopleTools installer. If you intend on using a translation that PeopleSoft provides but don't plan on modifying PeopleTools translations, you only need to install the language pack, not the Language Development Kit.

The tools that you need to customize Microsoft Windows resources are part of Microsoft's Visual C++ compiler package. You won't have to change or compile any C++ code, but you must use the resource compiler and linker. If you are translating any resources other than strings, you must use the Microsoft Developer Studio.

See Also

Enterprise PeopleTools 8.49 Hardware and Software Requirements

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 DLL or EXE file (module). At runtime, the system uses the resources in the alternate language DLLs that correspond to the user's language preference.

The PeopleSoft system uses a naming convention that includes the name of the base executable (the executable with which the alternate language DLL is associated) and the language code that identifies the language of the alternate language DLL. An alternate language DLL is named as follows:

`PSZZZZXX.DLL`

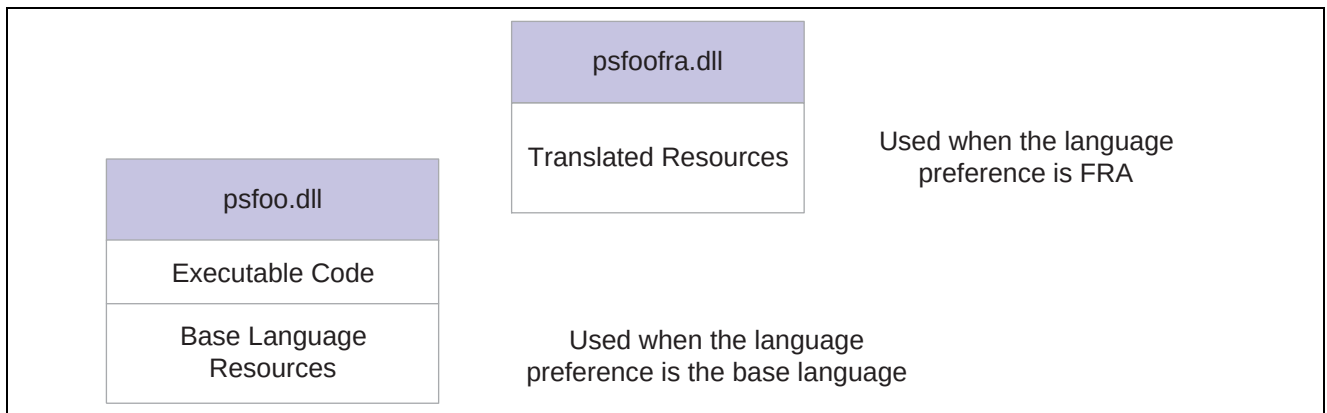
ZZZ

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.

The alternate language DLL is used only to store translated resources. All code is executed from the base-language executable. The following diagram illustrates the structure of a base-language module (PSFOO.DLL) and its French alternate language DLL (PSFOOFRA.DLL):



Base language DLL and alternate language DLL

The system uses the resources that are stored in the alternate language DLL that matches the user's language preference, if the DLL exists. Otherwise, it uses the resources that are stored in the base executable file.

Locating Resource Directories

Alternate language DLLs must be present on all application servers. Microsoft Windows clients in a three-tier installation also must have the alternate language DLLs present.

Microsoft Windows NT application servers and Microsoft Windows client machines 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.

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.

Directories and Files	Purpose
\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.

See Also

Chapter 16, “Translating PeopleTools,” Compile Translated Resource Files Using MAKEUNIX.BAT, page 243

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, including French, Spanish, Dutch, Portuguese, Japanese, German, Italian, Danish, Polish, Swedish, Traditional Chinese, Simplified Chinese, Korean, and Canadian French. 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.

This section discusses how to:

- Translate resource files.
- Compile translated resource files using MAKEALTL.BAT (Microsoft Windows).
- Compile translated resource files using MAKEUNIX.BAT (Unix/Linux).

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 [Chapter 15, “Adding New Languages,” page 225](#).

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 NT application servers only) Copy the resultant alternate language DLLs to the BIN\SERVER\WINX86 directory of your Microsoft Windows NT 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.

See Also

[Chapter 16, “Translating PeopleTools,” Compiling Translated Resource Files Using MAKEALTL.BAT, page 241](#)

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:\PT848\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:\PT848\BIN\CLIENT\WINX86

For customer builds the %PS_HOME%\res\stinggray directory should exist and the PS_HOME variable must be set. For internal builds the ..\src\inc\nonps\stinggray directory should exist, and the variable PSVER must be set.

If neither of them exist you can set CUSTOMER_BUILD to *NO*. For example in Microsoft windows:

```
set CUSTOMER_BUILD=NO
```

In addition, 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.

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 \PT848\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 property and text files.
- Translating images.

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 shows the files that contain translatable text:

Base Properties File	Translated Properties File
text.properties	text_<language_code>.properties
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 [Chapter 15, “Adding New Languages,” page 225](#).

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 Sun 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 PeopleSoft Application Designer.
3. Open the image.

Choose File, 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.

CHAPTER 17

Translating Application Definitions

This chapter provides an overview of PeopleSoft application definition translation processes and discusses how to:

- Use the Translation Designer feature.
- Use the translation pages.

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 PeopleSoft Application Designer, PeopleSoft Tree Manager, or the translations utilities. The translations utilities are a particularly powerful option for translating fields, translate values, and hardcoded 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 the Translation Designer

The Translation Designer within PeopleSoft 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.

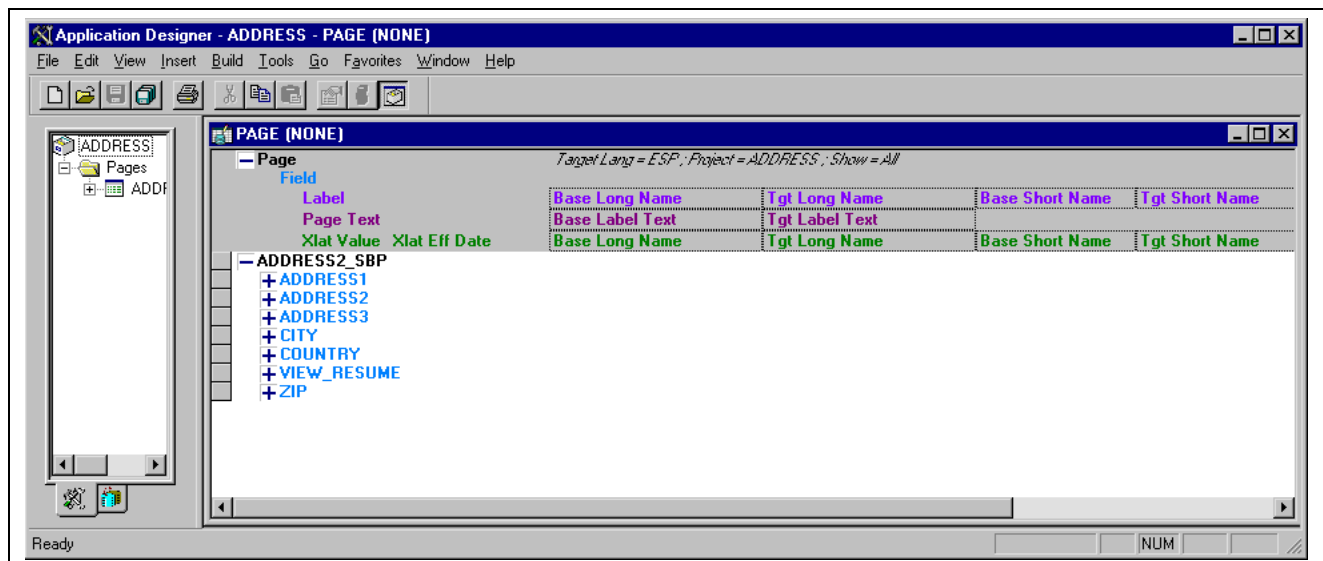
This section discusses how to:

- Open Translation Designer.
- Use Translation Designer display options.
- Work in Translation Designer.
- Translate system definitions.

Opening Translation Designer

Translation Designer shows all the translatable definitions in your current PeopleSoft Application Designer project, along with their base-language and translated text.

The grid contents are based on the PeopleSoft Application Designer project that is open when Translation Designer is started. This association with PeopleSoft 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.



Translating Page Definitions in Translation Designer

To open Translation Designer:

1. Set your language preference to the target language.

Because you use the PeopleSoft Application Designer to access Translation Designer, set your language preference using PeopleSoft Configuration Manager.

2. Open PeopleSoft 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.

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 PeopleSoft Application Designer window and then maximize Translation Designer within the window. For even more space, hide the other frames in the PeopleSoft 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.

See Also

Chapter 2, “Controlling International Preferences,” Setting Up Language Preferences for Microsoft Windows-Based PeopleTools, page 17

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.

- 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.

Field	Label	Base Long Name	Tgt Long Name	Base Short Name	Tgt Short Name
Page	Page Text	Base Label Text	Tgt Label Text	Base Short Name	Tgt Short Name
Xlat Value	Xlat Eff Date	Base Long Name	Tgt Long Name	Base Short Name	Tgt Short Name
ADDRESS2_SBP	Page - Group Box	Address	Address		
ADDRESS1	Lbl - ADDRESS1	Address Line 1	Dirección1	Address 1	Dirección1
ADDRESS2	Lbl - ADDRESS2	Address Line 2	Dirección2	Address 2	Dirección2
ADDRESS3	Lbl - ADDRESS3				
ADDRESS_TYPE	Lbl - ADD_TYPE	Address Type	Address Type	Type	Type
Xlt - H	2000-01-01	Home	Home	Home	Home
Xlt - M	2000-01-01	Post Office Box	Post Office Box	PO Box	PO Box
Xlt - O	2000-01-01	Other	Other	Other	Other
Xlt - W	2000-01-01	Work	Work	Work	Work
CITY	Lbl - CITY	City	Cuidad	City	Cuidad
COUNTRY	Lbl - COUNTRY	Country	País	Countryt	País
ZIP	Lbl - ZIP	Postal Code	Cd Postal	Zip	Cd Postal
	Lbl - ZIP2	ZIP Code		Zip	

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.

- +** 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:

- 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 PeopleSoft Application Designer functionality.
- Integrate with other applications.

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 PeopleSoft 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.

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 PeopleSoft Application Designer Functionality

Because Translation Designer is integrated into the PeopleSoft Application Designer, you can use all standard PeopleSoft 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 PeopleSoft 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 PeopleSoft 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.

See *Enterprise PeopleTools 8.49 PeopleBook: PeopleTools Security*

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.

Using the Translation Pages

This section provides an overview of the translation pages and discusses how to:

- Translate page definitions.
- Translate messages.
- Translate menus.
- Translate business processes.
- Translate PeopleSoft Process Scheduler.
- Translate mobile data.
- Translate portals.
- Translate trees.
- Translate records.
- Translate queries.
- Translate strings.
- Translate PeopleSoft Application Engine descriptions.
- Translate review page text.
- Translate application data.
- Translate HTML definitions.
- Translate Integration Broker definitions.
- Translate XML Publisher definitions.
- Translate ACE analytic model ID definitions.
- Translate time zones.

Understanding the Translation Pages

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 chapter.

To locate the translation pages, select PeopleTools, Translations, Translate System Definitions. The translation pages will help you to translate system definitions, including records, menus, menu items, components, component interfaces, queries, message channels, messages, message definitions, portal definitions, process definitions, PeopleSoft Application Engine programs, Business Interlinks, review page text, and HTML definitions.

You should translate PeopleTools definitions in the following order to ensure the least amount of duplication during your translation process:

1. Translate pages, messages, and menus—in any order (using Translation Designer).
2. Translate business processes, Process Scheduler, mobile pages, and portals—in any order.
3. Translate trees, records, queries, strings, application data, and PeopleSoft Application Engine—in any order.

Translating Page Definitions

All page objects except the HTML definitions can be translated directly in Translation Designer. However, you can also translate pages, translate values, and fields directly in PeopleSoft Application Designer if you are familiar with the designer tools.

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
Page HTML	PSPNLHTMLAREA	PSPNLHTMLLANG	Application Designer

Note. Don't be confused by the term *translate value* or *translate table*. The translate table is a common table that is used throughout PeopleTools and PeopleSoft applications and is not used for translation purposes.

Translating Translate Values

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

1. Change the PeopleSoft Configuration Manager language preference to the target language.
2. Open PeopleSoft Application Designer.
3. Open the field whose translate values you want to access.
4. From the File menu, select Definition Properties.
5. Access 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.

Translating Messages

When you translate messages, you translate the message set description, the messages, and the detailed explanations of the messages into a specific target language.

The following table lists the objects, base tables and related language tables related to translating message sets and message catalogs.

Object	Base Table	Related Language Table	Where to Translate
Message Sets	PSMSGSETDEFN	PSMSGSETLANG	Translate Messages page.
Message Catalog	PSMSGCATDEFN	PSMSGCATLANG	Translate Messages page.

To translate messages, use the Translate Message page. To access this page, select PeopleTools, Translate System Definitions, Messages.

Translate Messages

Selection Criteria

*Message Set Number:

 Message Number:
 To:
 *Language Code:

Get Messages

Messages
 Find | View All
First 1 of 1 Last

Base/Target Last Updated:

Base Message Text:

 Base Explanation:

 Target Message Text:

 Target Explanation:

Translate Messages page

To translate messages:

1. Access the Translate Messages page.

2. 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.

3. Select messages within the message set with which to work.

- 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.

4. Select the target language.

- a. From the Language Code drop-down list box, 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.

5. 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 displays.
- 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.

6. 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.

7. Click Save.

8. Repeat steps 6 and 7 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.

Translating Menus

The following table lists where to translate menu and component objects:

Object	Base Table	Related Language Table	Where to Translate
Menu Definitions	PSMENUDEFN	PSMENUDEFNLANG	Translate Menu page.
Menu Items	PSMENUITEM	PSMENUITEMLANG	Translate Menu Item page.

Object	Base Table	Related Language Table	Where to Translate
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 PeopleSoft Application Designer.
- Translate folder tab labels using the translation utilities.

Translating Menu Definitions

Use the Translate Menus page to translate menu definitions and menu group labels.

In a portal, the names that appear in the menu are actually content references. You must translate these using the Translate Menus Page. Use PeopleSoft Application Designer to change only popup menus.

To translate menus:

1. Select PeopleTools, Translations, Translate System Definitions, Menus.

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

2. In the Target Language field, select the language into which you want to translate the message set.
3. In the Menu Name field, select a set of menu definitions.

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

If you want to work through the menus alphabetically, enter the first characters of the menu name in the Menu Name field. If you want to translate a specific menu, enter the menu name in the field or click the Search button to prompt for a specific menu.

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 Microsoft Windows client navigation, position the accelerator key ampersand to the left of the character that will serve as the accelerator key for the menu or menu group.

6. Save the page.

Translate Menu Items

To translate menu items:

1. Select PeopleTools, Translations, Translate System Definitions, Menu Items.

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

2. In the Target Language field, select the language into which to translate the menu bar and menu item labels.
3. In the Menu Name field, select a set of menu definitions.

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

If you want to work through the menus alphabetically, enter the first characters of the menu name in the Menu Name field. If you want to translate a specific menu, enter the menu name in the field or click the Search button to prompt for a specific record.

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 Microsoft Windows client navigation, position the accelerator key ampersand to the left of the character that will serve as the accelerator key for the menu or menu group.

After you translate a menu bar label and exit the field, you must select whether to use the same translation for all occurrences of the bar label:

- Click *Yes* 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 save the page.
- Click *No* if you want to leave some of the menu items untranslated.

6. Save the page.

Translating Folder Tabs in PeopleSoft 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 PeopleSoft 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 PeopleSoft Application Designer:

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

To translate folder tab labels using the translation utilities:

1. Select PeopleTools, Translations, Translate System Definitions, Components.

The Translate Components page displays. 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.

2. In the Target Language field, select the language into which to translate the component labels.
3. In the Component Name edit box, select a set of component definitions.

Leave the Component Name field blank to select all the components in the system.

If you want to work through the components alphabetically, enter the first characters of the component name in the Component Name field. If you want to translate a specific component, enter the component name into the edit box or click the Search button to prompt for a specific record.

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 Microsoft Windows client navigation, position the accelerator key ampersand to the left of the character that will serve as the accelerator key for the menu or menu group.

6. Save the page.

Translating Business Processes

Translate menus before business processes, PeopleSoft Process Scheduler, and portals because the terms are related.

The following table lists where to translate business process objects:

Object	Base Table	Related Language Table	Where to Translate
Business Process Definition	PSBUSPROCDEFN	PSBUSPROCLANG	PeopleSoft Application Designer.
Activities	PSACTIVITYDEFN	PSACTIVITYLANG	PeopleSoft Application Designer.
Business Process Items	PSBUSPROCITEM	PSBUSPROCITEMLANG	PeopleSoft Application Designer.
Component Interfaces	PSBCDEFN	PSBCDEFNLANG	Translate Component Interfaces page.
Business Interlinks	PSIODEFN	PSIODEFNLANG	Translate Business Interlinks page.

This section discusses how to translate:

- Business process definitions (and their related objects).
- Component Interface messages.
- Business Interlinks.

Translating Business Process Definitions (and their related objects)

To translate business process definitions:

1. Set the language in PeopleSoft Configuration Manager.
2. Open PeopleSoft Application Designer.
3. Open the Business Process Map.
4. Overwrite the English with the translation.
5. Save.

Translating Component Interface Messages

To translate component interface messages:

1. Select PeopleTools, Translations, Translate System Definitions, Component Interfaces.

The Translate Component Interfaces page displays. 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.

2. In the Target Language field, select the language into which to translate the message set.
3. In the Name field, enter a component interface.

Click the Search button to prompt for a list of component interfaces.

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 OK to accept the change and return to the Translate Component Interfaces page.
6. Save the page.

Translating Business Interlinks

To translate interface objects:

1. Select PeopleTools, Translations, Translate System Definitions, Business Interlinks.
The Translate Business Interlinks page displays. The Translate Business Interlinks page enables you to select the base description of an interface object name and then translate the description into a specific target language.
2. In the Target Language field, select the language into which to translate.
3. In the IO Name field, enter an Interface Object name.
Click the Search button to prompt for a list of interface object names.
4. Click the Get Objects button to bring the interface object names into the definition list.
A set of interface object names appears in the Look Up List.
5. Translate the base descriptions into the target language.
Enter descriptions for the interface object name in the target language.
6. Save the page.

