

---

# Enterprise PeopleTools 8.50 PeopleBook: PeopleSoft Application Designer Developer's Guide

---

**September 2009**

Copyright © 1988, 2009, Oracle and/or its affiliates. All rights reserved.

### **Trademark Notice**

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

### **License Restrictions Warranty/Consequential Damages Disclaimer**

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

### **Warranty Disclaimer**

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

### **Restricted Rights Notice**

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. 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, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007).  
Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

### **Hazardous Applications Notice**

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

### **Third Party Content, Products, and Services Disclaimer**

This software and documentation may provide access to or information on content, products and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third party content, products and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third party content, products or services.

# Contents

## Preface

<b>PeopleSoft Application Designer Preface .....</b>	<b>xxi</b>
PeopleSoft Application Designer .....	xxi

## Part 1

### Getting Started with PeopleSoft Application Designer

#### Chapter 1

<b>Getting Started with PeopleSoft Application Designer .....</b>	<b>3</b>
PeopleSoft Application Designer Overview .....	3
PeopleSoft Application Designer Implementation .....	3
Installation-Level Steps .....	3
Main Application Development Steps .....	4
Other Sources of Information .....	5

## Part 2

### Developing and Customizing PeopleSoft Applications

#### Chapter 2

<b>Using PeopleSoft Application Designer .....</b>	<b>9</b>
Understanding PeopleSoft Application Designer .....	9
Using PeopleSoft Application Designer .....	9
Understanding the Eight Steps of Application Development .....	10
Starting PeopleSoft Application Designer .....	11
Using the PeopleSoft Application Designer Window Components .....	14
Setting PeopleTools Options .....	21
Working with PeopleCode .....	22
Using the Development Editors .....	24

Working with Definitions .....	25
Understanding Definitions .....	26
Opening Definitions .....	28
Viewing Definitions .....	29
Viewing and Editing Definition Properties .....	29
Creating Definitions .....	30
Saving Definitions .....	31
Closing Definitions .....	32
Renaming Definitions .....	32
Deleting Definitions .....	33
Finding Definitions .....	34
Using the Multiple Document Interface .....	36
Using Drag-and-Drop .....	37
Using Dynamic Toolbars and Menus .....	37
Using Pop-up Menus .....	37
Configuring the Environment .....	38
Working with Generated Files .....	40
Using Miscellaneous Definitions .....	40
Using Spell Check .....	42
Understanding Spell Check .....	42
Enabling Spell Check .....	46
Configuring Browser Options .....	46
Building and Maintaining Data .....	47
Data Building and Maintenance .....	48
Creating SQL Tables .....	48

## Chapter 3

<b>Working With Projects .....</b>	<b>49</b>
Understanding Projects .....	49
Viewing Projects .....	50
Accessing Definitions for a Project .....	50
Using the Upgrade View .....	51
Creating and Saving Projects .....	51
Creating a New Project .....	52
Saving a Project .....	52
Setting Project Properties .....	52
Inserting Definitions Into a Project .....	53
Removing Definitions From a Project .....	55
Merging Projects .....	55
Using Change Projects .....	57
Setting Project Options .....	57
Validating Projects .....	58

Setting System IDs .....	59
--------------------------	----

## Chapter 4

<b>Creating Field Definitions .....</b>	<b>61</b>
Understanding Field Definitions .....	61
Creating New Field Definitions .....	61
Creating a New Field Definition .....	62
Specifying Attributes for a New Field Definition .....	63
Specifying Character Field Attributes .....	65
Specifying Long Character Field Attributes .....	67
Specifying Number Field Attributes .....	68
Specifying Date Field Attributes .....	69
Specifying Time Field Attributes .....	70
Specifying DateTime Field Attributes .....	70
Specifying Image Field Attributes .....	70
Specifying ImageReference Field Attributes .....	71
Working With Additional Oracle Data Type Mappings .....	71
Working With Additional Microsoft SQL Server 2005 Data Type Mappings .....	72
Using Multiple Labels .....	73
Understanding Multiple Labels .....	73
Adding a New Label .....	73
Deleting a Label .....	74
Selecting Record Field Labels .....	74
Opening an Existing Field Definition .....	75
Viewing Field Properties .....	76
Changing Field Definitions .....	77
Understanding the Effects of Changing Field Definitions .....	77
Changing Field Types .....	77
Renaming Field Definitions .....	78
Deleting Field Definitions .....	79
Printing Field Definitions .....	80
Setting Custom Field Formats .....	80
Understanding Custom Field Formats .....	80
Understanding Format Notation .....	80
Changing Format Families .....	83
Editing Formats .....	85
Using Family Options .....	87
Testing Formats .....	88
Using the Translate Table .....	89
Understanding the Translate Table .....	89
Adding Values to the Translate Table .....	91
Changing Translate Values .....	92

Deleting Translate Values .....	93
Saving the Translate Table .....	93

## Chapter 5

<b>Planning Records, Control Tables, and TableSets .....</b>	<b>95</b>
Understanding the Planning Process .....	95
Table-Based Systems .....	95
Normalized Relational Databases .....	96
Record Definition Planning .....	98
Effective Dates .....	98
Control Tables .....	100
TableSets .....	101
Sharing Tables .....	104
Understanding Table Sharing .....	105
Adding the SetID Field to Record Definitions .....	107
Defining Set Control Fields .....	107
Modifying the Set Control Field .....	108
Creating SetIDs .....	109
Defining Record Groups .....	109
Defining TableSet Controls .....	110
Sharing Trees .....	110

## Chapter 6

<b>Creating Record Definitions .....</b>	<b>113</b>
Viewing Record Definitions .....	113
Common Elements Used in This Section .....	114
Viewing Basic Field Definitions .....	114
Viewing Key-Related Characteristics and Default Values .....	114
Viewing Editing Options .....	115
Viewing PeopleCode Program Types .....	116
Reordering Fields .....	116
Sizing and Sorting Columns .....	117
Saving Record Definitions .....	117
Naming Record Definitions .....	117
Creating a New Record .....	118
Opening an Existing Record .....	121
Manipulating Fields in Record Definitions .....	121
Understanding Fields in Record Definitions .....	121
Inserting Fields Into Records .....	122
Inserting Subrecords Into Records .....	122

Setting Record Properties .....	123
Setting General Properties .....	123
Setting Use Properties .....	123
Creating User-Defined Audit Record Definitions .....	126
Setting Record Field Properties .....	127
Understanding Record Field Properties .....	127
Understanding Record Key Considerations .....	127
Setting Record Field Use Properties .....	130
Adding From and Through Logic to the Search Page .....	135
Setting Record Field Edit Properties .....	137
Moving Fields in the Same Record .....	140
Moving Fields to Another Record Definition .....	140
Deleting a Field From a Record Definition .....	141
Renaming Record Definitions .....	141
Deleting Record Definitions .....	142
Printing Record Definitions .....	142
Creating SQL View and Dynamic View Select Statements .....	145

## Chapter 7

<b>Administering Data .....</b>	<b>147</b>
Understanding Data Administration and the Build Process .....	147
Running the Build Process .....	148
Understanding the Build Menu .....	149
Selecting the Build Scope .....	151
Selecting Build Options and Running the Build Process .....	151
Using Command Line Syntax to Build a Project .....	153
Creating Tables .....	155
Understanding the Create Table Process .....	155
Creating a Table .....	156
Specifying Create Options .....	157
Specifying Logging Options .....	159
Specifying Script Options .....	162
Confirming a Table Build .....	167
Creating Indexes .....	167
Understanding Indexes .....	168
Configuring Indexes .....	170
Adding a Custom Index .....	171
Editing the DDL .....	172
Altering Tables .....	173
Understanding the Alter Process .....	173
Determining Tables to be Altered After a Field Change .....	179
Altering a Table .....	179

Specifying Alter Options .....	180
Confirming Your Table Alter .....	183
Creating Triggers .....	183
Creating Views .....	184
Understanding Views .....	184
Creating Views .....	184
Using Joins in Views .....	185
Using Cross-Platform Meta-SQL for Dynamic Views .....	185
Administering DDL .....	186
Using the Record DDL .....	186
Setting the Tablespace .....	187
Using Physical Data Storage .....	188

## Chapter 8

<b>Designing Pages for Optimal Performance and Accessibility .....</b>	<b>191</b>
Improving Online Performance .....	191
Understanding Processing Modes .....	191
Refreshing a Page .....	192
Triggering Server Processing .....	192
Using PeopleCode in Deferred Mode .....	193
Using Page Controls in Deferred Mode .....	194
Using Error and Warning Messages .....	196
Designing Accessible Pages .....	197
Understanding Accessibility Issues .....	197
Using Labels .....	198
Using Color to Convey Meaning .....	200
Verifying Titles for Grids and Scroll Areas .....	201
Adding Unique Labels for Push Buttons and Links .....	201
Deactivating Adornment Images .....	201
Maintaining Logical Tab Order .....	201
Using Deferred Processing .....	201
Using Foreground and Background Colors .....	202
Using Instructional Text .....	202

## Chapter 9

<b>Using Page Controls .....</b>	<b>207</b>
Understanding Page Controls .....	207
Aesthetic Controls .....	208
Data Entry Controls .....	209
Function and Data Processing Controls .....	210

Using Charts .....	212
Using Frames .....	212
Using Group Boxes .....	213
Inserting and Moving a Group Box .....	213
Setting Group Box Properties .....	214
Using Horizontal Rules .....	216
Drawing a Horizontal Rule .....	216
Setting Horizontal Rule Properties .....	216
Using Images .....	217
Understanding Static Images .....	217
Adjusting Static Image Size and Shape .....	217
Setting Static Image Properties .....	217
Setting Image Properties .....	218
Using Static Text .....	219
Using Check Boxes .....	220
Setting Check Box Properties .....	220
Using Drop-Down List Boxes .....	221
Setting Drop-Down List Box Record Properties .....	221
Setting Drop-Down List Box Label and Use Properties .....	222
Using Radio Buttons .....	222
Understanding Radio Buttons .....	222
Setting Radio Button Record Properties .....	223
Setting Radio Button Label Properties .....	223
Setting Radio Button Use Properties .....	223
Using Edit Boxes and Long Edit Boxes .....	223
Understanding Edit Boxes and Long Edit Boxes .....	223
Using Popup Pages .....	225
Understanding Popup Pages .....	225
Defining Popup Pages .....	227
Associating Popup Pages with Fields .....	228
Enabling Message Catalog Popup Pages .....	229
Using Subpages .....	233
Understanding Subpages .....	233
Defining a Subpage .....	234
Inserting a Subpage Into a Page .....	234
Specifying Informational Subpage Labels .....	235
Using Secondary Pages .....	236
Understanding Secondary Pages .....	236
Defining Secondary Pages .....	237
Inserting a Secondary Page Control .....	239
Using HTML Areas .....	239
Understanding HTML Area Controls .....	240
Inserting an HTML Area .....	242
Populating an HTML Area .....	242
Changing an HTML Area Label .....	243