Translating PeopleSoft Process Scheduler

PeopleSoft Process Scheduler has three language-sensitive definitions:

- Process type definitions.
- Job definitions.
- Process definitions.

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	PRCSTPEDEFN	PRCSTPEDEFNLNG	PeopleTools, Process Scheduler, Process Types.

Object	Base Table	Related Language Table	Where to Translate
Process Jobs	PRCSJOBDEFN	PRCSJOBDEFNLANG	PeopleTools, Process Scheduler, Jobs.
Process Recurrences	PRCSRECUR	PRCSRECURLANG	PeopleTools, Process Scheduler, Recurrences.

To translate a PeopleSoft Process Scheduler Types, Jobs and Recurrences definitions:

1. Sign on to the PeopleSoft system using the target language.
2. Select PeopleTools, 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 Job.
 - To open a process definition, select Recurrences.
4. Translate the descriptive text fields into the target language.
Translate the Description field.
5. Save the page.

Translating Process Definitions

To translate process definitions:

1. Select PeopleTools, Translations, Translate System Definitions, Process Definitions.
The Translate Process Definitions page displays. 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.
2. In the Target Language field, 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 Look Up List.
5. Translate the process definitions description into the target language.
Enter descriptions for the process definitions in the target language.
6. Save the page.

Translating Mobile Data

This section describes how to translate:

- Mobile page names.
- Mobile view labels.

- Mobile page items, such as fields.

Note. To translate mobile data, you must also translate the `\bin\mobile\res\<lang>\mptext.properties` file.

The following table lists where to translate mobile data.

Object	Base Table	Related Language Table	Where To Translate
Mobile page	PSMPDEFN	PSMPDEFN	Translate Mobile Page page.
Mobile page	PSMPDEFNDEV	PSMPDEFNDEVLANG	Translate Mobile Page page.
Mobile Page Items	PSMPCONTDEV	PSMPCONTDEVLANG	Translate Mobile Page Items page.

See Chapter 15, “Adding New Languages,” Adding Text and Error Properties Files for Languages, page 234.

Translating Mobile Page Names and View Labels

Use the Translate Mobile Page Names page to translate Mobile page names and view labels.

Translate Mobile Page

Selection Criteria

Mobile Page Name:

Target Language:

French

Mobile Page List

Find | View All First ◀ 1 of 1 ▶ Last

Mobile Page Name:

CURRENCY_CD_MP

Description:

Currency Code CI

Long Description:

Translate Mobile Page

The fields with which you can work on the Translate Mobile Page are:

Mobile Page Name	Name of the mobile page to translate.
Target Language	From the drop-down list box select the target language for the translations.
Get Mobile Page	Click the button to refresh the page and display the Mobile Page List section, including the Description and Long Description fields.
Description	Enter the description for the page in the target language.
Long Description	Enter the long description for the page in the target language.

Mobile Page View Labels	Click the button to refresh the page and display the View Labels section, including the Detail View Label field and the List View Label field.
Detail View Label	Enter the detail view label in the target language.
List View Label	Enter the list view label in the target language.

To translate Mobile pages:

1. Select PeopleTools, Translations, Translate System Definitions, Mobile Pages. The Translate Mobile Page displays.
2. In the Mobile Page Name field, enter the name of the page to translate, or use the Lookup button to search for a page.
3. From the Target Language drop-down list box, select the target language.
4. Click the Get Mobile Page button. The page refreshes and the Mobile Page List displays.
5. Translate the Mobile page name:

Note. If available, the descriptions that you can translate display next to the fields in the base language.

- a. In the Description field, enter a translation for the page description in the target language.
- b. In the Long Description field, enter a translation for the long description in the target language.

Note. A default description in the target language may display in the field. If so, clear the field then enter a new description in the target language.

6. Translate the Mobile view labels:

Note. If available, the descriptions that you can translate display next to the fields in the base language.

- a. Click the Mobile Page View Labels button. The page refreshes and the View Labels page displays.
- b. In the Detail View Label field, enter a translation for the Detail View in the target language.
- c. In the List View Label field, enter a translation for the List View in the target language.

Note. A default description in the target language may display in the field. If so, clear the field then enter a new description in the target language.

- d. Click OK.

7. Click Save.

Translating Mobile Page Items

To translate Mobile page items use the Translate Mobile Page Items page.

This section provides an overview of the Translate Mobile Page Items page and describes how to:

- Select a mobile page and mobile page item(s) to translate.
- Choose predefined field label descriptions in the target language.
- Specify static text for a field label description in the target language.
- Suppress the display of field descriptions.

Translate Mobile Page Items page

The fields with which you can work on this page are:

Mobile Page Name	Name of the mobile page that contains the items to translate.
Target Language	Select from the drop-down list box the target language for the translations of the mobile page items.
Get Mobile Item	Click this button to retrieve the mobile page items for the selected mobile page name.
Parent/Item	This field is read-only. The Parent is the collection or object that contains the item. The Item is a collection or field on a page.
Base Label Text	Name of the mobile page item to be translated, in the base language.
Related Lang Short/Long/Text	Field that displays or where you enter the translation for the base label text. The translation that displays in this field is based upon your selection in the Label Type box. To the right of this field, the mobile device to which this field applies, such as laptop, PDA and so forth, displays.
View Label	Click this button to apply changes to the Related Lang Short/Long/Text field.
Related Lang Label ID	Click the Look Up button to select a different label ID and its associated long or short description.
Label Type	Determines the description of the label ID that displays in the Related Lang Short/Long/Text field when you click the View Labels button The options are: • None • Text • RFT Short • RFT Long
Detail View Label	In the target language, enter a label for the detail view of the page.
List View Label	In the target language, enter a label for the list view of the page.

Tab Label In the target language, enter a label for the tab.

To select a mobile page and mobile page item(s) to translate:

1. Select PeopleTools, Translations, Translate System Definitions, Mobile Page Items. The Translate Mobile Page Item page displays.
2. In the Mobile Page Name field, enter the name of the page that contains the page items you want to translate, or use the Lookup button to search for the page.
3. From the Target Language drop-down list box, select the target language.
4. Click the Get Mobile Item button.

The page refreshes and in the Mobile Page Items section, the first item associated with the Mobile page displays.

Click View All, or the Show Previous and Show Next buttons, to navigate through the results.

To choose predefined field label descriptions in the target language:

1. Select a mobile page with which to work.
2. In the Label Type box click the RFT Long button or the RFT Short button.
3. Click View Labels. The corresponding predefined RFT long or RFT short translation displays in the Related Lang Short/Long/Text field.
4. Click Save.

To specify static text for a field label description in the target language:

1. Select a mobile page and mobile page item with which to work.
2. In the Label Type box, click Text.
3. Click the View Labels button. Notice the value in the Related Lang Short/Long/Text field clears.
4. In the Related Lang Short/Long/Text field, enter a translation in the target language.
5. Click Save.

To suppress the display of a field description:

1. Select a mobile page and the mobile page item with which to work.
2. In the Label Type box click the None button.
3. Click Save.

Translating 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.

To translate portal objects:

1. Select PeopleTools, Translations, Translate System Definitions, Portal Objects. The Translate Portal Objects page displays.
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.
2. In the Target Language field, select the language into which to translate the Portal Objects.
3. Enter the Portal Object Name and 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 Look Up List.
5. Translate the Portal Objects description and label into the target language.
Enter descriptions for the portal objects in the target language.
6. Save the page.

Translating 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.

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, 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 displays.
6. In the Description field, enter a translation in the target language.
7. Click OK.
8. Save the page.

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, 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. Save the page.

Translating Records

The following table lists where to translate record definitions:

Object	Base Table	Related Language Table	Where to Translate
Record Definitions	PSRECDEFN	PSRECDEFNLANG	Translate Records page.

To translate record descriptions:

1. Select PeopleTools, Translations, Translate System Definitions, Records. The Translate Records page displays.
The Translate Records page lets you select a set of record definitions and then translate the short and long descriptions for those records into a specific target language.
2. In the Target Language field, select the language into which to translate the record descriptions.
3. In the Record field, select a set of record definitions.
Leave the Record field blank if you want to select all record definitions in the system.
If you want to work through the records alphabetically, enter the first characters of the record name in the Record field. If you want to translate a specific record, enter the record name in the Record field or use the Search button to prompt for a specific record.
4. Click the Get Record button to bring the record descriptions into the record list.
A set of record definitions appears in the Record List 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. Save the page.

Translating 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.

To translate queries using the Translate Query utility:

1. Select PeopleTools, Translations, Translate System Definition, Queries. The Translate Query page displays.
The Queries page enables you to translate query descriptions, heading labels, and query prompt descriptions into a specific target language.
2. In the Target Language field, select the language into which to translate the query descriptions and labels.
3. In the Query Name field, select a query.
Leave the Query Name field blank if you want to select all the queries in the system.
If you want to work through the fields alphabetically, enter the first characters of the query name in the Query Name field. If you want to translate a specific query, enter the query name in the Query Name field, or click the Search button to prompt for a specific record.
4. Click the Get Query button to bring the query descriptions into the query list.
A set of query descriptions 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 displays 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.
7. Save the page.

See Also

Enterprise PeopleTools 8.49 PeopleBook: PeopleSoft Query

Translating 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.

Translate Strings page

Translating String Definitions into a Non-Base Language

To translate string definitions into a non-base language:

1. Select PeopleTools, Translations, Translate System Definitions, Strings. The Translate Strings page displays.
2. In the Program ID field, enter or search for a program ID.
3. From the Language Code drop-down list box, 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.
- The String ID matches a field name in the database whose labels have *not* been translated. In this case, the message “Untranslated 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. Save the page.

See Also

Chapter 9, “Using Global Reporting and Data Analysis Tools,” Using the Strings Table for Language-Sensitive Text in Reports, page 122

Translating Application Engine Descriptions

The following table lists where to translate PeopleSoft 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

To translate Application Engine programs:

1. Select PeopleTools, Translations, Translate System Definitions, Application Engine Programs. The Translate Application Engine Programs page displays.

This page enables you to translate Application Engine program descriptions into a specific target language.

2. In the Target Language field, select the language into which to translate.
3. In the Program field, select an Application Engine program.

Leave the Program name field blank if you want to select all the programs in the system.

If you want to work through the fields alphabetically, enter the first characters of the program name in the Program field. If you want to translate a specific Application Engine program, enter the program name in the Program field, or click the Search button to prompt for a specific record.

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. Save the page.

See *Enterprise PeopleTools 8.49 PeopleBook: PeopleSoft Application Engine*, “Understanding PeopleSoft Application Engine”.

Translating 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 PeopleSoft Application Designer.

To check for override text in pages:

1. Select PeopleTools, Translations, Translate System Definitions, Review Page Text. The Review Page Text page displays. 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.

Leave the Page Name field blank if you want to select all the pages in the system.

If you want to work through the pages alphabetically, enter the first characters of the page name in the Page Name field. If you want to translate a specific page, enter the page name in the Page Name field, or click the Search button to prompt for a specific record.

3. In the Language Code field, select the language code of the page set.

For example, if the Page Name field contains *B*, and the Language Code field contains *French*, the page set will consist of page definitions that begin with *B* and whose language code is *FRA*.

4. Click the Get Page Text button to display the overridden page text.

A set of page definitions appears based on the settings in the Page Name and Language Code fields.

For each page that appears in the list, the label text is the override text on the page. Looking at the label text, you can tell whether it has been translated. If it has not been translated, use Translation Designer to do the translation.

See [Chapter 17, “Translating Application Definitions,” Using the Translation Designer, page 245](#).

5. Save the page.

Translating Application Data

The Application Data page enables you to translate system and customer data. This page does not allow you to translate managed objects.

To access the Translate Application Data page, select PeopleTools, Translations, Translate System Definitions, Application Data.

Note. To use the Application Data page you must be logged into the PeopleSoft Pure Internet Architecture (PIA) in the base language.

Before you can use this feature, 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.

Enabling Access to the Translate Application Data Page

To enable access to the Translate Application Data page:

1. Select PeopleTools, Security, Permissions & Roles, Permission Lists. The Permissions List page displays.
2. Click the permission list that your user ID is using. Your permission list displays.
3. Click the Pages tab.
4. Navigate to the Translate menu name and click the Edit Components link. The Component Permissions list displays.
5. Navigate to the TRANSLATE_DATA component and click the Edit Pages link. The Page Permissions page displays.
6. Click the Select All button.
7. Click the OK button, and then click the Save button.
8. Repeat steps 2 through 7 for other groups that need access to the page.
9. Select PeopleTools, Portal, Structure and Content.
10. Click PeopleTools.
11. Click Translate.
12. Click Translate System Definitions.
13. Click Edit for the Application Data row.
14. Clear the Hide in Navigation check box.
15. Save your changes.

Translating Application Data

To translate application data:

1. Select PeopleTools, Translations, Translate System Definitions, Application Data. The Translate Application Data page displays.
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. Select the record fields to translate and click Search.

When you click Search, the page displays 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 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 displays 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.

5. 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.

6. Click Save.

Translating HTML Definitions

You typically translate HTML definitions when they contain hardcoded text. Most HTML definitions should contain JavaScript and references to the Message Catalog, only and therefore don't require separate translation, however it is possible to hardcode text into HTML definitions which may require translation using this method.

To translate an HTML definition:

1. Change the PeopleSoft Configuration Manager language preference to the target language.
2. Login to the PeopleSoft system.
3. Open PeopleSoft Application Designer.
4. Open the HTML definition.
Choose File, Open and select the HTML definition you want to open.
5. Edit any language-sensitive text in the HTML object.
6. Save the HTML object.

Translating PeopleSoft Integration Broker Definitions

PeopleTools features several pages that you can use to translate PeopleSoft Integration Broker data into a specific target language

The following table lists where to translate PeopleSoft Integration Broker objects:

Object	Base Table	Related Language Table	Where to Translate
Handler	PSOPRHDLR	PSOPRHDLRLANG	Translate Service Operation Handlers page.
Message Definitions	PSMSGDEFN	PSMSGDEFNLANG	Translate Message Definitions.
Queue	PSQUEUEDEFN	PSQUEUEDEFNLANG	Translate Queues page.
Operation Version	PSPRSMATTR	PSPRSMATTRLANG	Translate Service Operation Versions page.
Routing	PSIBRTNGDEFN	PSRTNGDFNLANG	Translate IB Routings page.
Service	PSSERVICE	PSSERVICELANG	Translate IB Services page.
Service Operation	PSOPERATION	PSOPERATIONLANG	Translate Service Operations page.

The following steps for translating PeopleSoft Integration Broker objects apply to all of the objects in the table.

To translate a PeopleSoft Integration Broker object:

1. Select PeopleTools, Translations, Translate System Definition.
The Translate System Definitions menu appears.
2. In the left navigation area, click the menu item that corresponds to the object to translate. The choices are:
 - *Message Definitions.*
 - *Message Queues.*
 - *Routings.*
 - *Services.*
 - *Service Operation Handlers.*
 - *Service Operations.*
 - *Service Operation Versions.*

The corresponding translation page appears.

3. In the Target Language field, select the language into which to translate the record descriptions.
4. 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 click the Lookup button to search for one.
Queues	In the Queue Name field, enter the name of the queue to translate, or click the Lookup button to search for one.
Routings	In the Routing Name field, enter the name of the routing to translate, or click the Lookup button to search for one.
Services	In the Service field, enter the name of the service to translate, or click the Lookup button to search for one.
Service Operation Handler	In the Operation field, enter the name of the service operation handler to translate, or click the Lookup button to search for one.
Service Operations	In the Operation field, enter the name of the service operation to translate, or click the Lookup button to search for one.
Service Operation Versions	In the Operation field, enter the name of the service operation that contains the version to translate, or click the Lookup button to search for one.

5. Click the Get Objects button to bring the definition name, description and long descriptions into the object list.

Note. If no descriptions exist in the base language, no results are returned.

6. Translate the base-language descriptions into the target language.
The descriptions appear on the left; enter the translations in the Target Description fields on the right.
7. Save the page.

Translating XML Publisher Definitions

PeopleTools features several pages that you can use to translate XML Publisher data into a specific target language

The following table lists where to translate XML Publisher objects:

Object	Base Table	Related Language Table	Where to Translate
XMLP Data Source	PSXPDATASRC	PSXPDATASRCLNG	Translate XMLP Data Source Definitions

Object	Base Table	Related Language Table	Where to Translate
XMLP Report Definition	PSXPRPTDEFN	PSXPRPTDEFNLNG	Translate XMLP Report Definitions
XMLP Template Definition	PSXPTMPLDEFN	PSXPTMPLDEFNLNG	Translate XMLP Definitions Template

The following steps for translating XML Publisher objects apply to all of the objects in the table.

To translate an XML Publisher object:

1. Select PeopleTools, Translations, Translate System Definition.
The Translate System Definitions menu appears.
2. In the left navigation area, click the menu item that corresponds to the object to translate. The choices are:
 - *XMLP Data Sources*
 - *XMLP Report Definitions*.
 - *XMLP Template Definitions*.

The corresponding translation page appears.

3. In the Target Language field, select the language into which to translate the record descriptions.
4. Based on the object descriptions that you are translating, do the following:

Object Descriptions to Translate	Action
XMLP Data Sources	In the Data Source Type field, enter the data source type to translate, or click the Lookup button to search for one.
XMLP Report Definitions	In the Report Name field, enter the name of the report to translate, or click the Lookup button to search for one.
XMLP Template Definitions	In the Template ID field, enter the ID of the template to translate, or click the Lookup button to search for one.

5. Click the Get Objects button to bring the definition name, description and long descriptions into the object list.

Note. If no descriptions exist in the base language, no results are returned.

6. Translate the base-language descriptions into the target language.
The descriptions appear on the left; enter the translations in the Target Description fields on the right.
7. Save the page.

Translating 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 access the Translate Analytic Model page, select PeopleTools, Translations, Translate System Definitions, Translate Analytic Model.

To translate an ACE analytic model ID definition:

1. In the Model ID field, select the analytic model ID to translate.
Leave the Model ID field blank if you want to select all analytic model ID definitions in the system.
If you want to work through the records alphabetically, enter the first characters of the record name in the Model ID field. If you want to translate a specific record, enter the record name in the Model ID field or use the Search button to prompt for a specific record.
2. In the Target Language field, select the language into which to translate the record descriptions.
3. 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.
4. 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.
5. Save the page.

Translating Time Zone Labels

There are two ways to translate time zone labels in PeopleTools: using the Translation page or using the Time Zone Maintenance page.

Translating Time Zone Labels Using the Translation Page

To translate time zone labels using the Translation page.

1. Select PeopleTools, Translations, Translate System Definitions, Application Data.
2. In the Record to Translate field, enter *PSTIMEZONE*.
3. From the Target Language dropdown list box, select the language into which to translate the time zone label.
4. Click the Save button.

Translating Timezone Labels Using the Timezone Maintenance Page

To translate time zone labels using the Timezone Maintenance page:

1. Select PeopleTools, Utilities, International, Time Zones.
The Time Zone Data page appears.
2. Locate a time zone with which to work from the list.

3. In the Time Zone Standard Label field, enter a label for the time zone for when standard time is in effect.
4. In the Time Zone Daylight Label field, enter a label for the time zone for when daylight savings time is in effect (if applicable).
5. Click the Save button.

APPENDIX A

Converting PeopleSoft Systems to Unicode Databases

This appendix discusses:

- Planning Unicode Conversions.
- Converting to Unicode on Oracle Databases.
- Converting to Unicode on Microsoft SQL Server and DB2/UDB Databases.

Understanding Converting PeopleSoft Systems to Unicode Databases

PeopleSoft 8 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.

Note. The information described in this appendix is valid on PeopleSoft 8.x databases. It is not possible to convert a database to Unicode on PeopleTools 7.x or earlier releases.

Upgrades from PeopleSoft 7.0x or 7.5x non-Unicode databases directly to PeopleSoft 8.x Unicode databases are not certified.

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 *Enterprise PeopleTools 8.49 Hardware and Software Requirements* book for your PeopleTools release.

Unix and Microsoft Windows editions of the PeopleSoft application server support Unicode. There are no special settings or configuration options you need to set to enable this support.

SQR 6.1.4 or later versions may be bundled with your PeopleSoft applications. These versions of SQR are Unicode-enabled, however they 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.

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. A red paper on Unicode database sizing is available at http://www.peoplesoft.com/media/en/pdf/red_paper_unicode.pdf

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 database. 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.

Converting to Unicode on Oracle Databases

This section discusses how to:

- Export table structures using Data Mover scripts.
- Export database contents using the Oracle Export utility.
- Create new Oracle8i database instances using the UTF-8 character set.
- Pre-create Unicode database table structures using PeopleSoft Data Mover scripts.
- Import database contents using the Oracle Import utility.
- Specify Unicode databases in PeopleSoft systems.
- Run GRANT.SQL.
- Rerun Oracle database imports.

Understanding Converting PeopleSoft 8.x Systems on Oracle Databases to Unicode

Converting a PeopleSoft 8.x system on an Oracle database to Unicode requires creating a new Oracle instance with a UTF-8 character set and, if using byte semantics, new column sizes for PeopleSoft columns created as VARCHAR2.

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 the UTF-8 character set, and re-import the entire database using PeopleSoft Data Mover.

For larger databases, a better process is to use Data Mover to pre-build the structure of the tables and then use Oracle export/import to move the bulk of the data between the old and new Oracle instances.

Exporting Database Table Structures Using PeopleSoft Data Mover

This section describes how to use PeopleSoft Data Mover to export the database structure to a DAT file.

Note. This process exports the database table structure only and does not include the database contents.

Sign onto the PeopleSoft 8.x database using PeopleSoft Data Mover in non-bootstrap mode, and run the following script.

```
SET NO TRACE;  
SET OUTPUT <output file>;  
SET NO DATA;  
EXPORT *;
```

The DAT file created by PeopleSoft Data Mover will not be large – possibly 20-30 MB, as it will contain only the structure of your PeopleSoft tables, views and indexes– not the data.

Save the file created by the process. It will be referred to as the *structural export* in this appendix.

Exporting Database Contents Using the Oracle Export Utility

This section discusses how to:

- Set the NLS_LANG variable.
- Create an export parameter file.
- Export database contents.

Understanding Exporting Database Contents Using the Oracle Export Utility

On the server machine running the Oracle 8i database, use the Oracle export command (exp) to export all objects owned by the PeopleSoft OwnerID.

Setting the NLS_LANG Variable

Before running the export process, verify that the character set component of the NLS_LANG setting 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.

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 component.

To verify the current character set of the Oracle database, run the following SQL statement in SQL*Plus while logged in as a database administrator:

```
SELECT VALUE FROM SYS.V_$NLS_PARAMETERS  
WHERE PARAMETER = 'NLS_CHARACTERSET'  
/
```

Creating Export Parameter Files

After you have set the NLS_LANG environment variable appropriately based on your instance's current character set, create an export parameter file similar to the following example, substituting the appropriate username/password of the Owner ID and the location and name of the export file to be created.

You can use any text editor to create this file.

Note. Ensure that the files 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 OwnerID and adding 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. If you have insufficient space on any one file system to store the export file, you can use the filesize parameter of the Oracle Export utility. See your Oracle Utilities guide for more information on multi-file exports.

```
userid=<ownerid>/<password>
buffer=100000
file=<outputfile>.dmp
rows=Y
indexes=Y
log=<logfile>.txt
compress=Y
owner=<ownerid>
```

Exporting Database Contents

After you create the export parameter file as described in the previous section, call the Oracle export using the following command, substituting <parfile> with the name of the export parameter file.

```
exp parfile=<parfile>
```

The export (.dmp) file created by this process will be referred to as the “database export” in this document.

Creating New Oracle8i Database Instances Using the UTF-8 Character Set

This section discusses how to:

- Create a new Oracle8i database instance using the UTF-8 character set.
- Pre-create objects in the new Oracle database instance.

Creating New Oracle Database Instances

Next, you must create a new Oracle instance with a Unicode encoding (UTF-8) which 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 OwnerID 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.

When you create the new Oracle instance, be sure to specify *CHARACTER SET UTF8* at the end of the CREATE DATABASE command. See the *Oracle SQL Reference* guide for syntax information on the CREATE DATABASE SQL command.

Pre-Creating Objects in New Oracle Database Instances

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.
- The same tablespaces as used by the PeopleSoft OwnerID in the existing PeopleSoft database.
- The PS User ID and the PSDBOWNER table. To create these, run the DBOWNER.SQL script provided by PeopleSoft through SQL*Plus. 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 OwnerID and ConnectID with the same name as in the existing PeopleSoft database. You can create this ID by running the PSROLES.SQL, PSADMIN.SQL and CONNECT.SQL scripts through SQL*Plus. See the Enterprise PeopleTools installation guide for details on running these scripts at installation time.

See *Enterprise PeopleTools 8.49 Installation for your database platform*.

- Set up appropriate LISTENER.ORA and TNSNAMES.ORA entries for the new instance.

Pre-Creating Unicode Database Table Structures Using PeopleSoft Data Mover

Sign on to the newly created Oracle database using Data Mover. Login in bootstrap mode using the OwnerID and password.

Once logged in, execute the following Data Mover script, substituting the name of the structural export file you created in PeopleSoft Data Mover previously. Include the SET NO STATISTICS; command only if you are running PeopleTools 8.40 or higher. Without this command on PeopleTools 8.40, the Data Mover import process will fail once the PSDDLMODEL table is created:

```
SET NO TRACE;
SET NO DATA;
SET INPUT <input file>;
SET UNICODE ON;
SET STATISTICS OFF; /* Only include this statement if you are running PeopleTools⇒
8.40 or higher */
SET NO INDEX;
IMPORT *;
```

This script creates all the tables in the original PeopleSoft database in the new database with no rows of data, and without indexes. Indexes are omitted in this step, as it much more efficient to create indexes after loading table data as compared to loading data into pre-indexed tables.

Importing Database Contents Using the Oracle Import Utility

This section discusses how to:

- Clear the NLS_LANG variable.
- Create import parameter files.
- Run the Oracle Import utility.
- Rebuild temporary tables.

Understanding Importing Database Contents Using the Oracle Import Utility

Once the tables have been pre-created by Data Mover, you are ready to import the database export you created. You perform this import using Oracle's import utility (imp).

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 ensures that Oracle reads the character set information from the export file and performs the appropriate conversion to the database's UTF-8 character set.

Creating Import Parameter Files

Create an import parameter file, similar to the export parameter file you created during the export step. This file should have the following commands:

```
userid=<ownerid>/<password>
buffer=100000
file=<database export file>.dmp
rows=Y
indexes=Y
log=<logfile>.txt
ignore=Y
full=Y
```

Running the Oracle Import Utility

Run the Oracle Import utility from the command line using the following command, substituting in the name of the file containing the above parameters.

```
imp parfile=<parfile>
```

Ensure that the import completes successfully with no errors before continuing.

Rebuild Temporary Tables

After you run the Oracle Import Utility, you must rebuild the temporary tables.

To rebuild the temporary tables, run the following data mover command:

```
CREATE_TEMP_TABLE *
```

Specifying Unicode Databases in PeopleSoft Systems

When the import has completed successfully you must specify in PeopleSoft that the database is now a Unicode database. To do this, login to SQL*Plus as the database owner ID, and run the following SQL statement:

```
UPDATE PSSTATUS SET UNICODE_ENABLED=1;
```

See Also

Appendix A, "Converting PeopleSoft Systems to Unicode Databases," Rerunning Oracle Database Imports, page 288

Running GRANT.SQL

You are now ready to sign into the new PeopleSoft Unicode database. Log into SQL*Plus, and edit and run GRANT.SQL (provided in your PeopleTools installation in the SCRIPTS directory), to set up the appropriate grants to the ConnectID you created previously. Remember to edit the script first to include the correct Connect ID.

You should now be able to log into your new PeopleSoft Unicode database using PeopleSoft Application Designer.

Rerunning Oracle Database Imports

When the PSSTATUS table is imported from a non-Unicode export file, the UNICODE_ENABLED flag is 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.

See Also

[Appendix A, “Converting PeopleSoft Systems to Unicode Databases,” Specifying Unicode Databases in PeopleSoft Systems, page 289](#)

Converting to Unicode on Microsoft SQL Server and DB2/UDB Databases

This section discusses how to:

- Export a PeopleSoft database with PeopleSoft Data Mover.
- Create a new PTSYS Unicode database.
- Import database contents using PeopleSoft Data Mover.
- Specify the Unicode database in the PeopleSoft system.
- Rerun GRANT.SQL.
- Rerun Microsoft SQL Server and DB2/UDB database imports.

Understanding Converting to Unicode on Microsoft SQL Server and DB2/UDB Databases

On Microsoft SQL Server and DB2/UDB databases, PeopleSoft supports converting a PeopleSoft 8.x database to Unicode by re-creating the database using Data Mover to import database contents. 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 NCHAR columns in a single PeopleSoft database is not supported.

Exporting PeopleSoft Databases Using PeopleSoft Data Mover

Sign onto the PeopleSoft 8.x database using PeopleSoft Data Mover in non-bootstrap mode, and run the following script.

```
SET NO TRACE;
SET OUTPUT <output file>;
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/UDB PTSYS Unicode Databases

You will need to create a new SQL Server or DB2/UDB PTSYS database corresponding to your target PeopleSoft 8.x 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 Enterprise 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.

See *Enterprise PeopleTools 8.49 Installation for Microsoft SQL Server*, “Creating a Database Manually.”

See *Enterprise PeopleTools 8.49 Installation for DB2 UDB for Linux, UNIX, and Windows*, “Creating a Database Manually.”

See *Enterprise PeopleTools 8.49 Installation for your database platform*, “Creating a Database.”

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 <input file>;
SET UNICODE ON;
IMPORT *;
```

This will import the entire contents of your original database into the new database, while creating character columns using the NCHAR data type (for SQL Server) or VARBINARY data type (for DB2), and converting your data to Unicode.

Specifying Unicode Databases in PeopleSoft Systems

Once the database has finished importing, run the following SQL statement either through PeopleSoft Data Mover, PeopleSoft Query Analyzer (ISQLW.EXE), or db2connect.

```
UPDATE PSSTATUS SET UNICODE_ENABLED=1
```

This sets a flag that specifies to PeopleSoft that it is operating against a Unicode database.

See Also

Appendix A, “Converting PeopleSoft Systems to Unicode Databases,” Rerunning Microsoft SQL Server and DB2/UDB Database Imports, page 290

Rerunning GRANT.SQL

Before signing onto PeopleSoft Application Designer to create your views, you must first re-run GRANT.SQL (provided in your PeopleTools installation in the SCRIPTS directory) to grant access to the newly created tables to your ConnectID. Once this is completed, you should be able to sign into PeopleSoft Application Designer.

Rerunning Microsoft SQL Server and DB2/UDB 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.

See Also

Appendix A, “Converting PeopleSoft Systems to Unicode Databases,” Specifying Unicode Databases in PeopleSoft Systems, page 289

Glossary of PeopleSoft Enterprise Terms

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

	Administration, PeopleSoft Enterprise Stock Administration, and the COBRA Administration feature of the Base Benefits business process.
action template	In PeopleSoft Enterprise Receivables, outlines a set of escalating actions that the system or user performs based on the period of time that a customer or item has been in an action plan for a specific condition.
activity	In PeopleSoft Enterprise Learning Management, an instance of a catalog item (sometimes called a class) that is available for enrollment. The activity defines such things as the costs that are associated with the offering, enrollment limits and deadlines, and waitlisting capacities. In PeopleSoft Enterprise Performance Management, the work of an organization and the aggregation of actions that are used for activity-based costing. In PeopleSoft Enterprise Project Costing, the unit of work that provides a further breakdown of projects—usually into specific tasks. In PeopleSoft Workflow, a specific transaction that you might need to perform in a business process. Because it consists of the steps that are used to perform a transaction, it is also known as a step map.
address usage	In PeopleSoft Enterprise Campus Solutions, a grouping of address types defining the order in which the address types are used. For example, you might define an address usage code to process addresses in the following order: billing address, dormitory address, home address, and then work address.
adjustment calendar	In PeopleSoft Enterprise Campus Solutions, the adjustment calendar controls how a particular charge is adjusted on a student's account when the student drops classes or withdraws from a term. The charge adjustment is based on how much time has elapsed from a predetermined date, and it is determined as a percentage of the original charge amount.
administrative function	In PeopleSoft Enterprise Campus Solutions, a particular functional area that processes checklists, communication, and comments. The administrative function identifies which variable data is added to a person's checklist or communication record when a specific checklist code, communication category, or comment is assigned to the student. This key data enables you to trace that checklist, communication, or comment back to a specific processing event in a functional area.
admit type	In PeopleSoft Enterprise Campus Solutions, a designation used to distinguish first-year applications from transfer applications.
agreement	In PeopleSoft Enterprise eSettlements, provides a way to group and specify processing options, such as payment terms, pay from a bank, and notifications by a buyer and supplier location combination.
allocation rule	In PeopleSoft Enterprise Incentive Management, an expression within compensation plans that enables the system to assign transactions to nodes and participants. During transaction allocation, the allocation engine traverses the compensation structure from the current node to the root node, checking each node for plans that contain allocation rules.
alternate account	A feature in PeopleSoft Enterprise General Ledger that enables you to create a statutory chart of accounts and enter statutory account transactions at the detail transaction level, as required for recording and reporting by some national governments.
analysis database	In PeopleSoft Enterprise Campus Solutions, database tables that store large amounts of student information that may not appear in standard report formats. The analysis database tables contain keys for all objects in a report that an application program can use to reference other student-record objects that are not contained in the printed report. For instance, the analysis database contains data on courses that are considered

	for satisfying a requirement but that are rejected. It also contains information on courses captured by global limits. An analysis database is used in PeopleSoft Enterprise Academic Advisement.
Application Messaging	PeopleSoft Application Messaging enables applications within the PeopleSoft Enterprise product family to communicate synchronously or asynchronously with other PeopleSoft Enterprise and third-party applications. An application message defines the records and fields to be published or subscribed to.
AR specialist	Abbreviation for <i>receivables specialist</i> . In PeopleSoft Enterprise Receivables, an individual in who tracks and resolves deductions and disputed items.
arbitration plan	The arbiter when multiple price rules match the transaction. This plan determines the order in which the price rules are applied to the transaction base price.
assessment rule	In PeopleSoft Enterprise Receivables, a user-defined rule that the system uses to evaluate the condition of a customer's account or of individual items to determine whether to generate a follow-up action.
asset class	An asset group used for reporting purposes. It can be used in conjunction with the asset category to refine asset classification.
attribute/value pair	In PeopleSoft Enterprise Directory Interface, relates the data that makes up an entry in the directory information tree.
auction event	In PeopleSoft Strategic Sourcing, a sourcing event where bidders actively compete against one another to achieve the best price or score.
audience	In PeopleSoft Enterprise Campus Solutions, a segment of the database that relates to an initiative, or a membership organization that is based on constituent attributes rather than a dues-paying structure. Examples of audiences include the Class of '65 and Undergraduate Arts & Sciences.
authentication server	A server that is set up to verify users of the system.
base time period	In PeopleSoft Enterprise Business Planning, the lowest level time period in a calendar.
benchmark job	In PeopleSoft Enterprise Workforce Analytics Solution, a benchmark job is a job code for which there is corresponding salary survey data from published, third-party sources.
bid response	In PeopleSoft Strategic Sourcing, the response by a bidder to an event.
billing career	In PeopleSoft Enterprise Campus Solutions, the one career under which other careers are grouped for billing purposes if a student is active simultaneously in multiple careers.
bio bit or bio brief	In PeopleSoft Enterprise Campus Solutions, a report that summarizes information stored in the system about a particular constituent. You can generate standard or specialized reports.
book	In PeopleSoft Enterprise Asset Management, used for storing financial and tax information, such as costs, depreciation attributes, and retirement information on assets.
branch	A tree node that rolls up to nodes above it in the hierarchy, as defined in PeopleSoft Tree Manager.
budgetary account only	An account used by the system only and not by users; this type of account does not accept transactions. You can only budget with this account. Formerly called "system-maintained account."