Using Push Buttons and Links .....	243
Understanding Push Buttons and Links .....	244
Inserting Push Buttons or Links .....	244
Specifying Type Properties for Push Buttons or Links .....	245
Specifying Label Properties for Push Buttons or Links .....	250
Specify Use Properties for Push Buttons or Links .....	254
Changing Hover Text for the Push Button or Link Programmatically .....	256

## Chapter 10

<b>Using Scroll Areas, Scroll Bars, and Grids .....</b>	<b>257</b>
Using Scroll Areas and Scroll Bars .....	257
Understanding Scroll Areas and Scroll Bars .....	257
Setting Scroll Area General Attributes .....	258
Setting Scroll Area Label Properties .....	260
Setting Scroll Area Use Properties .....	261
Setting Scroll Bar Label Properties .....	262
Setting Scroll Bar Use Properties .....	262
Controlling Scrollable Data .....	264
Converting Scroll Bars to Grids .....	265
Using Grids .....	265
Understanding Grid Control Scope .....	266
Understanding Grid Properties .....	266
Understanding Grid Customization .....	267
Inserting and Resizing Grid Controls .....	269
Inserting and Manipulating Grid Columns .....	269
Setting Column Properties .....	271
Creating a Tabbed Grid .....	271
Freezing Grid Columns .....	273
Using Multiple Grids on a Page .....	273
Enabling Grid Personalization .....	274
Setting Grid General Properties .....	275
Setting Grid Label Properties .....	275
Setting Grid Use Properties .....	283

## Chapter 11

<b>Creating Page Definitions .....</b>	<b>289</b>
Understanding Page Design .....	289
Level-Based Controls .....	290
Keys for Accessing Pages .....	294
Multiple Occurrences of Data .....	295

Prompt Fields .....	296
Derived and Work Fields .....	298
Sensitive Data .....	299
Hidden Pages .....	299
Upgrade Considerations .....	299
End User Page Customization .....	299
Using Page Development Tools .....	300
Using Page Definition Menus .....	300
Using Toolbars .....	301
Using the Page Layout Grid .....	303
Creating New Page Definitions .....	304
Cloning Page Definitions .....	304
Creating a Blank Page .....	305
Adding Page Controls .....	305
Dragging Field Definitions From a Record Onto a Page .....	305
Dragging Record Fields From the Project Workspace Onto a Page .....	306
Dragging Record Definitions Onto a Page .....	306
Using the Page Control Toolbar .....	307
Using the Insert Menu .....	307
Manipulating Page Controls .....	307
Selecting Controls .....	308
Resizing Controls .....	308
Using the Page Control Inspector .....	309
Deleting Controls .....	310
Moving Controls on the Same Page .....	310
Copying and Moving Controls to Another Page .....	310
Working With Field Labels .....	310
Setting Page Field Properties for Controls .....	311
Setting Record Properties .....	312
Setting Label Properties .....	317
Setting Use Properties .....	321
Setting General Properties .....	327
Setting Options Properties .....	328
Creating Display Control and Related Fields .....	332
Using Display Control and Related Fields .....	333
Creating Related Edit Fields .....	334
Creating Invisible Control Fields .....	335
Ordering and Spacing Page Controls .....	335
Understanding Page Control Order and Spacing .....	336
Ordering Controls Visually .....	336
Ordering Controls Logically .....	336
Testing Page Control Order .....	337
Applying Control Order Rules .....	338
Changing Control Order Using the Order List .....	338
Finding a Field on a Page Definition .....	339

Using Default Order .....	339
Spacing Controls on Pages .....	340
Designing Inquiry Pages .....	340
Aligning Page Controls .....	340
Maximizing Performance .....	341
Accessing PeopleCode in Page Definitions .....	342
Viewing Pages in the Browser .....	343
Working With the View in Browser Feature .....	344
Changing the Default Browser .....	345
Generating HTML .....	346
Viewing the HTML for a Page .....	347
Identifying Page Definitions Online .....	347
Producing Pages .....	349
Understanding Page Production .....	350
Changing General Page Information .....	350
Changing Page Use Information .....	351
Saving Pages .....	354
Renaming and Deleting Pages .....	354
Printing Page Definitions .....	355
Altering Page Setup .....	355
Reading Your Page Definition Report .....	356
Working with Pages in Foreign Languages .....	358
Understanding Working with Pages in a Foreign Language .....	358
Performing a Save As with Foreign Language Pages .....	358
Deleting Foreign Language Pages .....	359

## Chapter 12

<b>Creating Component Definitions .....</b>	<b>361</b>
Understanding Component Definitions .....	361
What Component Definitions Manage .....	361
Component Buffer .....	362
Defining Components .....	363
Understanding the Component Definition Window .....	363
Creating a New Component Definition .....	364
Opening an Existing Component Definition .....	365
Adding Pages to Components .....	365
Reordering Pages in a Component .....	366
Copying or Moving Pages to Another Component .....	367
Setting Page Attributes .....	367
Validating Components .....	369
Specifying Market-Specific Components .....	370
Determining Where a Component Is Used .....	370

Setting Component Properties .....	371
Opening the Component Properties Dialog Box .....	371
Setting General Properties .....	371
Setting Use Properties .....	371
Setting Internet Properties .....	374
Selecting the Search Record .....	379
Understanding Search Records .....	379
Creating a Search Record .....	380
Overriding the Search Record .....	381
Accessing the Message Catalog .....	381
Enabling the Expert Entry Feature .....	382
Understanding Expert Entry .....	382
Enabling Expert Entry Through Security .....	383

## Chapter 13

<b>Creating Menu Definitions .....</b>	<b>385</b>
Understanding Menus .....	385
Defining Custom Component Menus .....	385
Creating Custom Component Menus .....	386
Setting Menu Item Properties .....	387
Setting General Properties .....	388
Setting Use Properties .....	388
Working With Existing Menu Definitions .....	388
Adding Components to a Menu .....	389
Renaming Menu Definitions .....	389
Copying a Menu Definition .....	389
Deleting Menu Items .....	390
Printing Menu Definitions .....	390
Setting Up Menu Security .....	391
Importing Custom Menu Groups Into Portals .....	391
Defining Pop-up Menus .....	391
Understanding Pop-up Menus .....	391
Creating Pop-up Menus .....	392
Defining Transfer Menu Items .....	394

## Chapter 14

<b>Using the Registration Wizard .....</b>	<b>397</b>
Understanding the Registration Wizard .....	397
Common Elements Used in This Chapter .....	397
Working With the Registration Wizard .....	398

Registering Components and iScripts .....	398
Registering a Component in Multiple Portals .....	404
Registering a Mobile Page .....	404

## Chapter 15

<b>Creating Style Sheet Definitions .....</b>	<b>407</b>
Understanding Style Sheets and Style Classes .....	407
Style Sheet Types .....	408
Style Classes .....	411
Creating Standard and Sub Style Sheets .....	413
Creating Standard Style Sheets or Sub Style Sheets .....	413
Renaming Standard Style Sheets or Sub Style Sheets .....	414
Deleting Standard Style Sheets or Sub Style Sheets .....	414
Inserting Sub Style Sheets into Style Sheets .....	415
Inserting Style Classes into Style Sheets .....	415
Inserting Override Style Classes into Style Sheets .....	416
Setting Style Class Attributes .....	416
Understanding Class Attributes .....	417
Understanding Generic and Installed Fonts .....	417
Accessing Class Attributes .....	418
Specifying Fonts .....	418
Setting Font Attributes for a Specific Language .....	420
Specifying Spacing and Alignment .....	420
Specifying Background Attributes .....	423
Referencing a Background Image URL .....	425
Specifying Border Attributes .....	426
Specifying Margins .....	427
Specifying Miscellaneous Attributes .....	429
Working With Free Form Sub Style Sheets .....	432
Creating Free Form Sub Style Sheets .....	433
Inserting Free Form Sub Style Sheets into Style Sheets .....	433
Converting Style Sheets to Free Form Sub Style Sheets .....	433
Renaming Free Form Sub Style Sheets .....	434
Deleting Free Form Sub Style Sheets .....	434
Including Meta Variables and Comments in Free Form Sub Style Sheets .....	434
Changing Font and Color Settings For Free Form Sub Style Sheets .....	435
Working with the Default Style Sheet .....	437
Configuring PSSTYLEDEF as the Default Style Sheet .....	437
Configuring PSSTYLEDEF_SWAN as the Default Style Sheet .....	438
Configuring Other Default Style Sheets .....	440
Overriding Default Page Style Properties .....	440
Overriding the Default Page Style Sheet .....	441

Overriding the Default Page Background Style .....	441
Overriding the Default Page Field Styles .....	442
Modifying Page Tabs .....	443
Understanding Cascading Style Sheet Tabs .....	443
Creating Tab Images .....	444
Customizing Page Tab Style Sheets .....	445
Modifying Push Buttons in SWAN Style Sheets .....	446
Understanding Cascading Style Sheet Push Buttons .....	446
Creating SWAN Style Sheet Push Button Images .....	447
Customizing SWAN Push Buttons in HTML Areas .....	447

## Chapter 16

<b>Creating Image Definitions .....</b>	<b>449</b>
Understanding Images .....	449
Creating New Image Definitions .....	450
Understanding Image Definitions .....	450
Creating an Image Definition .....	451
Setting Image Properties .....	451
Importing and Creating Alternate Image Types .....	452
Opening an Image Definition .....	453
Updating an Image Definition .....	453
Updating an Image Definition .....	453
Changing the Image Display Size .....	454
Specifying the Image Storage Format .....	454
Converting Images .....	454
Consolidating Images .....	455
Using the Image Catalog .....	457

## Chapter 17

<b>Creating HTML Definitions .....</b>	<b>463</b>
Understanding HTML Definitions .....	463
Creating an HTML Definition .....	463
Opening an HTML Definition .....	464
Referencing HTML Definitions Dynamically .....	464

## Part 3

### Maintaining Data

#### Chapter 18

<b>Constructing File Layouts and Performing Data Interchanges .....</b>	<b>467</b>
Understanding File Layouts .....	467
Understanding File Layouts .....	467
Field Formats in a Flat File .....	468
FIXED Format Files .....	469
CSV Format Files .....	470
XML Format Files .....	472
Constructing File Layouts .....	473
Creating a New File Layout Definition .....	473
Adding File Records and File Fields .....	474
Reordering File Layout Nodes .....	476
Naming File Layouts, Records, and Fields .....	476
Applying Field Formats .....	477
Specifying File Layout, Record, and Field Properties .....	478
Specifying File Layout Properties .....	478
Specifying File Layout Segment Properties .....	481
Specifying File Layout Field Properties .....	483
Performing Data Interchanges .....	487
Understanding the Import Process .....	487
Previewing Input Data .....	488
Generating and Running the Import Program .....	491
Exporting Data .....	491
Producing a Sample File Layout .....	492
Creating the File Layout Definition .....	492
Adjusting File Layout Properties .....	494
Inserting a Segment and a Field .....	495