budget check	In commitment control, the processing of source transactions against control budget ledgers, to see if they pass, fail, or pass with a warning.
budget control	In commitment control, budget control ensures that commitments and expenditures don't exceed budgets. It enables you to track transactions against corresponding budgets and terminate a document's cycle if the defined budget conditions are not met. For example, you can prevent a purchase order from being dispatched to a vendor if there are insufficient funds in the related budget to support it.
budget period	The interval of time (such as 12 months or 4 quarters) into which a period is divided for budgetary and reporting purposes. The ChartField allows maximum flexibility to define operational accounting time periods without restriction to only one calendar.
business activity	The name of a subset of a detailed business process. This might be a specific transaction, task, or action that you perform in a business process.
business event	In PeopleSoft Enterprise Receivables, defines the processing characteristics for the Receivable Update process for a draft activity. In PeopleSoft Enterprise Sales Incentive Management, an original business transaction or activity that may justify the creation of a PeopleSoft Enterprise Incentive Management event (a sale, for example).
business process	A standard set of 17 business processes are defined and maintained by the PeopleSoft Enterprise product families and are supported by the Business Process Engineering group. An example of a business process is Order Fulfillment, which is a business process that manages sales orders and contracts, inventory, billing, and so forth. See also <i>detailed business process</i>.
business unit constraints	In PeopleSoft Strategic Sourcing, these constraints apply to a selected Strategic Sourcing business unit. Spend is tracked across all of the events within the selected Strategic Sourcing business unit.
business task	The name of the specific function depicted in one of the business processes.
business unit	A corporation or a subset of a corporation that is independent with regard to one or more operational or accounting functions.
buyer	In PeopleSoft Enterprise eSettlements, an organization (or business unit, as opposed to an individual) that transacts with suppliers (vendors) within the system. A buyer creates payments for purchases that are made in the system.
buy event	In PeopleSoft Strategic Sourcing, for event creators, the purchase of goods or services, most typically associated with a request for quote, proposal, or reverse auction. For bidders, the sale of goods or services.
campus	In PeopleSoft Enterprise Campus Solutions, an entity that is usually associated with a distinct physical administrative unit, that belongs to a single academic institution, that uses a unique course catalog, and that produces a common transcript for students within the same academic career.
cash drawer	A repository for monies and payments taken locally.
catalog item	In PeopleSoft Enterprise Learning Management, a specific topic that a learner can study and have tracked. For example, "Introduction to Microsoft Word." A catalog item contains general information about the topic and includes a course code, description, categorization, keywords, and delivery methods. A catalog item can have one or more learning activities.
catalog map	In PeopleSoft Enterprise Catalog Management, translates values from the catalog source data to the format of the company's catalog.

catalog partner	In PeopleSoft Enterprise Catalog Management, shares responsibility with the enterprise catalog manager for maintaining catalog content.
categorization	Associates partner offerings with catalog offerings and groups them into enterprise catalog categories.
category	In PeopleSoft Enterprise Campus Solutions, a broad grouping to which specific comments or communications (contexts) are assigned. Category codes are also linked to 3C access groups so that you can assign data-entry or view-only privileges across functions.
channel	In PeopleSoft MultiChannel Framework, email, chat, voice (computer telephone integration [CTI]), or a generic event.
ChartField	A field that stores a chart of accounts, resources, and so on, depending on the PeopleSoft Enterprise application. ChartField values represent individual account numbers, department codes, and so forth.
ChartField balancing	You can require specific ChartFields to match up (balance) on the debit and the credit side of a transaction.
ChartField combination edit	The process of editing journal lines for valid ChartField combinations based on user-defined rules.
ChartKey	One or more fields that uniquely identify each row in a table. Some tables contain only one field as the key, while others require a combination.
checkbook	In PeopleSoft Enterprise Promotions Management, enables you to view financial data (such as planned, incurred, and actual amounts) that is related to funds and trade promotions.
checklist code	In PeopleSoft Enterprise Campus Solutions, a code that represents a list of planned or completed action items that can be assigned to a staff member, volunteer, or unit. Checklists enable you to view all action assignments on one page.
claimback	In the wholesale distribution industry, a contract between supplier and distributor, in which monies are paid to the distributor on the sale of specified products or product groups to targeted customers or customer groups.
class	In PeopleSoft Enterprise Campus Solutions, a specific offering of a course component within an academic term. See also <i>course</i> .
Class ChartField	A ChartField value that identifies a unique appropriation budget key when you combine it with a fund, department ID, and program code, as well as a budget period. Formerly called <i>sub-classification</i> .
clearance	In PeopleSoft Enterprise Campus Solutions, the period of time during which a constituent in PeopleSoft Enterprise Contributor Relations is approved for involvement in an initiative or an action. Clearances are used to prevent development officers from making multiple requests to a constituent during the same time period.
clone	In PeopleCode, to make a unique copy. In contrast, to <i>copy</i> may mean making a new reference to an object, so if the underlying object is changed, both the copy and the original change.
cohort	In PeopleSoft Enterprise Campus Solutions, the highest level of the three-level classification structure that you define for enrollment management. You can define a cohort level, link it to other levels, and set enrollment target numbers for it. See also <i>population</i> and <i>division</i> .

collection	To make a set of documents available for searching in Verity, you must first create at least one collection. A collection is set of directories and files that allow search application users to use the Verity search engine to quickly find and display source documents that match search criteria. A collection is a set of statistics and pointers to the source documents, stored in a proprietary format on a file server. Because a collection can only store information for a single location, PeopleTools maintains a set of collections (one per language code) for each search index object.
collection rule	In PeopleSoft Enterprise Receivables, a user-defined rule that defines actions to take for a customer based on both the amount and the number of days past due for outstanding balances.
comm key	See <i>communication key</i> .
communication key	In PeopleSoft Enterprise Campus Solutions, a single code for entering a combination of communication category, communication context, communication method, communication direction, and standard letter code. Communication keys (also called <i>comm keys</i> or <i>speed keys</i>) can be created for background processes as well as for specific users.
compensation object	In PeopleSoft Enterprise Incentive Management, a node within a compensation structure. Compensation objects are the building blocks that make up a compensation structure's hierarchical representation.
compensation structure	In PeopleSoft Enterprise Incentive Management, a hierarchical relationship of compensation objects that represents the compensation-related relationship between the objects.
component interface	A component interface is a set of application programming interfaces (APIs) that you can use to access and modify PeopleSoft Enterprise database information using a program instead of the PeopleSoft client.
condition	In PeopleSoft Enterprise Receivables, occurs when there is a change of status for a customer's account, such as reaching a credit limit or exceeding a user-defined balance due.
configuration parameter catalog	Used to configure an external system with PeopleSoft Enterprise. For example, a configuration parameter catalog might set up configuration and communication parameters for an external server.
configuration plan	In PeopleSoft Enterprise Incentive Management, configuration plans hold allocation information for common variables (not incentive rules) and are attached to a node without a participant. Configuration plans are not processed by transactions.
constituents	In PeopleSoft Enterprise Campus Solutions, friends, alumni, organizations, foundations, or other entities affiliated with the institution, and about which the institution maintains information. The constituent types delivered with PeopleSoft Enterprise Contributor Relations Solutions are based on those defined by the Council for the Advancement and Support of Education (CASE).
constraint	A business policy or rule that affects how a sourcing event is awarded. There are three types of constraints: business, global, and event.
content reference	Content references are pointers to content registered in the portal registry. These are typically either URLs or iScripts. Content references fall into three categories: target content, templates, and template pagelets.
context	In PeopleCode, determines which buffer fields can be contextually referenced and which is the current row of data on each scroll level when a PeopleCode program is running. In PeopleSoft Enterprise Campus Solutions, a specific instance of a comment or communication. One or more contexts are assigned to a category, which you link to

	3C access groups so that you can assign data-entry or view-only privileges across functions.
	In PeopleSoft Enterprise Incentive Management, a mechanism that is used to determine the scope of a processing run. PeopleSoft Enterprise Incentive Management uses three types of context: plan, period, and run-level.
control table	Stores information that controls the processing of an application. This type of processing might be consistent throughout an organization, or it might be used only by portions of the organization for more limited sharing of data.
cost plus contract line	A rate-based contract line associated with a fee component of Award, Fixed, Incentive, or Other. Rate-based contract lines associated with a fee type of None are not considered cost-plus contract lines.
cost plus pricing	In PeopleSoft Enterprise Pricer, a pricing method that begins with cost of goods as the basis.
cost profile	A combination of a receipt cost method, a cost flow, and a deplete cost method. A profile is associated with a cost book and determines how items in that book are valued, as well as how the material movement of the item is valued for the book.
cost row	A cost transaction and amount for a set of ChartFields.
counter sale	A face-to-face customer transaction where the customer typically selects items from the storefront or picks up products that they ordered ahead of time. Customers pay for the goods at the counter and take the goods with them instead of having the goods shipped from a warehouse.
course	In PeopleSoft Enterprise Campus Solutions, a course that is offered by a school and that is typically described in a course catalog. A course has a standard syllabus and credit level; however, these may be modified at the class level. Courses can contain multiple components such as lecture, discussion, and lab. See also <i>class</i> .
course share set	In PeopleSoft Enterprise Campus Solutions, a tag that defines a set of requirement groups that can share courses. Course share sets are used in PeopleSoft Enterprise Academic Advisement.
current learning	In PeopleSoft Enterprise Learning Management, a self-service repository for all of a learner's in-progress learning activities and programs.
data acquisition	In PeopleSoft Enterprise Incentive Management, the process during which raw business transactions are acquired from external source systems and fed into the operational data store (ODS).
data cube	In PeopleSoft Analytic Calculation Engine, a data cube is a container for one kind of data (such as Sales data) and works with in tandem with one or more dimensions. Dimensions and data cubes in PeopleSoft Analytic Calculation Engine are unrelated to dimensions and online analytical processing (OLAP) cubes in PeopleSoft Cube Manager.
data elements	Data elements, at their simplest level, define a subset of data and the rules by which to group them. For Workforce Analytics, data elements are rules that tell the system what measures to retrieve about your workforce groups.
dataset	A data grouping that enables role-based filtering and distribution of data. You can limit the range and quantity of data that is displayed for a user by associating dataset rules with user roles. The result of dataset rules is a set of data that is appropriate for the user's roles.

delivery method	In PeopleSoft Enterprise Learning Management, identifies the primary type of delivery method in which a particular learning activity is offered. Also provides default values for the learning activity, such as cost and language. This is primarily used to help learners search the catalog for the type of delivery from which they learn best. Because PeopleSoft Enterprise Learning Management is a blended learning system, it does not enforce the delivery method. In PeopleSoft Enterprise Supply Chain Management, identifies the method by which goods are shipped to their destinations (such as truck, air, and rail). The delivery method is specified when creating shipment schedules.
delivery method type	In PeopleSoft Enterprise Learning Management, identifies how learning activities can be delivered—for example, through online learning, classroom instruction, seminars, books, and so forth—in an organization. The type determines whether the delivery method includes scheduled components.
detailed business process	A subset of the business process. For example, the detailed business process named Determine Cash Position is a subset of the business process called Cash Management.
dimension	In PeopleSoft Analytic Calculation Engine, a dimension contains a list of one kind of data that can span various contexts, and it is a basic component of an analytic model. Within the analytic model, a dimension is attached to one or more data cubes. In PeopleSoft Cube Manager, a dimension is the most basic component of an OLAP cube and specifies the PeopleSoft metadata to be used to create the dimension's rollup structure. Dimensions and data cubes in PeopleSoft Analytic Calculation Engine are unrelated to dimensions and OLAP cubes in PeopleSoft Cube Manager.
direct receipt	Items shipped from a warehouse or vendor to another warehouse.
direct ship	Items shipped from the vendor or warehouse directly to the customer (formerly referred to as <i>drop ship</i>).
directory information tree	In PeopleSoft Enterprise Directory Interface, the representation of a directory's hierarchical structure.
division	In PeopleSoft Enterprise Campus Solutions, the lowest level of the three-level classification structure that you define in PeopleSoft Enterprise Recruiting and Admissions for enrollment management. You can define a division level, link it to other levels, and set enrollment target numbers for it. See also <i>population</i> and <i>cohort</i>.
document sequencing	A flexible method that sequentially numbers the financial transactions (for example, bills, purchase orders, invoices, and payments) in the system for statutory reporting and for tracking commercial transaction activity.
dynamic detail tree	A tree that takes its detail values—dynamic details—directly from a table in the database, rather than from a range of values that are entered by the user.
edit table	A table in the database that has its own record definition, such as the Department table. As fields are entered into a PeopleSoft Enterprise application, they can be validated against an edit table to ensure data integrity throughout the system.
effective date	A method of dating information in PeopleSoft Enterprise applications. You can predate information to add historical data to your system, or postdate information in order to enter it before it actually goes into effect. By using effective dates, you don't delete values; you enter a new value with a current effective date.
EIM ledger	Abbreviation for <i>Enterprise Incentive Management ledger</i> . In PeopleSoft Enterprise Incentive Management, an object to handle incremental result gathering within the scope of a participant. The ledger captures a result set with all of the appropriate traces to the data origin and to the processing steps of which it is a result.

elimination set	In PeopleSoft Enterprise General Ledger, a related group of intercompany accounts that is processed during consolidations.
entry event	In PeopleSoft Enterprise General Ledger, Receivables, Payables, Purchasing, and Billing, a business process that generates multiple debits and credits resulting from single transactions to produce standard, supplemental accounting entries.
equitization	In PeopleSoft Enterprise General Ledger, a business process that enables parent companies to calculate the net income of subsidiaries on a monthly basis and adjust that amount to increase the investment amount and equity income amount before performing consolidations.
equity item limit	In PeopleSoft Enterprise Campus Solutions, the amounts of funds set by the institution to be awarded with discretionary or gift funds. The limit could be reduced by amounts equal to such things as expected family contribution (EFC) or parent contribution. Students are packaged by Equity Item Type Groups and Related Equity Item Types. This limit can be used to assure that similar student populations are packaged equally.
event	A predefined point either in the Component Processor flow or in the program flow. As each point is encountered, the event activates each component, triggering any PeopleCode program that is associated with that component and that event. Examples of events are FieldChange, SavePreChange, and RowDelete. In PeopleSoft Enterprise Human Resources, also refers to an incident that affects benefits eligibility.
event constraints	In PeopleSoft Strategic Sourcing, these constraints are associated with a specific sourcing event. Spend is tracked within the selected event.
event propagation process	In PeopleSoft Enterprise Sales Incentive Management, a process that determines, through logic, the propagation of an original PeopleSoft Enterprise Incentive Management event and creates a derivative (duplicate) of the original event to be processed by other objects. PeopleSoft Enterprise Enterprise Sales Incentive Management uses this mechanism to implement splits, roll-ups, and so on. Event propagation determines who receives the credit.
exception	In PeopleSoft Enterprise Receivables, an item that either is a deduction or is in dispute.
exclusive pricing	In PeopleSoft Enterprise Order Management, a type of arbitration plan that is associated with a price rule. Exclusive pricing is used to price sales order transactions.
fact	In PeopleSoft Enterprise applications, facts are numeric data values from fields from a source database as well as an analytic application. A fact can be anything you want to measure your business by, for example, revenue, actual, budget data, or sales numbers. A fact is stored on a fact table.
financial aid term	In PeopleSoft Enterprise Campus Solutions, a combination of a period of time that the school determines as an instructional accounting period and an academic career. It is created and defined during the setup process. Only terms eligible for financial aid are set up for each financial aid career.
financial sanctions	For U.S. based companies and their foreign subsidiaries, a federal regulation from the Office of Foreign Assets Control (OFAC) requires that vendors be validated against a Specially Designated Nationals (SDN) list prior to payment. For PeopleSoft Payables, eSettlements, Cash Management, and Order to Cash, you can validate your vendors against any financial sanctions list (for example, the SDN list, a European Union list, and so on).
forecast item	A logical entity with a unique set of descriptive demand and forecast data that is used as the basis to forecast demand. You create forecast items for a wide range of uses, but they ultimately represent things that you buy, sell, or use in your organization and for which you require a predictable usage.

fund	In PeopleSoft Enterprise Promotions Management, a budget that can be used to fund promotional activity. There are four funding methods: top down, fixed accrual, rolling accrual, and zero-based accrual.
gap	In PeopleSoft Enterprise Campus Solutions, an artificial figure that sets aside an amount of unmet financial aid need that is not funded with Title IV funds. A gap can be used to prevent fully funding any student to conserve funds, or it can be used to preserve unmet financial aid need so that institutional funds can be awarded.
generic process type	In PeopleSoft Process Scheduler, process types are identified by a generic process type. For example, the generic process type SQR includes all SQR process types, such as SQR process and SQR report.
gift table	In PeopleSoft Enterprise Campus Solutions, a table or so-called <i>donor pyramid</i> describing the number and size of gifts that you expect will be needed to successfully complete the campaign in PeopleSoft Enterprise Contributor Relations. The gift table enables you to estimate the number of donors and prospects that you need at each gift level to reach the campaign goal.
GDS	Abbreviation for <i>Global Distribution System</i> . Broad-based term to describe all computer reservation systems for making travel plans.
GL business unit	Abbreviation for <i>general ledger business unit</i> . A unit in an organization that is an independent entity for accounting purposes. It maintains its own set of accounting books. See also <i>business unit</i> .
GL entry template	Abbreviation for <i>general ledger entry template</i> . In PeopleSoft Enterprise Campus Solutions, a template that defines how a particular item is sent to the general ledger. An item-type maps to the general ledger, and the GL entry template can involve multiple general ledger accounts. The entry to the general ledger is further controlled by high-level flags that control the summarization and the type of accounting—that is, accrual or cash.
GL Interface process	Abbreviation for <i>General Ledger Interface process</i> . In PeopleSoft Enterprise Campus Solutions, a process that is used to send transactions from PeopleSoft Enterprise Student Financials to the general ledger. Item types are mapped to specific general ledger accounts, enabling transactions to move to the general ledger when the GL Interface process is run.
global constraints	In PeopleSoft Strategic Sourcing, these constraints apply across multiple Strategic Sourcing business units. Spend is tracked across all of the events from the multiple Strategic Sourcing business units.
group	In PeopleSoft Enterprise Billing and Receivables, a posting entity that comprises one or more transactions (items, deposits, payments, transfers, matches, or write-offs). In PeopleSoft Enterprise Human Resources Management and Supply Chain Management, any set of records that are associated under a single name or variable to run calculations in PeopleSoft business processes. In PeopleSoft Enterprise Time and Labor, for example, employees are placed in groups for time reporting purposes.
ideal response	In PeopleSoft Strategic Sourcing, a question that requires the response to match the ideal value for the bid to be considered eligible for award. If the response does not match the ideal value, you can still submit the bid, but it will be disqualified and ineligible for award.
incentive object	In PeopleSoft Enterprise Incentive Management, the incentive-related objects that define and support the PeopleSoft Enterprise Incentive Management calculation process and results, such as plan templates, plans, results data, and user interaction objects.

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

keyword	In PeopleSoft Enterprise Campus Solutions, a term that you link to particular elements within PeopleSoft Enterprise Student Financials, Financial Aid, and Contributor Relations. You can use keywords as search criteria that enable you to locate specific records in a search dialog box.
KPI	An abbreviation for <i>key performance indicator</i> . A high-level measurement of how well an organization is doing in achieving critical success factors. This defines the data value or calculation upon which an assessment is determined.
KVI	Abbreviation for <i>Known Value Item</i> . Term used for products or groups of products where the selling price cannot be reduced or increased.
landlord	In PeopleSoft Real Estate Management, an entity that owns real estate and leases the real estate to tenants.
LDIF file	Abbreviation for <i>Lightweight Directory Access Protocol (LDAP) Data Interchange Format file</i> . Contains discrepancies between PeopleSoft Enterprise data and directory data.
learner group	In PeopleSoft Enterprise Learning Management, a group of learners who are linked to the same learning environment. Members of the learner group can share the same attributes, such as the same department or job code. Learner groups are used to control access to and enrollment in learning activities and programs. They are also used to perform group enrollments and mass enrollments in the back office.
learning components	In PeopleSoft Enterprise Learning Management, the foundational building blocks of learning activities. PeopleSoft Enterprise Learning Management supports six basic types of learning components: web-based, session, webcast, test, survey, and assignment. One or more of these learning component types compose a single learning activity.
learning environment	In PeopleSoft Enterprise Learning Management, identifies a set of categories and catalog items that can be made available to learner groups. Also defines the default values that are assigned to the learning activities and programs that are created within a particular learning environment. Learning environments provide a way to partition the catalog so that learners see only those items that are relevant to them.
learning history	In PeopleSoft Enterprise Learning Management, a self-service repository for all of a learner's completed learning activities and programs.
lease	In PeopleSoft Real Estate Management, a legally binding agreement between a landlord and a tenant, where the tenant rents all or part of a physical property from the landlord.
lease abstract	In PeopleSoft Real Estate Management, a summarized version of the complete lease contract with only the important terms. The lease abstract usually fits on one page and does not include legal terminology.
ledger mapping	You use ledger mapping to relate expense data from general ledger accounts to resource objects. Multiple ledger line items can be mapped to one or more resource IDs. You can also use ledger mapping to map dollar amounts (referred to as <i>rates</i>) to business units. You can map the amounts in two different ways: an actual amount that represents actual costs of the accounting period, or a budgeted amount that can be used to calculate the capacity rates as well as budgeted model results. In PeopleSoft Enterprise Warehouse, you can map general ledger accounts to the EW Ledger table.
library section	In PeopleSoft Enterprise Incentive Management, a section that is defined in a plan (or template) and that is available for other plans to share. Changes to a library section are reflected in all plans that use it.
line	In PeopleSoft Strategic Sourcing, an individual item or service upon which there can be a bid.

linked section	In PeopleSoft Enterprise Incentive Management, a section that is defined in a plan template but appears in a plan. Changes to linked sections propagate to plans using that section.
linked variable	In PeopleSoft Enterprise Incentive Management, a variable that is defined and maintained in a plan template and that also appears in a plan. Changes to linked variables propagate to plans using that variable.
LMS	Abbreviation for <i>learning management system</i> . In PeopleSoft Enterprise Campus Solutions, LMS is a PeopleSoft Enterprise Student Records feature that provides a common set of interoperability standards that enable the sharing of instructional content and data between learning and administrative environments.
load	In PeopleSoft Enterprise Inventory, identifies a group of goods that are shipped together. Load management is a feature of PeopleSoft Enterprise Inventory that is used to track the weight, the volume, and the destination of a shipment.
local functionality	In PeopleSoft Enterprise HRMS, the set of information that is available for a specific country. You can access this information when you click the appropriate country flag in the global window, or when you access it by a local country menu.
location	Locations enable you to indicate the different types of addresses—for a company, for example, one address to receive bills, another for shipping, a third for postal deliveries, and a separate street address. Each address has a different location number. The primary location—indicated by a <i>1</i> —is the address you use most often and may be different from the main address.
logistical task	In PeopleSoft Enterprise Services Procurement, an administrative task that is related to hiring a service provider. Logistical tasks are linked to the service type on the work order so that different types of services can have different logistical tasks. Logistical tasks include both preapproval tasks (such as assigning a new badge or ordering a new laptop) and postapproval tasks (such as scheduling orientation or setting up the service provider email). The logistical tasks can be mandatory or optional. Mandatory preapproval tasks must be completed before the work order is approved. Mandatory postapproval tasks, on the other hand, must be completed before a work order is released to a service provider.
market template	In PeopleSoft Enterprise Incentive Management, additional functionality that is specific to a given market or industry and is built on top of a product category.
mass change	In PeopleSoft Enterprise Campus Solutions, mass change is a SQL generator that can be used to create specialized functionality. Using mass change, you can set up a series of Insert, Update, or Delete SQL statements to perform business functions that are specific to the institution. See also <i>3C engine</i> .
match group	In PeopleSoft Enterprise Receivables, a group of receivables items and matching offset items. The system creates match groups by using user-defined matching criteria for selected field values.
MCF server	Abbreviation for <i>PeopleSoft MultiChannel Framework server</i> . Comprises the universal queue server and the MCF log server. Both processes are started when <i>MCF Servers</i> is selected in an application server domain configuration.
merchandising activity	In PeopleSoft Enterprise Promotions Management, a specific discount type that is associated with a trade promotion (such as off-invoice, billback or rebate, or lump-sum payment) that defines the performance that is required to receive the discount. In the industry, you may know this as an offer, a discount, a merchandising event, an event, or a tactic.

meta-SQL	Meta-SQL constructs expand into platform-specific SQL substrings. They are used in functions that pass SQL strings, such as in SQL objects, the SQLExec function, and PeopleSoft Application Engine programs.
metastring	Metastrings are special expressions included in SQL string literals. The metastrings, prefixed with a percent (%) symbol, are included directly in the string literals. They expand at run time into an appropriate substring for the current database platform.
multibook	In PeopleSoft Enterprise General Ledger, multiple ledgers having multiple-base currencies that are defined for a business unit, with the option to post a single transaction to all base currencies (all ledgers) or to only one of those base currencies (ledgers).
multicurrency	The ability to process transactions in a currency other than the business unit's base currency.
national allowance	In PeopleSoft Enterprise Promotions Management, a promotion at the corporate level that is funded by nondiscretionary dollars. In the industry, you may know this as a national promotion, a corporate promotion, or a corporate discount.
NDP	Abbreviation for <i>Non-Discountable Products</i> . Term used for products or groups of products where the selling price cannot be decreased.
need	In PeopleSoft Enterprise Campus Solutions, the difference between the cost of attendance (COA) and the expected family contribution (EFC). It is the gap between the cost of attending the school and the student's resources. The financial aid package is based on the amount of financial need. The process of determining a student's need is called <i>need analysis</i> .
node-oriented tree	A tree that is based on a detail structure, but the detail values are not used.
Optimization Engine	A PeopleTools component that Strategic Sourcing leverages to evaluate bids and determine an ideal award allocation. The award recommendation is based on maximizing the value while adhering to purchasing and company objectives and constraints.
pagelet	Each block of content on the home page is called a pagelet. These pagelets display summary information within a small rectangular area on the page. The pagelet provide users with a snapshot of their most relevant PeopleSoft Enterprise and non-PeopleSoft Enterprise content.
participant	In PeopleSoft Enterprise Incentive Management, participants are recipients of the incentive compensation calculation process.
participant object	Each participant object may be related to one or more compensation objects. See also <i>compensation object</i> .
partner	A company that supplies products or services that are resold or purchased by the enterprise.
pay cycle	In PeopleSoft Enterprise Payables, a set of rules that define the criteria by which it should select scheduled payments for payment creation.
payment shuffle	In PeopleSoft Enterprise Campus Solutions, a process allowing payments that have been previously posted to a student's account to be automatically reapplied when a higher priority payment is posted or the payment allocation definition is changed.
pending item	In PeopleSoft Enterprise Receivables, an individual receivable (such as an invoice, a credit memo, or a write-off) that has been entered in or created by the system, but hasn't been posted.

PeopleCode	PeopleCode is a proprietary language, executed by the PeopleSoft Enterprise component processor. PeopleCode generates results based on existing data or user actions. By using various tools provided with PeopleTools, external services are available to all PeopleSoft Enterprise applications wherever PeopleCode can be executed.
PeopleCode event	See <i>event</i> .
PeopleSoft Pure Internet Architecture	The fundamental architecture on which PeopleSoft 8 applications are constructed, consisting of a relational database management system (RDBMS), an application server, a web server, and a browser.
performance measurement	In PeopleSoft Enterprise Incentive Management, a variable used to store data (similar to an aggregator, but without a predefined formula) within the scope of an incentive plan. Performance measures are associated with a plan calendar, territory, and participant. Performance measurements are used for quota calculation and reporting.
period context	In PeopleSoft Enterprise Incentive Management, because a participant typically uses the same compensation plan for multiple periods, the period context associates a plan context with a specific calendar period and fiscal year. The period context references the associated plan context, thus forming a chain. Each plan context has a corresponding set of period contexts.
person of interest	A person about whom the organization maintains information but who is not part of the workforce.
personal portfolio	In PeopleSoft Enterprise Campus Solutions, the user-accessible menu item that contains an individual's name, address, telephone number, and other personal information.
phase	A level 1 task, meaning that if a task had subtasks, the level 1 task would be considered the phase.
pickup quantity	The product quantity that the customer is taking with them from the counter sales environment.
plan	In PeopleSoft Enterprise Sales Incentive Management, a collection of allocation rules, variables, steps, sections, and incentive rules that instruct the PeopleSoft Enterprise Incentive Management engine in how to process transactions.
plan context	In PeopleSoft Enterprise Incentive Management, correlates a participant with the compensation plan and node to which the participant is assigned, enabling the PeopleSoft Enterprise Incentive Management system to find anything that is associated with the node and that is required to perform compensation processing. Each participant, node, and plan combination represents a unique plan context—if three participants are on a compensation structure, each has a different plan context. Configuration plans are identified by plan contexts and are associated with the participants that refer to them.
plan template	In PeopleSoft Enterprise Incentive Management, the base from which a plan is created. A plan template contains common sections and variables that are inherited by all plans that are created from the template. A template may contain steps and sections that are not visible in the plan definition.
planned learning	In PeopleSoft Enterprise Learning Management, a self-service repository for all of a learner's planned learning activities and programs.
planning instance	In PeopleSoft Enterprise Supply Planning, a set of data (business units, items, supplies, and demands) constituting the inputs and outputs of a supply plan.
population	In PeopleSoft Enterprise Campus Solutions, the middle level of the three-level classification structure that you define in PeopleSoft Enterprise Recruiting and

	Admissions for enrollment management. You can define a population level, link it to other levels, and set enrollment target numbers for it. See also <i>division</i> and <i>cohort</i> .
portal registry	In PeopleSoft Enterprise applications, the portal registry is a tree-like structure in which content references are organized, classified, and registered. It is a central repository that defines both the structure and content of a portal through a hierarchical, tree-like structure of folders useful for organizing and securing content references.
predecessor task	A task that you must complete before you start another task.
price breaks	In PeopleSoft Strategic Sourcing, a price discount or surcharge that a bidder may apply based on the quantity awarded.
price components	In PeopleSoft Strategic Sourcing, the various components, such as material costs, labor costs, shipping costs, and so on that make up the overall bid price.
price list	Enables you to select products and conditions for which the price list applies to a transaction. During a transaction, the system either determines the product price based on the predefined search hierarchy for the transaction or uses the product's lowest price on any associated, active price lists. This price is used as the basis for any further discounts and surcharges.
price rule	The conditions that must be met for adjustments to be applied to the base price. Multiple rules can apply when conditions of each rule are met.
price rule conditions	Conditions that select the price-by fields, the values for the price-by fields, and the operator that determines how the price-by fields relate to the transaction.
price rule key	The fields that are available to define price rule conditions (which are used to match a transaction) on the price rule.
primacy number	In PeopleSoft Enterprise Campus Solutions, a number that the system uses to prioritize financial aid applications when students are enrolled in multiple academic careers and academic programs at the same time. The Consolidate Academic Statistics process uses the primacy number indicated for both the career and program at the institutional level to determine a student's primary career and program. The system also uses the number to determine the primary student attribute value that is used when you extract data to report on cohorts. The lowest number takes precedence.
primary name type	In PeopleSoft Enterprise Campus Solutions, the name type that is used to link the name stored at the highest level within the system to the lower-level set of names that an individual provides.
process category	In PeopleSoft Process Scheduler, processes that are grouped for server load balancing and prioritization.
process group	In PeopleSoft Enterprise Financials, a group of application processes (performed in a defined order) that users can initiate in real time, directly from a transaction entry page.
process definition	Process definitions define each run request.
process instance	A unique number that identifies each process request. This value is automatically incremented and assigned to each requested process when the process is submitted to run.
process job	You can link process definitions into a job request and process each request serially or in parallel. You can also initiate subsequent processes based on the return code from each prior request.

process request	A single run request, such as a Structured Query Report (SQR), a COBOL or Application Engine program, or a Crystal report that you run through PeopleSoft Process Scheduler.
process run control	A PeopleTools variable used to retain PeopleSoft Process Scheduler values needed at runtime for all requests that reference a run control ID. Do not confuse these with application run controls, which may be defined with the same run control ID, but only contain information specific to a given application process request.
product	A PeopleSoft Enterprise or third-party product. PeopleSoft organizes its software products into product families and product lines. Interactive Services Repository contains information about every release of every product that PeopleSoft sells, as well as products from certified third-party companies. These products appear with the product name and release number.
product adds	The pricing functionality where buying product A gets product B for free or at a price (formerly referred to as <i>giveaways</i>).
product bidding	In PeopleSoft Strategic Sourcing, the placing of a bid on behalf of the bidder, up or down to the bidder's specified amount, so that the bidder can be the leading bidder.
product category	In PeopleSoft Enterprise Incentive Management, indicates an application in the PeopleSoft Enterprise Incentive Management suite of products. Each transaction in the PeopleSoft Enterprise Incentive Management system is associated with a product category.
product family	A group of products that are related by common functionality. The family names that can be searched using Interactive Service Repository are Oracle's PeopleSoft Enterprise, PeopleSoft EnterpriseOne, PeopleSoft World, and third-party, certified partners.
product line	The name of a PeopleSoft Enterprise product line or the company name of a third-party certified partner. Integration Services Repository enables you to search for integration points by product line.
programs	In PeopleSoft Enterprise Learning Management, a high-level grouping that guides the learner along a specific learning path through sections of catalog items. PeopleSoft Enterprise Learning Systems provides two types of programs—curricula and certifications.
progress log	In PeopleSoft Enterprise Services Procurement, tracks deliverable-based projects. This is similar to the time sheet in function and process. The service provider contact uses the progress log to record and submit progress on deliverables. The progress can be logged by the activity that is performed, by the percentage of work that is completed, or by the completion of milestone activities that are defined for the project.
project transaction	In PeopleSoft Enterprise Project Costing, an individual transaction line that represents a cost, time, budget, or other transaction row.
promotion	In PeopleSoft Enterprise Promotions Management, a trade promotion, which is typically funded from trade dollars and used by consumer products manufacturers to increase sales volume.
prospects	In PeopleSoft Enterprise Campus Solutions, students who are interested in applying to the institution. In PeopleSoft Enterprise Contributor Relations, individuals and organizations that are most likely to make substantial financial commitments or other types of commitments to the institution.
proxy bidding	In PeopleSoft Strategic Sourcing, the placing of a bid on behalf of the bidder, up or down to the bidder's specified amount, so that the bidder can be the leading bidder.