#### Chapter 19

<b>Using Command Line Parameters .....</b>	<b>497</b>
Understanding PeopleSoft Application Designer Command Line Parameters .....	497
Starting PeopleSoft Application Designer Command Line Interface .....	498
Building Projects .....	500

Copying Definitions .....	502
Comparing Definitions for Upgrade Using the Command Line .....	509

## Chapter 20

<b>Using Change Control .....</b>	<b>517</b>
Understanding Change Control Features .....	517
Change Control Locking .....	517
Change Control History .....	519
Change Control Stamping .....	520
Understanding Implementation Considerations .....	521
Setting Up Change Control .....	522
Understanding Change Control Security .....	522
Appointing a Change Control Administrator .....	522
Enabling or Disabling Change Control .....	522
Using Projects .....	523
Understanding Projects .....	523
Using Multiple Databases for Development .....	524
Using Distributed Development Environments .....	524
Using Change Control .....	525
Locking and Unlocking Definitions .....	525
Viewing Locked Definitions .....	527
Inserting Comments .....	527
Deleting Page Definitions .....	529
Viewing Change Control History .....	529
Reporting Change Control Information .....	530

## Appendix A

<b>Understanding Functional Indexes .....</b>	<b>539</b>
Functional Indexes for Microsoft SQL Server 2000 .....	539
DB2/UDB for Linux, UNIX, and Windows .....	541
Functional Indexes for Informix .....	541

## Appendix B

<b>Referencing Selections for Command Line Parameters .....</b>	<b>545</b>
Database Type Selections .....	545
Definition Type Selections .....	546
Language Selections .....	550
Available Release Numbers .....	551

**Appendix C**

<b>Using Default Style Classes .....</b>	<b>553</b>
Default Style Classes .....	553

**Appendix D**

<b>Modifying Classic Style Sheet Tabs .....</b>	<b>573</b>
Understanding Classic Tab Definitions .....	573
Modifying Classic Style Sheet Tabs .....	574
Creating Tab Images .....	574

**Appendix E**

<b>Performing Bulk Operations .....</b>	<b>579</b>
Understanding Bulk Operations .....	579
Performing Bulk Insertions .....	579
Common Elements Used in This Section .....	580
Inserting Fields into Records .....	580
Inserting Record Fields onto Pages .....	580
Inserting Fields into Both Records and onto Pages .....	581
Performing Bulk Modifications .....	581
Modifying Record Fields .....	581
Modifying Page Fields .....	582
Reordering or Resizing Grid Fields .....	582
Changing Page Field Attributes .....	583
Performing Bulk Deletions .....	583

**Appendix F**

<b>Running PeopleTools Cross-Reference Reports .....</b>	<b>585</b>
Understanding Cross-Reference Reports .....	585
Running a Crystal Report .....	587

**Appendix G**

<b>Working With Records Used in the Mobile Synchronization Framework .....</b>	<b>589</b>
--	------------

Understanding Mobile Synchronization Framework Records .....	589
Configuring Records for the Mobile Synchronization Framework .....	589
Setting Record Properties .....	590
Working with Required Fields and Default Values .....	590
Example: Building Records for the Mobile Synchronization Framework .....	591

## **Appendix H**

<b>Creating Custom Plug-in Files for the Rich Text Editor .....</b>	<b>593</b>
Creating Custom Buttons .....	593
Using Images with Plug-Ins .....	594
Adding Multilingual Support .....	594
Adding Plug-Ins to the Configuration File .....	595
Specifying the Plugin File ID Value .....	595

<b>Index .....</b>	<b>597</b>
--------------------	------------



# PeopleSoft Application Designer Preface

This preface provides an overview of the contents in PeopleSoft Application Designer.

---

## PeopleSoft Application Designer

This PeopleBook provides an overview of PeopleSoft Application Designer and discusses how to:

- Develop and customize PeopleSoft Applications for the PeopleSoft Pure Internet Architecture.
- Maintain data using PeopleSoft Application Designer.
- Perform updates and upgrades using PeopleSoft Application Designer.

For information that is specific to your application, please refer to the application PeopleBook for your specific PeopleSoft product.

---

## PeopleBooks and the Online PeopleSoft Library

A companion PeopleBook called PeopleBooks and the Online PeopleSoft Library contains general information, including:

- Understanding the PeopleSoft online library and related documentation.
- How to send PeopleSoft documentation comments and suggestions to Oracle.
- How to access hosted PeopleBooks, downloadable HTML PeopleBooks, and downloadable PDF PeopleBooks as well as documentation updates.
- Understanding PeopleBook structure.
- Typographical conventions and visual cues used in PeopleBooks.
- ISO country codes and currency codes.
- PeopleBooks that are common across multiple applications.
- Common elements used in PeopleBooks.
- Navigating the PeopleBooks interface and searching the PeopleSoft online library.
- Displaying and printing screen shots and graphics in PeopleBooks.
- How to manage the PeopleSoft online library including full-text searching and configuring a reverse proxy server.
- Understanding documentation integration and how to integrate customized documentation into the library.
- Glossary of useful PeopleSoft terms that are used in PeopleBooks.

You can find this companion PeopleBook in your PeopleSoft online library.

## **Part 1**

# **Getting Started with PeopleSoft Application Designer**

## **Chapter 1**

### **Getting Started with PeopleSoft Application Designer**



## Chapter 1

# Getting Started with PeopleSoft Application Designer

This chapter provides an overview of the PeopleSoft Application Designer product and discusses the implementation tasks necessary to work with this tool.

---

## PeopleSoft Application Designer Overview

PeopleSoft Application Designer is the core development tool used to build and modify PeopleSoft applications. PeopleSoft Application Designer enables you to build a variety of definitions, including:

- Fields.
- Records (Tables).
- Pages.
- Components.
- Menus.
- PeopleCode.
- SQL.

---

## PeopleSoft Application Designer Implementation

This section discusses:

- Installation-level steps.
- Main application development steps.

### Installation-Level Steps

Before you can begin using PeopleSoft Application Designer, the following items must be in place.

<b>Step</b>	<b>Reference</b>
<p>Set up security.</p> <p>For developing and customizing PeopleSoft applications you will need access to the definitions involved, such as fields, records, pages, and so on. Your administrator can provide access by adjusting your definition security settings. There are also general PeopleTools permissions for maintaining and upgrading data that your administrator may need to provide access to as well. These include Build and Data Administration, Change Control, and Upgrade.</p>	<p>See <i>Enterprise PeopleTools 8.50 PeopleBook: Security Administration</i>, "Implementing Definition Security."</p> <p>See <i>Enterprise PeopleTools 8.50 PeopleBook: Security Administration</i>, "Setting Up Permission Lists," Setting PeopleTools Permissions.</p>
<p>Set up workstations.</p> <p>PeopleSoft Application Designer is a windows-based program. To run PeopleSoft Application Designer, you need to have a supported version of Windows with ample CPU and memory resources, and you need to have the workstation configured using PeopleSoft Configuration Manager so that there is connectivity to the appropriate database and all required environment variables are set.</p>	<p>See <i>Enterprise PeopleTools 8.50 PeopleBook: System and Server Administration</i>, "Using PeopleSoft Configuration Manager," Configuring Developer Workstations.</p>

## Main Application Development Steps

This table presents the main steps for creating a PeopleSoft application.

<b>Step</b>	<b>Reference</b>
Design application.	Review your in-house development cycle and documentation.
Create fields.	See <a href="#">Chapter 4, "Creating Field Definitions," page 61.</a>
Create record definitions.	<p>See <a href="#">Chapter 5, "Planning Records, Control Tables, and TableSets," page 95.</a></p> <p>See <a href="#">Chapter 6, "Creating Record Definitions," page 113.</a></p>
Build SQL tables.	See <a href="#">Chapter 7, "Administering Data," page 147.</a>
Create pages.	<p>See <a href="#">Chapter 8, "Designing Pages for Optimal Performance and Accessibility," page 191.</a></p> <p>See <a href="#">Chapter 9, "Using Page Controls," page 207.</a></p> <p>See <a href="#">Chapter 11, "Creating Page Definitions," page 289.</a></p>
Create components.	See <a href="#">Chapter 12, "Creating Component Definitions," page 361.</a>
Register components.	See <a href="#">Chapter 14, "Using the Registration Wizard," page 397.</a>

<b>Step</b>	<b>Reference</b>
Test the application.	Review your in-house testing procedures and documentation.

## Other Sources of Information

This section provides information to consider before you begin to use PeopleSoft Application Designer. In addition to implementation considerations presented in this section, take advantage of all PeopleSoft sources of information, including the installation guides, release notes, PeopleBooks, and training courses.

### **See Also**

*Enterprise PeopleTools 8.50 PeopleBook: Getting Started with Enterprise PeopleTools*, "Getting Started with Enterprise PeopleTools Preface"



## **Part 2**

# **Developing and Customizing PeopleSoft Applications**

### **Chapter 2**

#### **Using PeopleSoft Application Designer**

### **Chapter 3**

#### **Working With Projects**

### **Chapter 4**

#### **Creating Field Definitions**

### **Chapter 5**

#### **Planning Records, Control Tables, and TableSets**

### **Chapter 6**

#### **Creating Record Definitions**

### **Chapter 7**

#### **Administering Data**

### **Chapter 8**

#### **Designing Pages for Optimal Performance and Accessibility**

### **Chapter 9**

#### **Using Page Controls**

### **Chapter 10**

#### **Using Scroll Areas, Scroll Bars, and Grids**

### **Chapter 11**

#### **Creating Page Definitions**

### **Chapter 12**

#### **Creating Component Definitions**

**Chapter 13**  
**Creating Menu Definitions**

**Chapter 14**  
**Using the Registration Wizard**

**Chapter 15**  
**Creating Style Sheet Definitions**

**Chapter 16**  
**Creating Image Definitions**

**Chapter 17**  
**Creating HTML Definitions**

## Chapter 2

# Using PeopleSoft Application Designer

This chapter provides an overview of PeopleSoft Application Designer and discusses how to:

- Use PeopleSoft Application Designer.
- View PeopleCode.
- Work with definitions.
- Use miscellaneous definitions.
- Use spell check.
- Build and maintain data.