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

REN server	Abbreviation for <i>real-time event notification server</i> in PeopleSoft MultiChannel Framework.
requester	In PeopleSoft Enterprise eSettlements, an individual who requests goods or services and whose ID appears on the various procurement pages that reference purchase orders.
reservations	In PeopleSoft Expenses, travel reservations that have been placed with the travel vendor.
reversal indicator	In PeopleSoft Enterprise Campus Solutions, an indicator that denotes when a particular payment has been reversed, usually because of insufficient funds.
RFI event	In PeopleSoft Strategic Sourcing, a request for information.
RFx event	In PeopleSoft Strategic Sourcing, a request for proposal or request for a quote event when bidders submit their overall best bids and during which bidders do not actively compete against one another.
role	Describes how people fit into PeopleSoft Workflow. A role is a class of users who perform the same type of work, such as clerks or managers. Your business rules typically specify what user role needs to do an activity.
role user	A PeopleSoft Workflow user. A person's role user ID serves much the same purpose as a user ID does in other parts of the system. PeopleSoft Workflow uses role user IDs to determine how to route worklist items to users (through an email address, for example) and to track the roles that users play in the workflow. Role users do not need PeopleSoft user IDs.
roll up	In a tree, to roll up is to total sums based on the information hierarchy.
run control	A run control is a type of online page that is used to begin a process, such as the batch processing of a payroll run. Run control pages generally start a program that manipulates data.
run control ID	A unique ID to associate each user with his or her own run control table entries.
run-level context	In PeopleSoft Enterprise Incentive Management, associates a particular run (and batch ID) with a period context and plan context. Every plan context that participates in a run has a separate run-level context. Because a run cannot span periods, only one run-level context is associated with each plan context.
saved bid	In PeopleSoft Strategic Sourcing, a bid that has been created but not submitted. Only submitted bids are eligible for award.
score	In PeopleSoft Strategic Sourcing, the numerical sum of answers (percentages) to bid factors on an event. Scores appear only to bidders on auction events.
SCP SCBM XML message	Abbreviation for <i>Supply Chain Planning Supply Chain Business Modeler Extensible Markup Language message</i> . Supply Chain Business Modeler uses XML as the format for all data that it imports and exports.
search query	You use this set of objects to pass a query string and operators to the search engine. The search index returns a set of matching results with keys to the source documents.
search/match	In PeopleSoft Enterprise Campus Solutions and PeopleSoft Enterprise Human Resources Management Solutions, a feature that enables you to search for and identify duplicate records in the database.
seasonal address	In PeopleSoft Enterprise Campus Solutions, an address that recurs for the same length of time at the same time of year each year until adjusted or deleted.

section	In PeopleSoft Enterprise Incentive Management, a collection of incentive rules that operate on transactions of a specific type. Sections enable plans to be segmented to process logical events in different sections.
security event	In commitment control, security events trigger security authorization checking, such as budget entries, transfers, and adjustments; exception overrides and notifications; and inquiries.
sell event	In PeopleSoft Strategic Sourcing, for event creators, the sale of goods or services most typically associated with forward auctions. For bidders, the purchase of goods or services.
serial genealogy	In PeopleSoft Enterprise Manufacturing, the ability to track the composition of a specific, serial-controlled item.
serial in production	In PeopleSoft Enterprise Manufacturing, enables the tracing of serial information for manufactured items. This is maintained in the Item Master record.
service impact	In PeopleSoft Enterprise Campus Solutions, the resulting action triggered by a service indicator. For example, a service indicator that reflects nonpayment of account balances by a student might result in a service impact that prohibits registration for classes.
service indicator	In PeopleSoft Enterprise Campus Solutions, indicates services that may be either withheld or provided to an individual. Negative service indicators indicate holds that prevent the individual from receiving specified services, such as check-cashing privileges or registration for classes. Positive service indicators designate special services that are provided to the individual, such as front-of-line service or special services for disabled students.
session	In PeopleSoft Enterprise Campus Solutions, time elements that subdivide a term into multiple time periods during which classes are offered. In PeopleSoft Enterprise Contributor Relations, a session is the means of validating gift, pledge, membership, or adjustment data entry . It controls access to the data entered by a specific user ID. Sessions are balanced, queued, and then posted to the institution's financial system. Sessions must be posted to enter a matching gift or pledge payment, to make an adjustment, or to process giving clubs or acknowledgements. In PeopleSoft Enterprise Learning Management, a single meeting day of an activity (that is, the period of time between start and finish times within a day). The session stores the specific date, location, meeting time, and instructor. Sessions are used for scheduled training.
session template	In PeopleSoft Enterprise Learning Management, enables you to set up common activity characteristics that may be reused while scheduling a PeopleSoft Enterprise Learning Management activity—characteristics such as days of the week, start and end times, facility and room assignments, instructors, and equipment. A session pattern template can be attached to an activity that is being scheduled. Attaching a template to an activity causes all of the default template information to populate the activity session pattern.
setup relationship	In PeopleSoft Enterprise Incentive Management, a relationship object type that associates a configuration plan with any structure node.
share driver expression	In PeopleSoft Enterprise Business Planning, a named planning method similar to a driver expression, but which you can set up globally for shared use within a single planning application or to be shared between multiple planning applications through PeopleSoft Enterprise Warehouse.
short-term customer	A customer not in the system who is entered during sales order entry using a template.

single signon	With single signon, users can, after being authenticated by a PeopleSoft Enterprise application server, access a second PeopleSoft Enterprise application server without entering a user ID or password.
source key process	In PeopleSoft Enterprise Campus Solutions, a process that relates a particular transaction to the source of the charge or financial aid. On selected pages, you can drill down into particular charges.
source transaction	In commitment control, any transaction generated in a PeopleSoft Enterprise or third-party application that is integrated with commitment control and which can be checked against commitment control budgets. For example, a pre-encumbrance, encumbrance, expenditure, recognized revenue, or collected revenue transaction.
sourcing objective	For constraints, the option to designate whether a business rule is required (mandatory) or is only recommended (target).
speed key	See <i>communication key</i> .
SpeedChart	A user-defined shorthand key that designates several ChartKeys to be used for voucher entry. Percentages can optionally be related to each ChartKey in a SpeedChart definition.
SpeedType	A code representing a combination of ChartField values. SpeedTypes simplify the entry of ChartFields commonly used together.
staging	A method of consolidating selected partner offerings with the offerings from the enterprise's other partners.
standard letter code	In PeopleSoft Enterprise Campus Solutions, a standard letter code used to identify each letter template available for use in mail merge functions. Every letter generated in the system must have a standard letter code identification.
statutory account	Account required by a regulatory authority for recording and reporting financial results. In PeopleSoft Enterprise, this is equivalent to the Alternate Account (ALTACCT) ChartField.
step	In PeopleSoft Enterprise Sales Incentive Management, a collection of sections in a plan. Each step corresponds to a step in the job run.
storage level	In PeopleSoft Enterprise Inventory, identifies the level of a material storage location. Material storage locations are made up of a business unit, a storage area, and a storage level. You can set up to four storage levels.
subcustomer qualifier	A value that groups customers into a division for which you can generate detailed history, aging, events, and profiles.
Summary ChartField	You use summary ChartFields to create summary ledgers that roll up detail amounts based on specific detail values or on selected tree nodes. When detail values are summarized using tree nodes, summary ChartFields must be used in the summary ledger data record to accommodate the maximum length of a node name (20 characters).
summary ledger	An accounting feature used primarily in allocations, inquiries, and PS/nVision reporting to store combined account balances from detail ledgers. Summary ledgers increase speed and efficiency of reporting by eliminating the need to summarize detail ledger balances each time a report is requested. Instead, detail balances are summarized in a background process according to user-specified criteria and stored on summary ledgers. The summary ledgers are then accessed directly for reporting.
summary time period	In PeopleSoft Enterprise Business Planning, any time period (other than a base time period) that is an aggregate of other time periods, including other summary time periods and base time periods, such as quarter and year total.

summary tree	A tree used to roll up accounts for each type of report in summary ledgers. Summary trees enable you to define trees on trees. In a summary tree, the detail values are really nodes on a detail tree or another summary tree (known as the <i>basis</i> tree). A summary tree structure specifies the details on which the summary trees are to be built.
syndicate	To distribute a production version of the enterprise catalog to partners.
system function	In PeopleSoft Enterprise Receivables, an activity that defines how the system generates accounting entries for the general ledger.
system source	The system source identifies the source of a transaction row in the database. For example, a transaction that originates in PeopleSoft Enterprise Expenses contains a system source code of BEX (Expenses Batch). When PeopleSoft Enterprise Project Costing prices the source transaction row for billing, the system creates a new row with a system source code of PRP (Project Costing pricing), which represents the system source of the new row. System source codes can identify sources that are internal or external to the PeopleSoft Enterprise system. For example, processes that import data from Microsoft Project into PeopleSoft Enterprise applications create transaction rows with a source code of MSP (Microsoft Project).
TableSet	A means of sharing similar sets of values in control tables, where the actual data values are different but the structure of the tables is the same.
TableSet sharing	Shared data that is stored in many tables that are based on the same TableSets. Tables that use TableSet sharing contain the SETID field as an additional key or unique identifier.
target currency	The value of the entry currency or currencies converted to a single currency for budget viewing and inquiry purposes.
task	A deliverable item on the detailed sourcing plan.
tax authority	In PeopleSoft Enterprise Campus Solutions, a user-defined element that combines a description and percentage of a tax with an account type, an item type, and a service impact.
template	A template is HTML code associated with a web page. It defines the layout of the page and also where to get HTML for each part of the page. In PeopleSoft Enterprise, you use templates to build a page by combining HTML from a number of sources. For a PeopleSoft Enterprise portal, all templates must be registered in the portal registry, and each content reference must be assigned a template.
tenant	In PeopleSoft Real Estate Management, an entity that leases real estate from a landlord.
territory	In PeopleSoft Enterprise Sales Incentive Management, hierarchical relationships of business objects, including regions, products, customers, industries, and participants.
third party	A company or vendor that has extensive PeopleSoft Enterprise product knowledge and whose products and integrations have been certified and are compatible with PeopleSoft Enterprise applications.
tiered pricing	Enables different portions of a schedule to be priced differently from one another.
time span	A relative period, such as year-to-date or current period, that various PeopleSoft General Ledger functions and reports can use when a rolling time frame, rather than a specific date, is required.
total cost	In PeopleSoft Strategic Sourcing, the estimated dollar cost (sum of real price dollars and potential “soft” or non-price dollars) of a particular award approach.

travel group	In PeopleSoft Expenses, the organization's travel rules and policies that are associated with specific business units, departments, or employees. You must define at least one travel group when setting up the PeopleSoft Expenses travel feature. You must define and associate at least one travel group with a travel vendor.
travel partner	In PeopleSoft Expenses, the travel vendor with which the organization has a contractual relationship.
3C engine	Abbreviation for <i>Communications, Checklists, and Comments engine</i> . In PeopleSoft Enterprise Campus Solutions, the 3C engine enables you to automate business processes that involve additions, deletions, and updates to communications, checklists, and comments. You define events and triggers to engage the engine, which runs the mass change and processes the 3C records (for individuals or organizations) immediately and automatically from within business processes.
3C group	Abbreviation for <i>Communications, Checklists, and Comments group</i> . In PeopleSoft Enterprise Campus Solutions, a method of assigning or restricting access privileges. A 3C group enables you to group specific communication categories, checklist codes, and comment categories. You can then assign the group inquiry-only access or update access, as appropriate.
trace usage	In PeopleSoft Enterprise Manufacturing, enables the control of which components will be traced during the manufacturing process. Serial- and lot-controlled components can be traced. This is maintained in the Item Master record.
transaction allocation	In PeopleSoft Enterprise Incentive Management, the process of identifying the owner of a transaction. When a raw transaction from a batch is allocated to a plan context, the transaction is duplicated in the PeopleSoft Enterprise Incentive Management transaction tables.
transaction state	In PeopleSoft Enterprise Incentive Management, a value assigned by an incentive rule to a transaction. Transaction states enable sections to process only transactions that are at a specific stage in system processing. After being successfully processed, transactions may be promoted to the next transaction state and "picked up" by a different section for further processing.
Translate table	A system edit table that stores codes and translate values for the miscellaneous fields in the database that do not warrant individual edit tables of their own.
tree	The graphical hierarchy in PeopleSoft Enterprise systems that displays the relationship between all accounting units (for example, corporate divisions, projects, reporting groups, account numbers) and determines roll-up hierarchies.
tuition lock	In PeopleSoft Enterprise Campus Solutions, a feature in the Tuition Calculation process that enables you to specify a point in a term after which students are charged a minimum (or <i>locked</i>) fee amount. Students are charged the locked fee amount even if they later drop classes and take less than the normal load level for that tuition charge.
unclaimed transaction	In PeopleSoft Enterprise Incentive Management, a transaction that is not claimed by a node or participant after the allocation process has completed, usually due to missing or incomplete data. Unclaimed transactions may be manually assigned to the appropriate node or participant by a compensation administrator.
universal navigation header	Every PeopleSoft Enterprise portal includes the universal navigation header, intended to appear at the top of every page as long as the user is signed on to the portal. In addition to providing access to the standard navigation buttons (like Home, Favorites, and signoff) the universal navigation header can also display a welcome message for each user.
update access	In PeopleSoft Enterprise Campus Solutions, a type of security access that permits the user to edit and update data.

See also *inquiry access*.

user interaction object	In PeopleSoft Enterprise Sales Incentive Management, used to define the reporting components and reports that a participant can access in his or her context. All PeopleSoft Enterprise Sales Incentive Management user interface objects and reports are registered as user interaction objects. User interaction objects can be linked to a compensation structure node through a compensation relationship object (individually or as groups).
variable	In PeopleSoft Enterprise Sales Incentive Management, the intermediate results of calculations. Variables hold the calculation results and are then inputs to other calculations. Variables can be plan variables that persist beyond the run of an engine or local variables that exist only during the processing of a section.
VAT exception	Abbreviation for <i>value-added tax exception</i> . A temporary or permanent exemption from paying VAT that is granted to an organization. This term refers to both VAT exoneration and VAT suspension.
VAT exempt	Abbreviation for <i>value-added tax exempt</i> . Describes goods and services that are not subject to VAT. Organizations that supply exempt goods or services are unable to recover the related input VAT. This is also referred to as exempt without recovery.
VAT exoneration	Abbreviation for <i>value-added tax exoneration</i> . An organization that has been granted a permanent exemption from paying VAT due to the nature of that organization.
VAT suspension	Abbreviation for <i>value-added tax suspension</i> . An organization that has been granted a temporary exemption from paying VAT.
warehouse	A PeopleSoft Enterprise data warehouse that consists of predefined ETL maps, data warehouse tools, and DataMart definitions.
weight or weighting	In PeopleSoft Strategic Sourcing, how important the line or question is to the overall event. Weighting is used to score and analyze bids. For RFx and RFI events, weightings may or may not appear to bidders.
work order	In PeopleSoft Enterprise Services Procurement, enables an enterprise to create resource-based and deliverable-based transactions that specify the basic terms and conditions for hiring a specific service provider. When a service provider is hired, the service provider logs time or progress against the work order.
worker	A person who is part of the workforce; an employee or a contingent worker.
workset	A group of people and organizations that are linked together as a set. You can use worksets to simultaneously retrieve the data for a group of people and organizations and work with the information on a single page.
worksheet	A way of presenting data through a PeopleSoft Enterprise Business Analysis Modeler interface that enables users to do in-depth analysis using pivoting tables, charts, notes, and history information.
worklist	The automated to-do list that PeopleSoft Workflow creates. From the worklist, you can directly access the pages you need to perform the next action, and then return to the worklist for another item.
XML link	The XML Linking language enables you to insert elements into XML documents to create a links between resources.
XML schema	An XML definition that standardizes the representation of application messages, component interfaces, or business interlinks.
XPI	Abbreviation for <i>eXtended Process Integrator</i> . PeopleSoft XPI is the integration infrastructure that enables both real-time and batch communication with JD Edwards EnterpriseOne applications.

yield by operation

In PeopleSoft Enterprise Manufacturing, the ability to plan the loss of a manufactured item on an operation-by-operation basis.

zero-rated VAT

Abbreviation for *zero-rated value-added tax*. A VAT transaction with a VAT code that has a tax percent of zero. Used to track taxable VAT activity where no actual VAT amount is charged. Organizations that supply zero-rated goods and services can still recover the related input VAT. This is also referred to as exempt with recovery.

Index

A

- A4 paper, printing 124
- additional documentation xvi
- Adobe Reader 139
- Agfa Monotype 56
- ALTER-PRINTER command 135
- Alternate Character Architecture 62
- alternate language DLLs
 - delivered 237, 240
 - locating language resources 239
 - translating resource files 240
 - understanding 238
- American National Standards Institute (ANSI), *See* ANSI character sets
- ANSI character sets
 - forcing binary sorts in SQL 115
 - non-Unicode vs Unicode data, identifying 83
 - understanding 46, 49
- Append_Report_Translation function 128
- Application Data page 272, 273
- Application Designer
 - defining the text search scope 211
 - setting fonts 56
 - translating definitions in projects 246
 - translating folder tabs 257
 - Translation Designer 245
 - See Also* Translation Designer
 - understanding field length semantics 58
- Application Engine definitions, translating 271
- application fundamentals xv
- application servers
 - locating language resources 239
 - selecting character sets 55
 - supporting email character sets 57
 - using MAKEUNIX.BAT 243
- Arabic
 - character set support 143
 - SQR support 139
- ASCII
 - native2ascii utility 235
 - saving properties files 235
 - sorting data 111

- understanding 40
- UTF-8 encoding 44
- auditing report, base language swap 178

B

- Baltic character set 144
- base languages
 - designing base-language pages 162
 - setting in Configuration Manager 18
 - swapping 177, 180
 - swapping, checking space/resources for 180
 - swapping, prerequisites for 178
 - swapping, running the audit report for 178
 - understanding 167
 - understanding base language tables 167
- base records
 - fixing invalid related language records 178
 - fixing key fields 179
- base tables
 - cloning 170
 - fixing orphaned rows 179
 - maintaining 189
 - related language table and base table – one-to-one view 183
 - related language table and base tables – one-to-two view 184
 - related language tables and base tables – two-to-two view 185
 - related language tables, associating with 171
 - related language tables, understanding 170
 - storing language-sensitive descriptions 171
 - understanding 168
- base time zones
 - %BaseTimeZone system variable 38
 - changing 29
 - setting 28
 - understanding 27
- %BaseTimeZone system variable 36, 38
- Basic Multilingual Plane (BMP) 43

- bi-directional text
 - Arabic character set support 143
 - Hebrew character set support 147
 - printing in SQR for PeopleSoft 139
 - SQR support 139
- bidirectional, *See* bidirectional language support
- bidirectional language support 195
 - customizing PIA pages for 198
 - features 195
 - Hijri (Umm al-Qura), enabling 198
 - platforms supported 196
 - translations provided 195
 - understanding 195
- bidirectional pages, *See* PeopleSoft Pure Internet Architecture
- binary sorting
 - forcing binary sorts in SQL 115
 - understanding 111
 - using 114
- %BINARYSORT meta-SQL
 - binary sorting 114
 - forcing binary sorts in SQL 115
 - sorting in COBOL 71, 116
- BIND/SELECT log 81
- Bitstream 56
- BMP 43
- browsers
 - managing character sets/fonts 55
 - setting language preferences 15
 - setting up fonts in SQR for PeopleSoft 131
- business interlinks, translating 260

C

- cbl2uni.bat command 73
- Celtic character set 144
- character sets
 - ANSI 46
 - See Also* ANSI character sets
 - Arabic 143
 - See Also* Arabic
 - Baltic 144
 - Celtic 144
 - Chinese 144
 - See Also* Chinese
 - COBOL, encodings for 69
 - COBOL, supporting for 48
 - Cyrillic 145
 - databases, selecting for 54

- databases, supporting in 48
- DBCS 46
 - See Also* DBCS
- Greek 146
- Gurmukhi 147
- Hebrew 147
 - See Also* Hebrew
- Icelandic 147
- Japanese 62
 - See Also* Japanese
- Korean 149
 - See Also* Korean
- Latin 149
- Malayalam 152
- managing in PeopleSoft Pure Internet Architecture 47, 55
- non-Unicode databases, checking field length for 60
- non-Unicode, converting from Unicode to 48
- non-Unicode, determining appropriate 228
- non-Unicode, understanding 46
 - See Also* ANSI character sets
- Nordic 152
- performing file operations 48
- SBCS 46
 - See Also* SBCS
- selecting for application servers 55
- selecting for CSV queries 232
- selecting for SQR messages 130
- selecting for the Verity search engine 232
- selecting Linux default 55
- selecting Unix default 55
- setting default 231
- setting for files being opened 141
- setting up files for PostScript printing 137
- setting/supporting for email 57
- Shift-JIS 42
 - See Also* Shift-JIS
- Slavic 152
- supporting for third-party products 48
- supporting in SQR for PeopleSoft 142
- Symbol 152
- Thai 153
- Turkish 153
- understanding 39, 40
- understanding encodings 143

- understanding field length
 - semantics 58
- understanding keyboard layouts 57
- understanding types 49
- Unicode 42
 - See Also* Unicode
- Vietnamese 154
- character strings
 - comparing in COBOL 116
 - detecting/converting characters 62
 - identifying ANSI vs Unicode data 83
 - measuring length 60, 140
 - parsing Unicode 71, 85
 - requesting maps for 88
 - shifting DBCSs for character sets 47
- characters
 - character sets 39
 - See Also* character sets
 - sorting 111
 - See Also* sorting
 - strings 140
 - See Also* character strings
 - using digit group separators 24
- Chinese
 - printing in PDF format 133, 134
 - printing with PostScript fonts 138
 - setting up fonts 132
 - sorting 112
 - standard name formats 62
 - supporting character set encodings 144
 - transforming keyboard strokes into ideographs 57
- %clienttimezone system variable 38
- %ClientTimeZone system variable 36
- COBOL 95
 - See Also* PeopleSoft COBOL on z/OS
 - binary sorting 114
 - converting programs to run on
 - Unicode 72
 - See Also* COBOL Conversion utility
 - identifying ANSI vs Unicode data 83
 - parsing Unicode strings 71, 85
 - PTPSTRFN function 85
 - See Also* PTPSTRFN function
 - setting column lengths in dynamic
 - SQL 83
 - sorting 71, 116
 - storage requirements for Unicode 70
 - supporting character sets 48
 - Unicode encodings in 69
 - Unicode-specific error messages 71
- COBOL Conversion utility
 - expanding fields 74
 - fine-tuning programs 82
 - handling arrays comprising a single character 78, 85
 - handling date/time fields 77
 - handling redefined character fields 77
 - handling SQL buffer setup data 74, 83
 - identifying converted programs 74
 - running 73
 - understanding 72
 - understanding error logs 81
 - using directives 79
- COCCUR directive 79, 80
- codepages, *See* character sets
- codepoints 39
- Cognos PowerPlay, *See* PowerPlay
- collation, *See* sorting
- comma-separated value (CSV) format, *See* CSV format
- commands
 - ALTER-PRINTER 135
 - cbl2uni.bat 73
 - COBOL Conversion utility 73
 - DEFINE-PRINTER 135
 - MAKEALTL 242
 - pscbl.mak 73
 - SQR Open 141
 - SQR OPEN 130, 141
 - SWAP_BASE_LANGUAGE 177, 180
- comments, submitting xx
- common elements xx
- component interfaces messages, translating 259
- Configuration Manager
 - adding languages 232
 - setting fonts 56
 - setting languages 18
- contact information xx
- ConvertDatetimeToBase function 38
- ConvertTimeToBase function 38
- CP1250 137
- cross-references xix
- CSV format
 - downloading query results 230
 - report output 130, 139
 - selecting character sets for queries 232
 - understanding 143
- Cube Manager

- understanding cubes 120
- using Essbase cubes 121
- using PowerPlay cubes 120
- currencies
 - currency fields 63
 - See Also* currency fields
 - setting precision 125
 - setting the display for PeopleTools 24
- Currency Code table 63
- currency fields
 - connecting to the system-wide multicurrency check box 66
 - enabling system-wide settings 66
 - hiding 65
 - setting up 64
 - setting up display properties 64
 - sizing 67
 - understanding formatting 63
 - understanding multicurrency settings 65
- %CurrentDateTime meta-SQL 27
- Customer Connection website xvi
- Cyrillic character set 145

D

- Data Mover
 - adding translations to databases 174
 - listing translatable tables 211
 - making tables searchable 223
 - swapping base languages 177, 180
- databases
 - adding translations 174
 - checking field length for non-Unicode 60
 - selecting character sets 54
 - supporting character sets 48
 - swapping base languages 177, 178
 - understanding field length semantics 58
 - upgrading customizations to multilingual 176
- dates
 - datetime field time zones 35
 - exception date log 82
 - formatting in SQRs 125
 - handling date/datetime fields in the COBOL Conversion utility 77
- DATETIME.SQC program 125
- Daylight Savings page 32
- daylight savings time

- setting for time zone definitions 31
- understanding 27
- viewing start/end dates 32
- DB2 UDB for Linux, Unix, and Windows (Unicode support) 44
- DB2 UDB for OS/390 and z/OS
 - creating related language tables 181
 - shifting DBCSs for character sets 47
 - sorting in COBOL 71
 - understanding field length semantics 60
- DBCS
 - checking field length for non-Unicode databases 60
 - checking field lengths on Japanese databases 61
 - understanding 46
- decimal precision
 - setting for currencies 63, 125
 - understanding multicurrency settings 65
 - using the International Field Size utility 67
- Define Personalizations page 24
- DEFINE-PRINTER command 135
- definitions
 - accessing 251
 - cloning search 213
 - finding references 251
 - language 225
 - opening 251
 - string 122
 - See Also* string definitions
 - translating 245
 - translating ACE analytic model ID definitions 277
 - translating Application Engine 271
 - translating business process 259
 - translating HTML 274
 - translating in order 252
 - translating menu 255
 - translating page 253
 - See Also* page definitions
 - translating PeopleSoft Integration Broker definitions 275
 - translating Process Scheduler 260
 - translating record 268
 - translating tree 266
 - using translation pages 252
 - viewing in Translation Designer 247

- viewing time zone 30
- documentation
 - printed xvi
 - related xvi
 - updates xvi
- double-byte character sets (DBCS), *See* DBCS

E

- email character sets 43, 57
- environment variables
 - INCLUDE 241
 - running MAKEALTL.BAT 241
 - running MAKEUNIX.BAT 243
 - TOOLBIN 241
- errors
 - adding error property files for languages 234
 - BIND/SELECT log 81
 - COBOL conversion logs 81
 - decimal precision 65
 - exception date log 82
 - exception log 81
 - running COBOL Unicode 71
 - running the audit report for base language swaps 178
 - swapping base languages 180
- Essbase
 - Unicode support 48
 - using multiple-language cubes 121
- EUDCEDIT program 134
- Excel
 - copying translations from Translation Designer 251
 - international versions 159
 - SQR for PeopleSoft output 143
 - translating PS/nVision layout to 154
- exception date log 82
- exception log 81
- EXPAND directive 79
- EXPEOF directive 79, 80

F

- fields
 - currency 63
 - See Also* currency fields
 - numeric 21
 - See Also* numeric fields
 - page 36

- See Also* page fields
- record 162
 - See Also* record fields
- translating definitions 247
- translating labels 161
- using the Get_Field_Information procedure 127

- flat files 143

- folder tabs, translating 257, 258

- fonts

- foundries 56
- managing in PeopleSoft Pure Internet Architecture 55
- obtaining 56
- PDF 133
 - See Also* PDF fonts
- printing limitations in SQR for PeopleSoft 139
- public domain 56
- setting in Configuration Manager 56
- setting in PeopleCode Editor 56
- setting up files for PostScript printing 137
- setting up in SQR for PeopleSoft 131

- Format_Currency_Amt function 125

- Format_Currency_Amt_Numeric function 125

- functions

- Append_Report_Translation 128
- ConvertDatetimeToBase 38
- ConvertTimeToBase 38
- Format_Currency_Amt 125
- Format_Currency_Amt_Numeric 125
- Get_Field_Information 127
- Init_Report_Translation 126
- IsDaylightSavings 38
- Length() 141
- Lengthp() 141
- Lengtht() 141
- SetLanguage 13
- string 62
- Substr() 141
- Substrp() 141
- Substrt() 141

G

- General Profile Information page 14

- Get_Field_Information function 127

- Global Multi-Language CD-ROM 173

- Global Technology

- implementing 4
- understanding xxiii, 3
- globalization
 - designing global-ready pages 162
 - developing global applications 161
 - displaying global currency fields 65
 - Global Technology xxiii
 - See Also* Global Technology
 - PS/nVision design-time globalization features 155
 - SQR for PeopleSoft 124
 - See Also* SQR for PeopleSoft
- glossary 291
- GMT 27
- Greek character set 146
- Greenwich mean time (GMT) 27
- Gurmukhi character set 147

H

- Hebrew
 - character set support 147
 - SQR support 139
- Help documentation 15
- Hijri (Umm al-Qura), enabling 198
- HTML
 - pages 55
 - See Also* HTML pages
 - report output 130, 139, 143
 - translating definitions 274
- HTML pages
 - character set support 44
 - managing character sets/fonts 55
- Hyperion Essbase, *See* Essbase

I

- Icelandic character set 147
- images
 - adding to languages 235
 - translating 244
- IME 55, 57
- INCLUDE environment variable 241
- %include% environment variable 242
- Init_Report_Translation function 126
- input method editor (IME) 55, 57
- Inquiry pages 222
- international
 - handling international characters 61
 - preferences 11
 - text 129

- See Also* international text
- International Field Size utility 67
- International Field Sizes table 67
- International Organization for Standardization (ISO), *See* ISO
- International Preferences page 13
- international text
 - printing in SQRs 131
 - processing 129
 - setting up files for PCL/line printing 135
 - setting up files for PostScript printing 137
 - using in SQR for PeopleSoft 139
- Internet Explorer
 - preferred languages 15
- IsDaylightSavings function 38
- ISO
 - character sets 39
 - locales 15
 - See Also* ISO locales

- ISO locales
 - determining language codes 226
 - selecting for the PSLANGUAGES table 231
 - understanding language codes 15

J

- Japanese
 - character sets for outgoing email 57
 - checking field lengths 61
 - detecting/converting characters 62
 - formatting report dates/times 125
 - non-Unicode DBCS settings 61
 - printing in PDF format 133, 134
 - printing with PostScript fonts 138
 - setting up fonts 132
 - setting up PCL printing 135
 - setting up user-defined fonts 134
 - Shift-JIS character set 42
 - See Also* Shift-JIS
 - sorting 112
 - standard name formats 62
 - supporting character set encodings 148
 - transforming keyboard strokes into ideographs 57
- Java
 - native2ascii utility 235
 - See Also* native2ascii utility
 - translating properties files 243

K

- key fields
 - designing related language records 181
 - fixing errors for base language swaps 178
- keyboard layouts, foreign language 55, 57
- Korean
 - printing in PDF format 133, 134
 - printing with PostScript fonts 138
 - setting up fonts 132
 - sorting 112
 - standard name formats 62
 - supporting character set encodings 149
 - transforming keyboard strokes into ideographs 57

L

- labels
 - designing for global-ready pages 162
 - labelling time zones 30
 - retrieving in PS/nVision 158
 - translating 161
 - translating for mobile page views 262
 - translating PS/nVision layout to Excel 154
 - using the Strings table 122
 - See Also* Strings table
- %Language system variable 13
- %Language_Base system variable 13
- LANGUAGE_CD field 229
- languages
 - adding error/text property files 234
 - adding for signon 233
 - adding images 235
 - adding to Configuration Manager 232
 - adding to the PSLANGUAGES table 230
 - base 162
 - See Also* base languages
 - bidirectional language support 195
 - choosing for searches 209
 - codes, adding 225, 229
 - codes, determining 226
 - codes, understanding PeopleSoft
 - /ISO 15
 - cubes 120
 - See Also* Cube Manager
 - databases, using non-Unicode 54

- defining locales 22
- definitions 225
- enabling 231
- fonts 55
 - See Also* fonts
- Global Multi-Language CD-ROM 173
- language-specific configuration files 129
- Multi Language Entry 191
 - See Also* Multi Language Entry
- PeopleTools Language Development Kit 237
- preferences for browsers, applying 15
- preferences, setting/changing 12
- preferences, setting/changing Windows 17
- preferences, understanding 11
- profiles, controlling user 14
- profiles, overriding user 14
- PS/nVision, setting for 154, 155
- queries, language-sensitive 119
- queries, scheduled 120
- related language records 178
 - See Also* related language records
- related language tables 167
 - See Also* related language tables
- selecting for spell check 232
- session languages, changing 13
- session languages, choosing 12
- session languages, understanding 12
- setting in Configuration Manager 18
- setting up files for PCL/line printing 135
- setting up files for PostScript printing 137
- setting up PDF fonts for SQRs 133
- sorting 111
 - See Also* sorting
- Terminology Management tool 203
 - See Also* Terminology Management tool
- translating 161
 - See Also* translation
- using the Append_Report_Translation procedure 128
- using the Get_Field_Information procedure 127
- using the Init_Report_Translation procedure 126
- using the Strings table 122

See Also Strings table
 Latin character set 149
 Latin2 137
 Length() function 141
 Lengthp() function 141
 Lengtht() function 141
 %lib% environment variable 242
 licensing
 fonts 55
 IMEs 57
 Linux
 compiling COBOL 73
 determining character sets 228
 locating language resources 239
 PSSQR.UNX 129
 See Also PSSQR files
 selecting default character sets 55
 setting up files for PostScript
 printing 137
 using MAKEUNIX.BAT 243
 Locale Defaults page 23
 Locale Definition page 22
 locales
 defining 22
 ISO 15
 See Also ISO locales
 setting locale-based formatting 23
 understanding locale-based
 formatting 19
 Verity 231

M

MAKEALTL command 242
 MAKEALTL.BAT 241
 MAKEUNIX.BAT 243
 Malayalam character set 152
 Manage Installed Languages page 230
 menus
 translating definitions 255
 translating items 256
 messaging
 replacing message text 217
 translating component interface
 messages 259
 translating messages 254
 meta-SQL
 %BINARYSORT 71
 See Also %BINARYSORT meta-SQL
 %CurrentDateTime 27
 Microsoft Excel, *See* Excel

Microsoft Internet Explorer, *See* Internet Explorer
 Microsoft Windows, *See* Windows
 mobile pages
 translating data 261
 translating items 263
 translating page names/view labels 262
 MSSQL Server
 SQL ORDER BY sorts 113
 supporting Unicode 44
 understanding field length
 semantics 59
 Multi Language Entry
 editing data 190
 enabling 191
 entering data 192
 understanding 191
 My Personalizations page
 defining user-specific formatting 21
 understanding locale-based
 formatting 20

N

native2ascii utility 235
 NOCBGN directive 79
 NOCEND directive 79
 NOCLN directive 79
 nonshifting DBCSs 46
 Nordic character set 152
 notes xix
 numeric fields
 defining user-specific formatting 21
 handling in the COBOL Conversion
 utility 77
 setting system-wide formatting 24

O

objects
 adding translations 174
 designing for easy translation 161
 modifying searchable 223
 selecting search objects 222
 storing in multiple languages 167
 translating 243
 translating Application Engine 271
 translating business interlink 260
 translating business process 258
 translating menu 255
 translating message 254

- translating mobile 261
- translating portal 265
- translating Process Scheduler 260
- translating query 268
- translating record 268
- translating string 270
- upgrade issues 223
- viewing searchable 223
- Oracle
 - checking space/resources for base
 - language swaps 180
 - SQL ORDER BY sorts 113
 - supporting Unicode 44
 - understanding field length
 - semantics 58
- P**
 - page definitions
 - translating 247, 253
 - understanding 245
 - Page Field Properties dialog box 38
 - page fields
 - checking field length for non-Unicode
 - databases 60
 - enabling length checking 61
 - setting time zones 36, 38
 - understanding field length
 - semantics 58
 - pages
 - designing global-ready 162
 - entering data in multiple languages 192
 - fields 36
 - See Also* page fields
 - freezing layout 162
 - HTML 55
 - See Also* HTML pages
 - search status 206
 - sizing/ordering controls 162
 - translating labels 161
 - translating review page text 272
 - using translation pages 252
 - %path% environment variable 242
 - PCL printing 135, 143
 - PDF fonts
 - printing Chinese, Japanese, and Korean
 - text 134
 - setting up for SQRs 133
 - setting up Japanese user-defined
 - fonts 134
 - PeopleBooks
 - ordering xvi
 - PeopleCode
 - binary sorting 114
 - forcing binary sorts in SQL 115
 - functions 13
 - See Also* functions
 - maintaining programs on related
 - language records 182
 - meta-SQL 114
 - See Also* meta-SQL
 - setting fonts 56
 - PeopleCode COBOL on z/OS
 - understanding 95
 - PeopleCode, typographical
 - conventions xviii
 - PeopleSoft Application Designer, *See* Application Designer
 - PeopleSoft COBOL, *See* COBOL
 - PeopleSoft COBOL on z/OS
 - data usage for Unicode storage 96
 - fine tuning 108
 - input/output 97
 - Unicode encodings 95
 - PeopleSoft COBOL on z/OZ
 - Unicode Conversion Utility for
 - z/OS 98
 - PeopleSoft Configuration Manager, *See* Configuration Manager
 - PeopleSoft Cube Manager, *See* Cube Manager
 - PeopleSoft Data Mover, *See* Data Mover
 - PeopleSoft nVision Layout Definition
 - dialog box 156
 - PeopleSoft Pure Internet Architecture 196
 - See Also* bidirectional language support
 - bidirectional pages, accessing 196
 - bidirectional pages, developing 198
 - bidirectional pages, developing HTML
 - areas 199
 - bidirectional pages, navigating in 197
 - bidirectional pages, working with
 - images 199
 - managing character sets 47, 55
 - managing fonts 55
 - supporting IMEs 57
 - PeopleSoft Query 119
 - PeopleTools
 - Global Technology xxiii
 - See Also* Global Technology
 - handling international characters 61