---

## Understanding PeopleSoft Application Designer

Every PeopleSoft Pure Internet Architecture application contains a collection of related definitions that work together for a specific purpose. Developing and adapting PeopleSoft applications is a step-by-step process in which you define and build the definitions, establish relationships among definitions, implement security, run your PeopleSoft application in an internet browser, and test every aspect thoroughly. You use one interactive tool, PeopleSoft Application Designer, for the majority of these activities. PeopleSoft Application Designer is an integrated development environment that enables you to work with the numerous definitions of a business application in a single work area.

---

## Using PeopleSoft Application Designer

This section provides an overview of the eight steps of application development and describes how to:

- Start PeopleSoft Application Designer.
- Use the Application Designer window components.
- Set PeopleTools Options

## Understanding the Eight Steps of Application Development

Application development is a systematic process. Between the design and implementation of the application, you define and build application definitions, establish relationships, implement security, and test every aspect thoroughly. When developing in the PeopleSoft environment, it is helpful to look at the development phase as an eight step process:

1. Design the application.
2. Create field definitions.
3. Create record definitions.
4. Build the application data tables.
5. Create page definitions.
6. Define components.
7. Register the components.
8. Test the application.

### ***Step One — Designing the Application***

Create a development plan. The first phase of development in any environment is always planning. It is not uncommon that 80% of the effort is spent on designing a solid plan before taking the first step. In PeopleSoft, you will need to fully understand the other seven steps of development to do a good job of planning. When you are finished planning, you should have a business requirements document of the tables required and the relationships between them, a list of definitions required (fields, records, and so on) complete with detailed specifications and naming conventions, mock-ups of the page layouts, and the navigational hierarchy to give users access to the pages you create.

### ***Step Two — Creating Field Definitions***

Field definitions in PeopleSoft are slightly different from other environments. Fields are defined apart from any tables they may belong to and can be re-used in any number of separate database tables. The field definition is the basis for all the steps that follow.

### ***Step Three — Creating Record Definitions***

PeopleTools-based applications are table-based systems. When creating a record definition, you are actually defining two layers of information: the record level and the field level. At the record level you determine the structure of the database table and the ultimate purpose of the record definition as well as how it will be used in the system. At the field level, you determine the characteristics of the fields and any online processing that you want to occur behind the scenes when a user enters data.

### ***Step Four — Building the Tables***

Once you define the record definition you must create or *build* the underlying SQL tables that house the application data that your users will enter online in the production environment. This Build process is the centerpiece of the data administration features in PeopleSoft Application Designer. The build process takes all of the properties of a record definition and creates a database table from it. Unless this step is taken, there is no table in the database to accept application data.

### ***Step Five — Creating Page Definitions***

Pages are the user interface for the application where you present the data to the users and enable them to enter data into the application data table created in the previous step. In this step you add page controls to a page and that you associate with record fields and configure the display of the page. The page is presented to the user in a browser window as HTML.

### ***Step Six — Creating Component Definitions***

Component definitions represent complete business transactions in PeopleSoft applications and are central to navigation and security access. A component holds one or more pages that the user navigates between using tabs or links. Here you define the search record and search attributes, processing mode, and toolbar buttons that display at the bottom of each page in the component.

### ***Step Seven — Registering the Component***

The Registration Wizard makes registering a component easy. In this step you accomplish several things: attach the component to a menu, assign security access for users, and place navigational links in the registry. Once you register a component, the user interface is complete and you can view the application in the browser.

### ***Step Eight — Testing the Application***

The final step of application development is to thoroughly test the application before delivering the finished product to users. It is important to test all aspects of the application by opening the transaction in your browser and entering test data into the application. As issues arise you can cycle back through each of these steps to resolve them.

## **Starting PeopleSoft Application Designer**

Open PeopleSoft Application Designer from the Start menu. Or, if you have set up a shortcut on your desktop you can open the application from there. At the PeopleSoft Signon dialog box enter the Database Name or Application Server Name, (depending on your connection type) User ID, and Password. PeopleTools opens automatically to PeopleSoft Application Designer.

---

**Note.** Certain PeopleSoft utilities require setting an environment variable, PS\_SERVER\_CFG, to run properly. However, PeopleSoft Application Designer isn't compatible with PS\_SERVER\_CFG being set. Before you start PeopleSoft Application Designer, you must ensure that PS\_SERVER\_CFG is not set. You can make this convenient by using a DOS batch file to unset PS\_SERVER\_CFG, launch PeopleSoft Application Designer, then after PeopleSoft Application Designer exits, reset PS\_SERVER\_CFG to its previous value.

---

In addition to launching PeopleSoft Application Designer from the Start menu, you can also do so using the command line syntax and the available parameters listed here.

PeopleSoft Application Designer offers a variety of command-line parameters that you can use to control the database to which it connects. By using these parameters, you can automatically navigate to the part of the system that you need.

Use the values from the following tables when specifying certain parameters.

### **Syntax**

The following command line syntax applies to all actions identified in this chapter:

PSIDE [-parameter value [-parameter value . . .]]

Use this command line syntax to start PeopleSoft Data Mover:

PSDMT [-parameter value [-parameter value . . .]]

You can include as many or as few parameters as you need.

Each parameter starts with a hyphen (-) or a forward slash (/). The value for each parameter follows the hyphen or slash, separated by zero or more spaces. In general, the value doesn't need to have quotation marks around it, even if it has internal spaces—the system treats all text following the parameter as part of the value, up to the next parameter or the end of the command line.

---

**Note.** You must enclose a value in quotation marks only when it includes a hyphen or forward slash, or to include leading or trailing spaces. If the value itself includes a quotation mark character, precede the double quote with a backslash (\).

---

### **Available Parameters**

The available parameters for PSIDE.EXE fall into two general categories:

- Parameters providing login information.
- Parameters setting general options.

If the command line includes login parameters, it uses them only if no PeopleSoft applications are currently running. If a PeopleSoft application is already running, the system starts a new instance by using the same login information as the active instance.

This table lists all of the available parameters:

<b>Parameter</b>	<b>Value</b>	<b>Description</b>
-CT	Database type	Enter the type of database to which you are connecting. Valid values are ORACLE, INFORMIX, SYBASE, MICROSOFT, DB2ODBC, and DB2 UDB for Linux, UNIX, and Microsoft (note the spelling of MICROSOFT).

<b><i>Parameter</i></b>	<b><i>Value</i></b>	<b><i>Description</i></b>
-CS	Server name	Enter the name of the database server for the database that you're connecting to. This setting is required for some database types.
-CD	Database name	Enter the name of the database to which you are connecting, as you would enter it into the PeopleSoft Signon dialog box.
-CO	User ID	Enter the PeopleSoft user ID to log in with.
-CP	Password	Enter the password for the specified user ID.
-CI	Connect ID	Enter the ID that is used to connect to the database server.
-CW	Password	Enter the password for the specified connect ID.
-SS	NO	Use to suppress the display of the PeopleSoft splash screen. Any other value causes the splash screen to display.
-SN	NO	Use to suppress the sound that plays when you sign in to the PeopleSoft system. Any other value causes the sound to play.
-QUIET	(none)	Use to run in quiet mode, so that no message boxes appear.
-HIDE	(none)	Use to hide the PeopleSoft Application Designer interface.

### ***Example***

This command line statement starts PeopleSoft Application Designer and signs in to the DEP7TST database:

```
PSIDE -CT MICROSOFT -CS SEPNTDB05 -CD EP7TST -CO VP1 -CP VP1
```

## Using the PeopleSoft Application Designer Window Components

The PeopleSoft Application Designer window includes these six main elements:

<i><b>Element</b></i>	<i><b>Description</b></i>
Title bar	Displays the name of the open project and active definition.
Menu	Provides access to PeopleSoft Application Designer commands and features.
Toolbar	Displays buttons that are useful in editing the active definition.
Project workspace	Provides a graphical representation of the components that are contained in a project. The project workspace has two folder tabs: <ul style="list-style-type: none"> <li>• Development View.</li> <li>• Upgrade View.</li> </ul>
Definition workspace	Displays individual definitions that you open.
Output window	Contains the output text from PeopleSoft Application Designer operations, such as Build (SQL Create and Alter), Find Definition References, Upgrade, Results, Validate, and PeopleCode Log.

---

**Note.** PeopleSoft recommends that you maximize the Application Designer window to make optimal use of the different workspaces.

---

### ***Application Designer Window Menus***

This section reviews the general PeopleSoft Application Designer menus. Specific menus appear for each individual definition when it is open and active. Those menus are covered in the sections about each definition.

---

**Note.** The menu commands that display from each menu often depend on the type of definition that is open in the definition workspace. Those menu items are described in the corresponding chapters.

---

In addition to the standard MicroSoft Windows File menu commands, there are several PeopleTools specific commands. This table lists the PeopleTools related File menu commands:

<b><i>File Menu Commands</i></b>	<b><i>Usage</i></b>
Save Project	Save the project.
Save Project As	Save the active project definition. This enables you to change the name of the project as you save it.
Print Project	Print an Upgrade Project report for all the elements in the open project.
Merge Projects	Create a new project from selected definitions of two existing projects.
Project Properties	Edit properties for the open project.
Definition Properties	Provide general comments and change attributes for the active definition. You can change attributes for Use, Type, and Internet properties depending on the definition type.
View Report	View a report of the definition in an open window in the workspace.
Report from File, Print Preview	View a print preview of a selected report file that is saved to a disk.
Report from File, Print	Print a selected report file from a disk.
Report from File, View Report	View a report from a selected report file that is saved to a disk.
List of last opened definitions	View and open one of the last five opened definitions.

In addition to the standard MicroSoft Windows based commands, such as Cut and Paste, there are two PeopleTools specific commands, as listed in the table below.

<b><i>Edit Menu Commands</i></b>	<b><i>Usage</i></b>
Find Definition References	Search for all related definitions and references (fields, records, pages, PeopleCode, and so on) to the current open definition. The results appear on the Display window: Find Definition References tab.

<b><i>Edit Menu Commands</i></b>	<b><i>Usage</i></b>
Find in	Search through all PeopleCode, SQL programs, HTML definitions, and Free form sub style sheets for a text string that you describe in a dialog box. You can also specify which type of PeopleCode and SQL programs to search.

This table lists View menu commands:

<b><i>View Menu Commands</i></b>	<b><i>Usage</i></b>
View Definition	View a definition that is referenced in the active definition. For example, if you highlight a record field in a record definition, selecting this option opens the selected field definition.
View PeopleCode	View PeopleCode characteristics of the active definition.
Filtering	Select a filter option for your view of upgrade: <ul style="list-style-type: none"> <li>• No Filtering.</li> <li>• Selected for Upgrade Action.</li> <li>• Not Selected for Upgrade Action.</li> <li>• Custom Filtering.</li> </ul>
Toolbar	Activate and deactivate the toolbar.
Status Bar	Activate and deactivate the status bar.
Project Workspace	Activate and deactivate the project workspace display.
Output Window	Activate and deactivate the output window display.

This table lists Insert menu commands:

<b><i>Insert Menu Commands</i></b>	<b><i>Usage</i></b>
Current Definition into Project	Insert the active definition into the project.

<b><i>Insert Menu Commands</i></b>	<b><i>Usage</i></b>
Definitions into Project	Display a dialog box to insert definitions into the project.

<b><i>Build Menu Commands</i></b>	<b><i>Usage</i></b>
Current Definition	Display a dialog box to build or alter the record definition that is currently active in the definition workspace.
Project	Display a dialog box to build (SQL Create and SQL Alter) all records in the project.
Settings	Display a dialog box in which you set parameters that apply to build operations.

This table lists the Debug menu commands:

<b><i>Debug Menu Commands</i></b>	<b><i>Usage</i></b>
PeopleCode Debugger Mode	<p>Start the PeopleCode Debugger, invoking a debugging session and causing a full Debug menu to appear. When in Debugger mode, a Local Variables pop-up window displays in the definition workspace.</p> <p>See <i>Enterprise PeopleTools 8.50 PeopleBook: PeopleCode Developer's Guide</i>, "Debugging Your Application."</p>

This table lists the Tools menu commands:

<b><i>Tools Menu Commands</i></b>	<b><i>Usage</i></b>
Validate Project	Validate attributes of the active project.
Validate Database for WSRP	<p>Validates all of the appropriate definitions within a database for WSRP compliance.</p> <p>See <i>Enterprise PeopleTools 8.50 PeopleBook: PeopleTools Portal Technologies</i>, "Using WSRP to Consume and Produce Remote Portlets," Validating WSRP-Compliance.</p>

<b><i>Tools Menu Commands</i></b>	<b><i>Usage</i></b>
Compile All PeopleCode	Validate all of the PeopleCode in the current database.
Compile Project PeopleCode	Validate the PeopleCode in the current project.
Diff / Merge Project PeopleCode	Compare and merge two PeopleCode definitions during an upgrade.
Diff / Merge External Text	Compare and merge two versions of an external text file (such as a Data Mover script or a SQL script) during an upgrade.
Compare and Report	Compares two projects and reports on the differences.
Copy Project	Copy the current project to a database or a file, or export a project from a file to your database.
Data Administration	Open dialog boxes that describe space allocation in the database.
Change Control	Open dialog boxes to view history or set systemwide change control options as the administrator.
Upgrade	Upgrade an application project.
Translate	Translate base and operator language databases.
Miscellaneous Definitions	<p>Set custom field formats, toolbars, colors, styles, access profiles, and file type codes.</p> <p><b>Important!</b> Styles are a deprecated feature. Set styles using style sheet definition.</p> <p>See <a href="#">Chapter 15, "Creating Style Sheet Definitions," Understanding Style Sheets and Style Classes, page 407.</a></p>
Bulk Operations	Insert, modify, or delete certain definition types across your entire database in one operation.
Options	Select commands specifying insertion commands, definition language preference, validating options, PeopleCode fonts, definition owner ID, and more.

This table lists Window menu commands:

<b>Window Menu Commands</b>	<b>Usage</b>
Cascade	Cascade all of the definitions that are not minimized in the definition workspace into overlapping layers of definitions with the definition names visible.
Tile	Reorganize and resize all of the definitions that are not minimized in the definition workspace so that they are all visible and do not overlap.
Arrange Icons	Neatly group together all of the minimized definitions in the definition workspace.
Close All	Close all of the definitions in the definition workspace.

This table lists Help menu commands:

<b>Help Menu Commands</b>	<b>Usage</b>
Current Window	Provides context-sensitive help that is related to the active window by opening the appropriate PeopleBook in the appropriate location.
PeopleBooks Library	Takes you to the PeopleSoft Online Library (the online documentation for PeopleTools and the PeopleSoft applications that you have licensed).
About PeopleTools	Displays the PeopleTools release level, application release, user ID, database name, database type, and application server name.

### **Application Designer Window Toolbar**

In addition to the standard MicroSoft Windows toolbar button commands, such as cut, paste, and save, there are several PeopleTools specific commands. The command name appears in parentheses following the description.



Open the Definition Properties dialog box (File, Definition Properties).



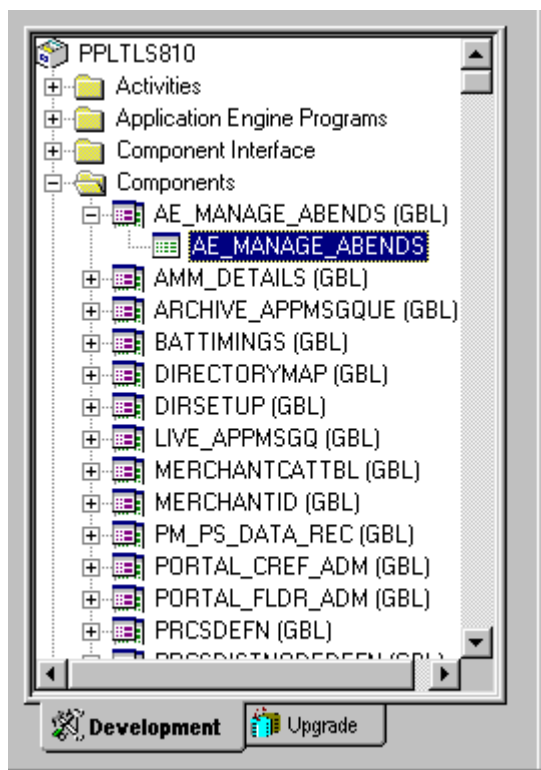
Open the Build dialog box (Build, Current Definition).



Toggle the display of the project workspace (View, Project Workspace).

## Project Workspace

View projects and their associated definitions in the project workspace. A project organizes a collection of definitions in logical groups for easier development, adaptation, maintenance, and upgrade.



Project workspace shown with a project open

Work with a development project or an upgrade project by selecting either the Development tab or the Upgrade tab at the bottom of the project workspace.

The text on the Development and Upgrade tabs might not appear if there is not enough room to display it onscreen.

## Definition Workspace

Use the definition workspace to create and modify definitions, which can be maximized, minimized, cascaded, or tiled in the workspace.

## Output Window

The output window displays status messages of various PeopleSoft Application Designer operations. The output is organized into different categories by using tabs at the bottom of the window.

The text displayed in the output window is context-sensitive, enabling you to select it to perform related operations. For example, if you have a field definition open, select Edit, Find Definition References to list the definitions that reference the active field definition. The list appears in the output window. Then, double-click any of the related definitions listed to open them.

The output window contains several tabs that relate to numerous functions that you can perform with PeopleSoft Application Designer:

<b>Tab</b>	<b>Usage</b>
Build	Create tables, indexes, and views, and alter tables. If you select the <i>Log to Output Window</i> option in the Build settings, then errors, warnings, and informational messages appear here. You can control the level of detail that appears.
Find Definition References	View definitions that are used or referenced by other definitions. Double-click a definition to open it.
Upgrade	View upgrade actions that are initiated from PeopleSoft Application Designer.
Results	View messages related to project operations, like opening a project or inserting definitions into the project.
Validate	Validate projects and components, and view the results.
Find in	View the Find Indialog box output. Double-click a definition to open it. The Find In search can be limited to a specific project, rather than the entire database.
PeopleCode Log	View errors that you receive when you're in the debugger stepping through code.

## Setting PeopleTools Options

You can set preferences for various features in the Options dialog box. Access the Options dialog box by selecting Tools, Options. Each tab controls options for different areas of PeopleSoft Application Designer. Most of these options are described in further detail in other, more relevant areas of the PeopleBooks documentation. The table below provides a general description of each tab and a link to further details about the options:

<b>Project</b>	Set various project options. See <a href="#">Chapter 3, "Working With Projects," Setting Project Options, page 57.</a>
----------------	---

<b>Validate</b>	<p>Use this tab to set project validation options and to activate the spell check feature for PeopleSoft Application Designer.</p> <p>See <a href="#">Chapter 3, "Working With Projects," Validating Projects, page 58.</a></p> <p>See <a href="#">Chapter 2, "Using PeopleSoft Application Designer," Enabling Spell Check, page 46.</a></p>
<b>Editors</b>	<p>Set Word Wrap options for the various editors in PeopleSoft Application Designer.</p> <p>See <i>Enterprise PeopleTools 8.50 PeopleBook: PeopleCode Developer's Guide</i>, "Using the PeopleCode Editor."</p> <p>See <i>Enterprise PeopleTools 8.50 PeopleBook: PeopleCode Developer's Guide</i>, "Using the SQL Editor."</p>
<b>General</b>	<p>Set the Undo/Redo Buffer Depth. This option determines how many layers of editability you have in the PeopleSoft Application Designer. The default value is 10. This means that you can "undo" up to 10 immediately previous actions taken, such as adding or deleting page fields from a page.</p> <p>Set the Maximum Most Recently Used Definition. The most recently used definitions appear at the bottom of the File menu. The default value is 5.</p> <p>Choose the destination you would like for your report output in the Report Output Directory field.</p>
<b>Owner Id</b>	<p>Set the default owner Id for new definitions you create. Alternatively, leave the option blank, the default, to allow users to select the owner Id they wish. The Owner Id helps identify which applications team last made a change to a definition.</p>
<b>Image</b>	<p>Set the preferred storage format for newly created image definitions.</p> <p>See <a href="#">Chapter 16, "Creating Image Definitions," Specifying the Image Storage Format, page 454.</a></p>
<b>Browser</b>	<p>Enter here the base URI to be used by the browser to locate the application and the node name.</p> <p>See <a href="#">Chapter 2, "Using PeopleSoft Application Designer," Configuring Browser Options, page 46.</a></p>