- installing PeopleSoft-provided translations 173
 - Language Development Kit 237
 - setting options 28
 - See Also* PeopleTools Options page
 - sorting 111
 - See Also* sorting
 - supporting Unicode 43
 - Terminology Management tool 203
 - See Also* Terminology Management tool
 - translating 237
 - See Also* translation
 - PeopleTools Options page
 - enabling field length checking 61
 - enabling system-wide currency settings 66
 - Multi-Currency check box 65, 66
 - setting base time zones 28
 - setting sort orders 115
 - personalization, *See* My Personalizations page
 - Plane 0 43
 - portals, translating 265
 - PostScript printing 137, 143
 - PowerPlay
 - Unicode support 48
 - using local-language cubes 120
 - prerequisites xv
 - printed documentation xvi
 - printing
 - international text in SQR for PeopleSoft 139
 - international text in SQRs 131
 - setting up files for PCL/line 135
 - setting up files for PostScript printing 137
 - using A4 paper 124
 - Process Scheduler
 - language-specific configuration files 129
 - translating definitions 260
 - programs
 - DATETIME.SQC 125
 - EUDCEDIT 134
 - SETENV.SQC 125
 - SQRTRANS.SQC 126
 - projects
 - translating definitions in 246
 - upgrading customizations to multilingual databases 176
 - viewing in Translation Designer 247
 - properties files
 - adding for languages 234
 - saving/formatting 235
 - translating 243
 - using the native2ascii utility 235
 - PS/nVision
 - design-time globalization features 155
 - retrieving labels 158
 - runtime language features 157
 - searching for files 157
 - setting up string criteria 156
 - translating reports 157
 - understanding 154
 - variables 158
 - See Also* PS/nVision variables
 - PS/nVision Report Request page 157
 - PS/nVision variables
 - evaluating 122
 - understanding 158
 - pscbl.mak command 73
 - PSLANGUAGES table
 - adding new languages 230
 - determining the signon language 16
 - PSPNLFIELD table 180
 - PSSQR files
 - advanced settings 130
 - checking encoding settings 139
 - setting up files for PostScript printing 137
 - setting up fonts 131
 - setting up PDF fonts 133
 - understanding 129
 - PSTZOFFSET table 31
 - %PSVER% environment variable 243
 - PTPDYSQL program 83
 - PTPSTRFN function
 - requesting character string map 88
 - requesting single character length 85
 - understanding 85
- ## Q
- queries
 - generating time zone offsets 31
 - language-sensitive 119
 - languages for scheduled 120
 - translating 268

R

- Record Field Properties dialog box 37
- record fields
 - associating with group boxes 162
 - setting time zones 35, 37
- records
 - base 179
 - See Also* base records
 - record fields 162
 - See Also* record fields
 - related language 178
 - See Also* related language records
 - translating 268
- Regional Options dialog box 24
- related documentation xvi
- related language records
 - designing 181
 - fixing invalid 178
 - making LANGUAGE_CD a key field 178
- related language tables
 - associating with base tables 171
 - base and related language – one-to-one view 183
 - base tables and related language table – two-to-one view 184
 - base tables and related language tables – two-to-two view 185
 - cloning 170
 - creating 181
 - fixing swap errors 179
 - maintaining 189
 - requirements 170
 - storing translations 171
 - structure 170
 - understanding 167
 - using 172
- related language views
 - creating 182
 - fixing wrong key structures 178
 - one base table and one related language table 183
 - two base tables and one related language table 184
 - two base tables and two related language tables 185
- related languages
 - records 178
 - See Also* related language records

- tables 167

- See Also* related language tables

- views 178

- See Also* related language views

- Replace Data page 215

- Replace Messages page 217

- Replace process

- disabling 210, 217

- enabling 217

- replacing message text 217

- replacing non-message text 215

- reviewing replacement text 219

- running 218

- understanding 205

- undoing replacements 219, 220

- verifying replacement text 216

- viewing/replacing search results 215

- Replace Results/Undo page 219, 222

- reports

- PS/nVision 154

- See Also* PS/nVision

- SQR for PeopleSoft 124

- See Also* SQR for PeopleSoft

S

- SBCS

- checking field length for non-Unicode

- databases 60

- understanding 46

- Search Profile page 208, 221

- Search Results/Replace page 218

- searches

- choosing languages 209

- criteria, cloning 213

- criteria, defining 208

- criteria, reviewing 221

- criteria, sample 213

- defining text 212

- defining the scope 210

- enabling processing 209

- granting search privileges 207

- objects, modifying searchable 223

- objects, selecting 222

- objects, viewing searchable 223

- page search status 206

- results, reviewing 215

- results, viewing all 222

- results, viewing search-only 222

- results, viewing/replacing 215

- running searches 214

- searching/replacing in Translation Designer 250
 - Terminology Management tool 204
 - security
 - accessing definitions 251
 - granting search privileges 207
 - personalizing locale-based formatting 20
 - Set International Field Sizes SQR utility 67
 - SETENV.SQC program 125
 - SetLanguage function 13
 - Shift-JIS
 - checking field length for non-Unicode databases 61
 - forcing binary sorts in SQL 115
 - limitations 42
 - nonshifting DBCSs 47
 - understanding 42
 - shifting DBCSs 47
 - signon
 - adding languages for 233
 - applying browser language preferences for 15
 - bidirectional PIA pages 196
 - overriding user profile languages 14
 - session languages, choosing 12
 - session languages, understanding 12
 - Simple Mail Transfer Protocol (SMTP) character sets 57
 - single-byte character sets (SBCS), *See* SBCS
 - Slavic character set 152
 - SMTP character sets 57
 - sorting
 - binary 111, 114
 - See Also* binary sorting
 - characters with diacritics 112
 - database-level SQL ORDER BY 113
 - ideographic languages 112
 - PeopleTools in-memory 114
 - performing in COBOL/SQR 116
 - setting sort orders 115
 - special characters 112
 - understanding 111, 112
 - specified time zones 28
 - spell check, selecting language for 230, 232
 - SPF Viewer 139
 - SQL
 - database-level SQL ORDER BY sorts 113
 - forcing binary sorts in 115
 - setting column lengths in dynamic 83
 - SQR
 - PeopleSoft 124
 - See Also* SQR for PeopleSoft
 - running SWPAUDIT.SQR 178
 - sorting 117
 - SQR for PeopleSoft
 - checking encoding settings 139
 - configuring to process international text 129
 - formatting report dates/times 125
 - measuring string length 140
 - printing for A4 paper 124
 - printing international text 131
 - printing limitations for international text 139
 - PSSQR files 129
 - See Also* PSSQR files
 - setting currency precision 125
 - setting up files for PCL/line printing 135
 - setting up files for PostScript printing 137
 - setting up fonts 131
 - setting up PDF fonts 133
 - Strings table enabled SQR (sample) 128
 - supporting character sets 142
 - translating reports 126
 - understanding output types 143
 - using the Append_Report_Translation procedure 128
 - using the Get_Field_Information procedure 127
 - using the Init_Report_Translation procedure 126
 - using the OPEN command 141
 - working with international text 139
- SQR Open command 141
 - SQR OPEN command 130, 141
 - \$sqr-encoding system variable 139
 - \$sqr-encoding-console system variable 139
 - \$sqr-encoding-database-api system variable 139
 - \$sqr-encoding-file-input system variable 139

- \$sqr-encoding-file-output system variable 139, 141
- \$sqr-encoding-report-output system variable 139
- \$sqr-encoding-source system variable 139
- SQRTANS.SQC program 126
- status
 - page search 206
 - Replace process 210, 218
- string definitions
 - evaluating PS/nVision variables 122
 - printing/formatting SQR output for multiple languages 126
- string variables
 - evaluating PS/nVision 122
 - using for PS/nVision 155
- strings
 - characters 140
 - See Also* character strings
 - definitions 122
 - See Also* string definitions
 - setting up criteria for PS/nVision 156
 - string functions 62
 - Strings table 122
 - See Also* Strings table
 - translating 270
 - variables 155
 - See Also* string variables
- Strings table
 - enabled SQR (sample) 128
 - evaluating PS/nVision variables 122
 - Strings Table utility 155
 - translating SQRs 126
 - translating strings 270
 - understanding 122
 - using the Append_Report_Translation procedure 128
 - using the Get_Field_Information procedure 127
- style sheets 56
- Substr() function 141
- Substrp() function 141
- Substrt() function 141
- suggestions, submitting xx
- SWAP_BASE_LANGUAGE
 - command 177, 180
- SWPAUDIT.SQR 178
- Symbol character set 152
- SYSDATE system variable 27

- system variables
 - %BaseTimeZone 36, 38
 - checking encoding settings 139
 - %clienttimezone 38
 - %ClientTimeZone 36
 - %Language 13
 - %Language_Base 13
 - \$sqr-encoding-file-output 141
 - SYSDATE 27

T

- tables
 - base 168
 - See Also* base tables
 - Currency Code 63
 - International Field Sizes 67
 - PSLANGUAGES 16
 - See Also* PSLANGUAGES table
 - PSPNLFIELD 180
 - PSTZOFFSET 31
 - related language 167
 - See Also* related language tables
 - Strings 122
 - See Also* Strings table
 - Translate 229
 - See Also* Translate table
 - understanding base language 167
- Terminology Management tool
 - choosing languages to search 209
 - defining search text 212
 - defining the search scope 210
 - enabling search processing 209
 - granting search privileges 207
 - objects, modifying searchable 223
 - objects, selecting search 222
 - objects, viewing searchable 223
 - Replace process 205
 - See Also* Replace process
 - running searches 214
 - search criteria, cloning 213
 - search criteria, defining 208
 - search criteria, reviewing 221
 - search criteria, sample 213
 - search results, reviewing 215
 - search results, viewing all 222
 - search results, viewing/replacing 215
 - search status 206
 - search-only results, viewing 222
 - searching 204
 - understanding 203

- Undo process 206
 - See Also* Undo process
- upgrade considerations 223
- terms 291
- text
 - adding text property files for
 - languages 234
 - bi-directional 139
 - See Also* bi-directional text
 - defining for searches 212
 - international 129
 - See Also* international text
- Text Search Records page 222, 223
- Thai character set 153
- time fields
 - handling in the COBOL Conversion
 - utility 77
 - understanding time zone display
 - settings 35
- Time Zone Data page
 - Daylight Savings tab 31
 - Time Zone tab 30
- Time Zone Offset Generation page 31
- time zones
 - base 27
 - See Also* base time zones
 - complying with Energy Policy Act of
 - 2005 33
 - control precedence 36
 - maintaining 30
 - setting daylight saving time
 - properties 31
 - setting display options when developing
 - PeopleSoft application objects 35
 - setting in page field properties 38
 - setting in record field properties 37
 - translating time zone labels 278
 - understanding 27
 - understanding display settings 35
 - viewing daylight savings time start/end
 - dates 32
 - viewing definitions 30
- Tiro Typeworks 56
- TLSINST1 utility 67
- TOOLBIN environment variable 241
- %toolbin% environment variable 242
- Translate Application Data page 273
- Translate IB Routings page 275
- Translate IB Services page 275
- Translate Message Definitions page 275
- Translate Message Queues page 275
- Translate Messages page 254
- Translate Mobile Page Items page 263
- Translate Mobile Page Names page 262
- Translate Query utility 269
- Translate Service Operation Handlers
 - page 275
- Translate Service Operation Versions
 - page 275
- Translate Service Operations page 275
- Translate table
 - adding language codes 229
 - translating values 247, 253
- Translate XMLP Data Source Definitions
 - page 276
- Translate XMLP Definitions Template
 - page 276
- Translate XMLP Report Definitions
 - page 276
- Translation Designer
 - copying translations to Excel 251
 - entering translations 250
 - expanding/collapsing nodes 249
 - navigating 250
 - opening 246
 - resizing columns/rows 249
 - searching/replacing 250
 - translating review page text 272
 - understanding 245
 - using Application Designer
 - features 251
 - using the information bar 247
 - viewing/hiding rows 249
- translation pages 252
- translations
 - adding to existing databases 174
 - bidirectional language support 195
 - designing global-ready pages 162
 - developing applications 161
 - installing PeopleSoft-provided 173
 - locating resource directories 239
 - message definitions 275
 - Multi Language Entry 190, 191
 - See Also* Multi Language Entry
 - PS/nVision 157
 - queues 275
 - required files and tools 237
 - routing definitions 275
 - service operation handlers 275
 - service operation versions 275

- service operations 275
- services 275
- size impact 174
- SQR for PeopleSoft 126
- storing in related language tables 171
- Terminology Management tool 203
 - See Also* Terminology Management tool
- time zone labels 278
- Translate table 253
 - See Also* Translate table
- translating ACE analytic model ID definitions 277
- translating application data 272, 273
- translating Application Engine definitions 271
- translating business interlinks 260
- translating business processes 258
- translating component interface messages 259
- translating folder tabs 257, 258
- translating HTML definitions 274
- translating images 244
- translating menu definitions 255
- translating menu items 256
- translating messages 254
- translating mobile data 261, 262
- translating mobile page items 263
- translating objects 243
- translating pages 253
- translating PeopleSoft Integration Broker definitions 275
- translating portals 265
- translating Process Scheduler definitions 260
- translating properties files 243
- translating queries 268
- translating record definitions 268
- translating resource files 240
- translating review page text 272
- translating strings 270
- translating trees 266
- Translation Designer 245
 - See Also* Translation Designer
- understanding 237
- understanding custom 225
- understanding options 245
- understanding related language tables 168
- understanding selective 161

- upgrading customizations to multilingual databases 176
- using alternate-language DLLs 238
- using MAKEALTL.BAT 241, 243
- using the Append_Report_Translation procedure 128
- using the Get_Field_Information procedure 127
- using the Init_Report_Translation procedure 126
- XMLP data sources 276
- XMLP report definitions 276
- XMLP template definitions 276
- trees, translating 266
- Turkish character set 153
- typographical conventions xviii

U

- UCS encoding 42, 44, 45, 69
- Undo Messages page 220
- Undo process
 - enabling/disabling 219
 - running 220
 - understanding 206
 - undoing replacements 219, 220
- Unicode 95
 - See Also* PeopleSoft COBOL on z/OS architecture/support 47
 - binary sorting, using 114
 - binary sorts in SQL, forcing 115
 - character sets, application server 55
 - character sets, converting 48
 - character sets/fonts in PeopleSoft Pure Internet Architecture, managing 55
 - COBOL Conversion utility 72
 - See Also* COBOL Conversion utility
 - COBOL, storage requirements for 70
 - converting PeopleSoft systems 281
 - converting PeopleSoft systems on DB2/UDB databases 288
 - converting PeopleSoft systems on Microsoft SQL server databases 288
 - converting PeopleSoft systems on Oracle databases 283
 - databases, supporting in 48
 - databases, using Unicode-encoded 54
 - encoding (examples) 45
 - encodings 43
 - encodings for PeopleSoft COBOL 69
 - encodings, supported 153

- error messages 71
 - identifying ANSI vs Unicode data 83
 - operations that don't support 228
 - parsing strings 71, 85
 - printing Chinese, Japanese, and Korean PDF text 134
 - printing limitations in SQR for PeopleSoft 139
 - processing international text 129
 - saving properties files 235
 - setting column lengths in dynamic SQL 83
 - understanding 42
 - understanding field length semantics 58
 - Unicode Conversion Utility for z/OS 98
 - applying patches 98
 - converted COBOL programs, identifying 100
 - converted data, understanding 100
 - error logging 106
 - exception logging 106
 - running, automatically 99
 - running, manually 99
 - summary logging 106
 - universal coordinated time (UTC) 27
 - Universal Multiple-Octet Coded Character Set (UCS) 42
 - Unix
 - adding translations to databases 174
 - compiling COBOL 73
 - determining character sets 228
 - locating language resources 239
 - PSSQR.UNX file 129
 - See Also* PSSQR files
 - selecting default character sets 55
 - setting up files for PostScript printing 137
 - using MAKEUNIX.BAT 243
 - upgrades
 - adding translations 174
 - changing terminology 223
 - upgrading customizations to multilingual databases 176
 - US-ASCII, *See* ASCII
 - user profile language preferences 14
 - User Profiles page 14
 - User Search Options page 207
 - users
 - granting search privileges 207
 - understanding user time zones 28
 - UTC 27
 - UTF-16 encoding 44
 - UTF-32 encoding 43, 44
 - UTF-8 encoding
 - running COBOL 69
 - understanding 44, 45
 - utilities
 - COBOL Conversion 72
 - See Also* COBOL Conversion utility
 - International Field Size 67
 - native2ascii 235
 - Strings Table 155
 - Translate Query 269
 - translation 258
- ## V
- variables
 - environment 241
 - See Also* environment variables
 - PS/nVision 158
 - See Also* PS/nVision variables
 - string 155
 - See Also* string variables
 - system 13
 - See Also* system variables
 - Verity locales 231
 - Vietnamese character set 154
 - visual cues xix
- ## W
- warnings xix
 - BIND/SELECT log 81
 - COBOL conversion logs 81
 - exception date log 82
 - exception log 81
 - web servers
 - bypassing the signon page 14
 - modifying the signon page 233
 - translating properties files 243
 - Windows
 - adding languages to Configuration Manager 232
 - adding translations to databases 174
 - character sets for CSV queries, selecting 232
 - character sets, supporting Unicode 44
 - character sets/languages, managing 56
 - compiling COBOL 73

- copying translations from Translation Designer 251
- locating language resources 239
- PS/nVision path search feature 157
- PSSQR.INI file 129
 - See Also* PSSQR files
- setting regional options 24
- setting up files for PostScript printing 137
- setting up user-defined fonts 134
- setting/changing language preferences 17
- supporting IMEs 57
- using MAKEALTL.BAT 241

X

- XLAT values, *See* Translate table