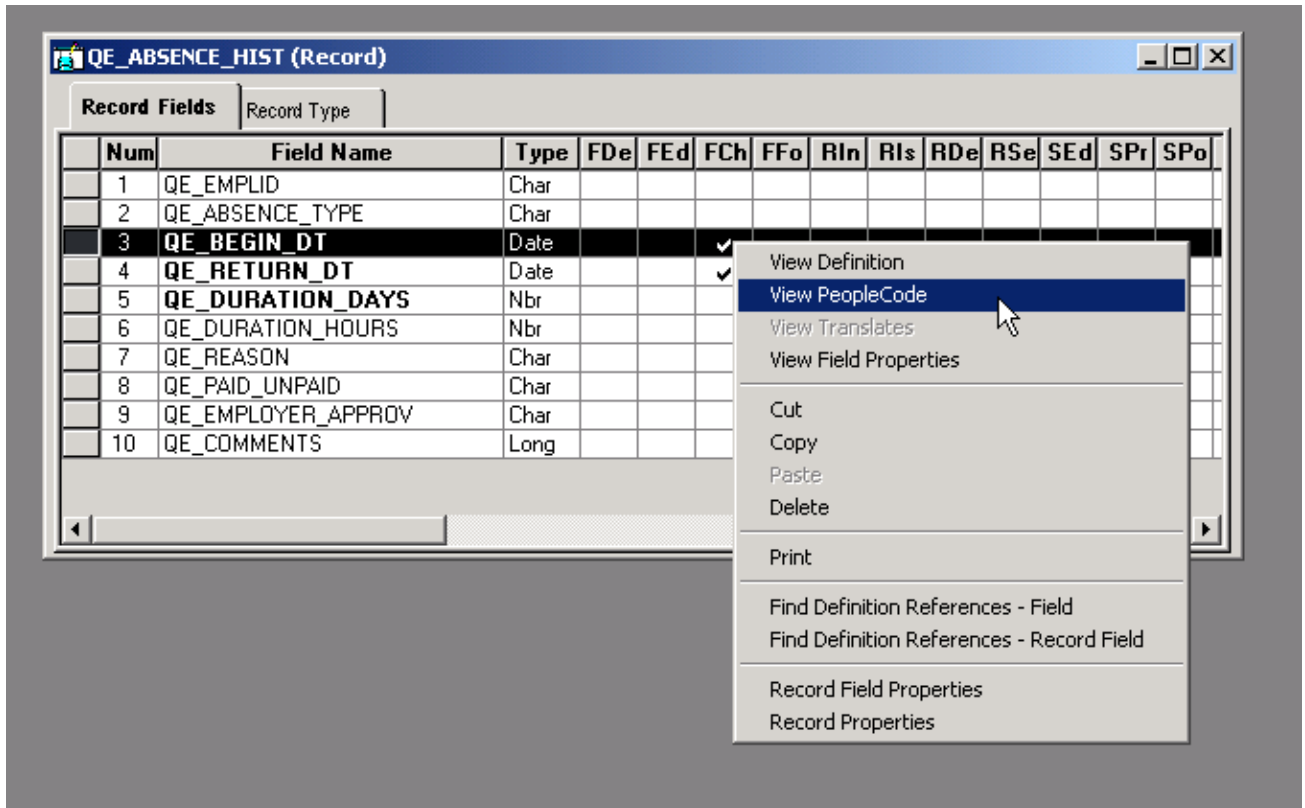
---

## Working with PeopleCode

This section provides a brief explanation of working with PeopleCode and discusses the main development editors.

PeopleCode is the structured programming language built into PeopleTools that extends the functionality of the PeopleTools environment. All PeopleCode programs are associated with a parent definition. These PeopleCode programs are considered part of the definitions of their parent components and you edit them as part of the definitions.

PeopleCode programs can be associated with many items, such as record fields, pages, components, and application messages. You can access any PeopleCode program from its related definition. As an example, one of the ways to access PeopleCode from one of the many places that PeopleCode is stored is to access Record Field PeopleCode from an open record definition.



Accessing FieldChange PeopleCode from the BEGIN\_DT field

To view Record Field PeopleCode from PeopleSoft Application Designer:

1. Open a record definition in PeopleSoft Application Designer.
2. Click the View PeopleCode toolbar button to view the PeopleCode display for the record.
3. Select a field containing PeopleCode in the record definition.
4. Double-click the cell under the column heading for the type of PeopleCode that you want to view.

You can also right-click on a definition and use the pop-up menu to access or view the PeopleCode.

### See Also

*Enterprise PeopleTools 8.50 PeopleBook: PeopleCode Developer's Guide*, "Accessing PeopleCode and Events"

## Using the Development Editors

This section describes the three main development editors:

- SQL editor
- PeopleCode editor
- Application Packages editor

### ***SQL Editor***

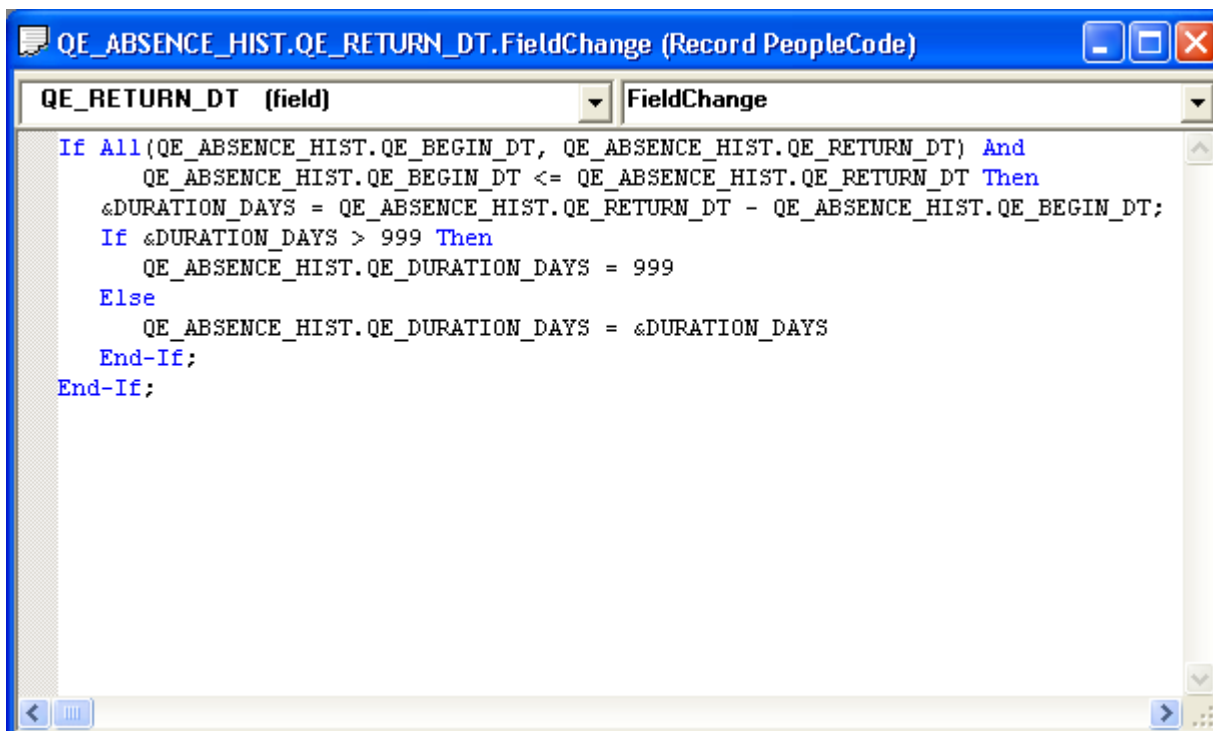
The SQL Editor is a tool that enables you to construct SQL definitions. SQL Editor has a similar look and feel to the PeopleCode editor. You can access SQL Editor from these definitions:

- Records that are based on SQL and dynamic view.
- PeopleSoft Application Engine actions.
- PeopleCode editor.

### ***PeopleCode Editor***

Any method of accessing PeopleCode programs displays a window where PeopleCode programs can be input or modified. This window is the PeopleCode editor. The PeopleCode editor enables you to edit and navigate all PeopleCode programs that belong to the same parent definition.

On the left side of the window is a drop-down list box showing the current field and other available fields, records, and components. Any field with PeopleCode attached is listed in bold. The right side of the window has a drop-down list box listing the current PeopleCode event and other acceptable events for that definition. Events already containing PeopleCode for that definition are listed in bold.



Working in the PeopleCode editor

### Application Packages Editor

From an Application Package you can access the PeopleCode programs that are associated with the classes of the package. The Application Packages editor and the PeopleCode editor interfaces are similar. You can add, delete, and change text; you can use the find and replace function; you can validate the syntax. When you save your application package, the code is automatically formatted (indented and so on), just as it is in the PeopleCode editor.

#### See Also

*Enterprise PeopleTools 8.50 PeopleBook: PeopleCode Developer's Guide*, "Using the SQL Editor"

*Enterprise PeopleTools 8.50 PeopleBook: PeopleCode Developer's Guide*, "Using the PeopleCode Editor"

*Enterprise PeopleTools 8.50 PeopleBook: PeopleCode Developer's Guide*, "Creating Application Packages and Classes"

## Working with Definitions

This section provides an overview of definitions and discusses how to:

- Open definitions.
- View definitions.

- View and edit definition properties.
- Create definitions.
- Save definitions.
- Close definitions.
- Rename definitions.
- Delete definitions.
- Find definitions.

## Understanding Definitions

You can create or modify many types of definitions in PeopleSoft Application Designer. This table lists development definitions:

<b>Definition</b>	<b>Description</b>
Activity	Workflow map showing the individual steps, events, and routings that comprise a complete activity in a business process.
Analytic Model	Analytic models are used exclusively with the PeopleSoft Analytic Calculation Engine.  <i>See Enterprise PeopleTools 8.50 PeopleBook: Analytic Calculation Engine, "Designing and Editing Analytic Models."</i>
Application Engine	Program comprising SQL statements and PeopleCode programs to run in batch mode as an alternative to using COBOL or SQR programs.
Application Package	Used to create Application Classes, which could be used to either extend the functionality of existing PeopleTools classes (Rowset, Array, and so on) or to create your own classes.
Approval Rule Set	Workflow maps that provide a visual representation of approval rules. Virtual Approver reads approval rule sets to determine who must approve transactions.
Business Interlink	Provides a gateway between PeopleSoft applications and the services of an external system.

<b>Definition</b>	<b>Description</b>
Business Process	Workflow maps that provide a visual overview of the activities involved in a procedure.
Component	Represents a logical business transaction or a set of logically related pages that are processed together.
Component Interface	Externalizes access to a component so that it can be used by a third party or an application message.
Field	Individual pieces of data—such as an employee ID—that can be entered by the user and stored in the database, in a column on a table, or in a view.
File Layout	Definition (or mapping) of a file to be processed. It identifies where data fields are located in a file.
HTML	Definition that represents static HTML that can be included on a page definition.
Image	Stores and displays images, such as employee photos, product pictures, and so on.
Menu	Enables access to the components that you build and the pages contained in the components.
Message	Based on a multilevel structure, similar to components, that defines the data to insert into the application message at runtime.
Message Channel	These correspond to groups of message definitions and help order messages properly, enhance scalability, and provide a simple way to define processing characteristics of many similar messages as a single group.
Mobile Page	A PeopleSoft Pure Internet Architecture page based on a synchronizable component interface for display on disconnected mobile devices, such as laptops or PDAs.
Page	Pages provide a way to enter, view, and edit data online. The system validates user input, writes it to the database, and retrieves and displays it upon request.

<b>Definition</b>	<b>Description</b>
Project	User-defined collection of related definitions created for developing, adapting, or upgrading a PeopleSoft application.
Problem Type	Used in optimization to tie the optimization application records, the optimization transactions, and the optimization plug-in together as one entity.
Record	All of the data that resides in PeopleSoft applications is stored in tables, or records, as part of a relational database system. Each record definition describes the properties of an underlying SQL table.
SQL	Can be entire SQL programs or fragments of SQL statements that you want to reuse.
Style Sheet	Collection of styles that can be used by PeopleSoft Pure Internet Architecture application pages.

## Opening Definitions

Use the definition workspace when accessing or creating definitions.

To open a definition:

1. Select File, Open or click the corresponding toolbar button.

The Open Definition dialog box appears.

2. Select the type of definition that you want to open from the Definition drop-down list box.
3. Provide selection criteria.

Enter a definition name or description (or the beginning characters of either). Other selection criteria options might be available, depending on the definition type.

4. Click Open, or press ENTER, to display definitions matching the selection criteria that you entered.

To clear the current selection criteria and start over, click New Search. To change how the search list is displayed, perform one of these actions:

- Click the List button to see only the names of the definitions.
- Click the Details button to see the names and descriptions, which is the default.

5. Select the definition to open.

6. Double-click the definition that you want to open in the definition workspace, or highlight the definition and click Open.

You can also use the shift-click selection technique to select more than one definition to open in a single action, or right-click to view a pop-up menu, where you can open, print, rename, or delete the selected definition.

---

**Note.** The print menu is disabled when you right-click any of the following definitions: activity, approval rule set, business process, component interface, component, project, and mobile page.

---

---

**Note.** When the definition is open and able to be modified, "Ready" appears in the lower left-hand corner of the workspace.

---

## Viewing Definitions

In the project workspace, you can view a definition that is referenced by the active definition.

To view a definition, select View, View Definition. You can also right-click the definition to be referenced and select View Definition from the pop-up menu.

## Viewing and Editing Definition Properties

Each definition type has properties that are associated with it. To view or edit definition properties, select File, Definition Properties. You can also click the toolbar button or press Alt+Enter, or right-click on the definition and select Properties from the pop-up menu.

The Definition Properties dialog box almost always includes a General tab that provides a place to display and enter descriptions and comments regarding the definition. An optional Owner ID drop-down list box is provided to designate the owner of an individual definition. This helps identify which application team last made a change to a definition. Set the default owner ID for new definitions in the Options dialog box, Owner Id tab.

**Record Properties**

General Use

AMM\_DERIVED

Description: App Msg Monitor work record

Record Definition:  
This derived record is used by the Application Message Monitor.

Owner ID: PeopleTools

Last Updated  
Date/Time: 04/11/2000 4:53:51PM  
By Operator: PTDMO

OK Cancel

Property sheet example

With PeopleSoft Application Designer, every definition that you can open (menus, pages, components, records, fields, and so on) has an associated property sheet. Definition properties that are specific to the definition type are grouped together in these definition property sheets. For example:

- Record Type and Record Use properties are both found in the Definition Properties sheet for records.
- Translates are found in the Definition Properties sheet for fields.
- Menu groups and menu labels are found in the Definition Properties sheet for menus.
- Search record, actions, and internet settings are found in the Definition Properties sheet for components.

## Creating Definitions

When building or adapting an application, try to use existing definitions as much as possible; when that's not possible, create new definitions.

To create a new definition:

1. Select File, New or click the corresponding toolbar button.
2. Select the definition type.

The new definition appears in the definition workspace.

## Saving Definitions

To retain the changes that you make, you must save the definition. However, you must name new definitions before you can save them.

To save a definition:

1. Click the Save button on the toolbar, or select either File, Save or File, Save As to save a definition.

The Save and Save As commands save a single definition. Use the File, Save All command or the Save All toolbar button to save all open definitions at once.

2. If you're saving a page or component, specify the language or market.

For a page definition, the Language drop-down list box appears; for a component, the Market drop-down list box appears.

### **Definition Naming Conventions**

When naming definitions, use the following guidelines:

- Use only uppercase alphabetic characters (A–Z), numeric characters (0–9), or underscore (\_). Other special characters, such as the dash (—), dollar sign, number sign, question mark, and so on, are not permitted.
- Names must begin with an alphabetic character.
- Embedded spaces are not permitted.

Instead, use an underscore.

- Avoid using accented characters.

Names containing these characters may not sort consistently, depending on workstation or locale settings, which results in problems during an upgrade.

- System definitions such as records, fields, styles, and so on, generally have a name and a label or description. Generally, it is recommended to use US-ASCII characters only for the definition name. Labels or descriptions, which are used for the display to the user, can contain non-ASCII characters.

---

**Note.** Existing Business Process and Business Process Map definitions may already contain mixed case characters. Nonetheless, newly created definition names must be all uppercase.

---

## Closing Definitions

To close a definition, either select File, Close or click the Close Window icon on the title bar of the active definition's window. You can also select Window, Close All to close all open definitions.

## Renaming Definitions

You can rename existing definitions in a database at any time.

Name	Type	Description
ACL_APPDES_MISC	Secondar...	Miscellaneous Permissions
ACL_APPDES_OBJECTS	Secondar...	Object Permissions
ACL_APPDES_TOOLS	Secondar...	Tools Permission
ACL_COMPONENT2	Secondar...	Component Permissions
ACL_COMP_INTERFACE	Standard ...	Component Interface Permission
ACL_GENERAL	Standard ...	Permission List General
ACL_ID_GENERAL_SBP	Subpage ...	
ACL_ID_SBP	Subpage ...	
ACL_MENU2	Standard ...	Permission Lists Menu
ACL_MESSAGEMONITOR	Standard ...	Permission List Message Mon
ACL_MISCTOOLS	Standard ...	Permission List Misc Tools

Rename Definition dialog box

To rename a definition in the database:

1. Select File, Rename.

The Rename Definition dialog box looks like the Open Definition dialog box, and you select definitions in the same way.

2. Specify the definition type and any selection criteria that are useful in locating the definition that you want to rename.

The Rename Definition dialog box does not have a Select button. So, after entering either a partial name or description of the definition, press Enter to retrieve a list of definitions matching your selection criteria.

3. Select the definition that you want to rename.

4. Click the Rename button in the dialog box and double-click the selected definition, or right -click to select Rename from the pop-up menu.

The selected definition name in the Rename Definition dialog box is in edit mode.

5. Change the selected definition name and press Enter.

You are prompted to close any open definitions, if any are open, before you can save the new name. You are also prompted with a Confirm Rename dialog box before you can save the change.

6. Click OK to complete the renaming of the selected definition.

## Deleting Definitions

Deleting a definition is different from removing a definition from a project. When you remove a definition from a project, the reference to the definition is removed from the project, but the definition still exists in the database and thus may be a part of any other project. Deleting a definition, however, permanently removes the definition from the database. Any projects or other definitions that contains the deleted definition are affected.

Before deleting a definition you should find all references to that definition by running the Find Definition References process. For example, a field definition that you want to delete may be referenced by both record and page definitions.

---

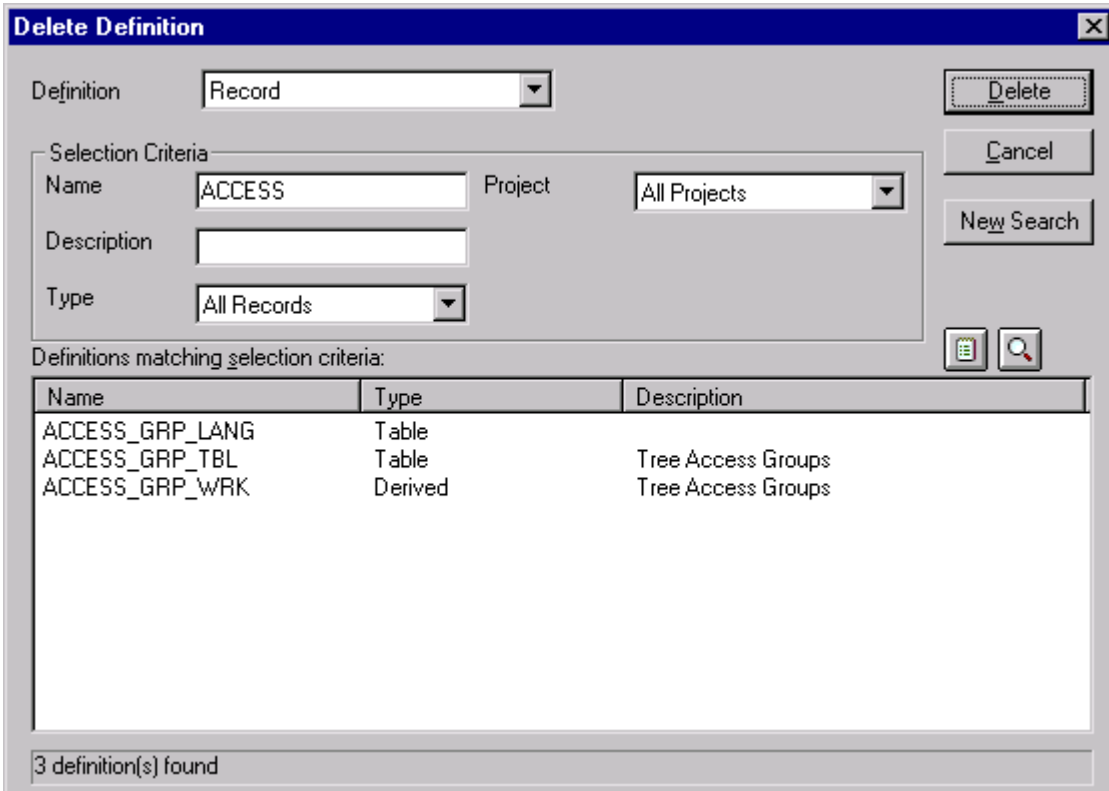
**Note.** If you delete a component definition, you must delete all component interfaces that are based on that definition.

---

---

**Note.** A component definition must contain at least one page. If you attempt to delete a page definition that is the only page definition within a component, the system does not allow the delete to occur and displays an error message. If the page definition is the only page within a component, the error message appears during a File, Delete or during an upgrade copy to database (with a Delete action).

---



The dialog box is titled "Delete Definition". It contains a "Definition" dropdown menu set to "Record". Below this is a "Selection Criteria" section with fields for "Name" (containing "ACCESS"), "Project" (containing "All Projects"), "Description" (empty), and "Type" (containing "All Records"). To the right of these fields are buttons for "Delete", "Cancel", and "New Search". Below the selection criteria is a table titled "Definitions matching selection criteria:". The table has three columns: "Name", "Type", and "Description". It lists three definitions: "ACCESS\_GRP\_LANG" (Table), "ACCESS\_GRP\_TBL" (Table), and "ACCESS\_GRP\_WRK" (Derived). At the bottom of the dialog, it says "3 definition(s) found".

Name	Type	Description
ACCESS_GRP_LANG	Table	
ACCESS_GRP_TBL	Table	Tree Access Groups
ACCESS_GRP_WRK	Derived	Tree Access Groups

Delete Definition dialog box

To delete a definition from the database:

1. Select File, Delete.

The Delete Definition dialog box looks like the Open Definition dialog box, and you select definitions in exactly the same way.

2. Specify the definition type and any selection criteria that are useful in locating the definition that you want to delete.

After entering either a partial name or description of the definition, press Enter to retrieve a list of definitions matching your selection criteria.

3. Select the definition that you want to delete.

To select multiple definitions, hold down either Shift or Ctrl while clicking the appropriate definitions.

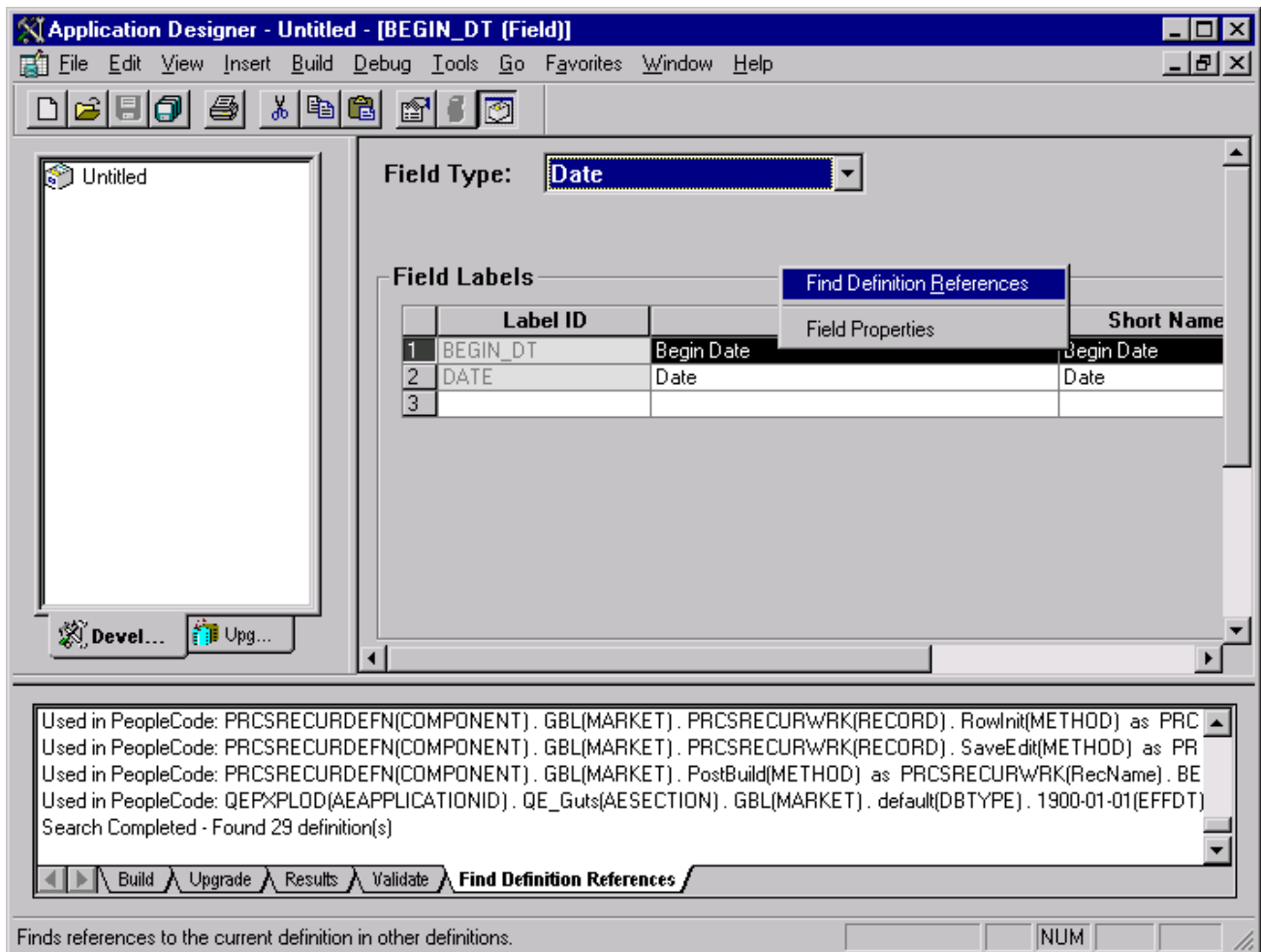
4. Click the Delete button and confirm that you want to delete the selected definition.

## Finding Definitions

Before making significant changes, such as deleting a definition, it is critical to see what other parts of the application may be impacted by this change. The Find Definition References feature performs a comprehensive search of the database pinpointing all definitions in which the changed definition is located.

Although a project keeps track of your definitions, the definitions are not actually embedded in the project.

**Note.** There is a distinction between *related* definitions and *referenced* definitions. A related definition is one that *is used by* the current definition. A definition reference is one that *uses* the current definition. For example, in the case of a component, pages in the component are related definitions. The menus that use the component are its definition references.



### Finding definition references

To find definition references:

1. Open the definition in the definition workspace.
2. Select Edit, Find Definition References, or right-click the definition and select Find Definition References from the pop-up menu.

After you select this item, a search of the database takes place, and the results appear on the Find Definition References tab of the output window.

3. Select any definition that is displayed in the output window by double-clicking it.

The selected definition opens in the definition workspace.

- After you find a definition, select the definition reference and right-click to view the pop-up menu.

**Note.** Double-clicking any entry in the Find Definition References output window opens the appropriate mode of PeopleSoft Application Designer. If you double-click a related record that was found, the record opens, and the field for which you were looking is highlighted.

## Using the Multiple Document Interface

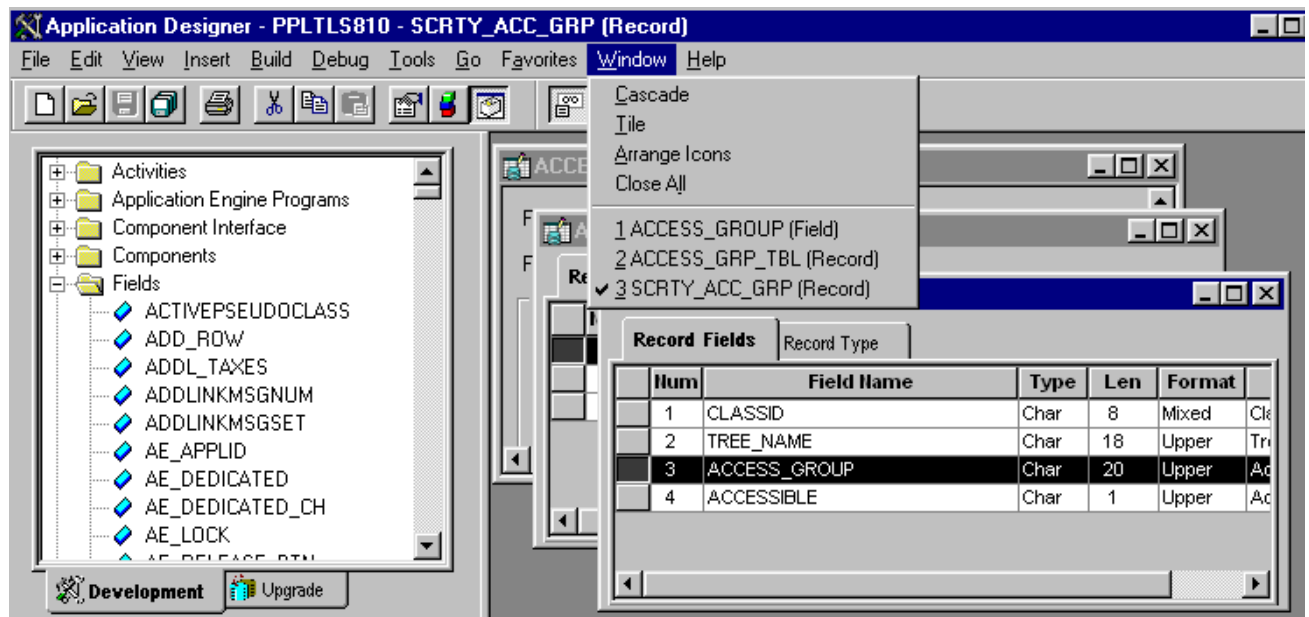
PeopleSoft Application Designer makes use of MDI. This means that each window in the definition workspace of PeopleSoft Application Designer represents a view of a definition, and you can have multiple windows open at the same time. These definition windows must stay inside the borders of the definition workspace, but they can be maximized, minimized, cascaded, or tiled in that space.

You can have different types of definition windows open at the same time, such as menu, field, record, and page windows. This enables you to work with different definitions simultaneously, simplifying the process of designing a group of related definitions.

### Active Windows

Although you can have multiple windows open at the same time, only one window can be active at any moment. Make a window active by selecting it in the definition workspace. The active window always appears on top with the title bar in *active window title* color.

In addition to clicking on a window to make it active, you can also use the Window menu. This menu displays the windows that are currently open with a check mark next to the active one. Change the window that is currently active by selecting it from the menu.



Window menu

Other commands from the Window menu include Cascade, Tile, Arrange Icons, or Close All windows.

## Using Drag-and-Drop

PeopleSoft Application Designer makes ample use of drag-and-drop mouse techniques. For example, you can drag a field from a record definition and drop it onto a page definition. You can then drag the page onto a component and drag the component onto a menu. Using the drag-and-drop technique is faster and simpler than using menu commands to manipulate definitions.

To drag and drop a definition:

1. Open the definition in either the project workspace or the definition workspace from which you want to drag.

For example, to drag a field from an existing record definition to a new page definition, first open the record containing the field. You can open it in the definition workspace or expand the record in the project workspace so that the fields are displayed. With the appropriate field displayed, you can drag and drop it into another definition.

2. Open the definition to which you want to drag.
3. Drag the definition from one definition to the other.

## Using Dynamic Toolbars and Menus

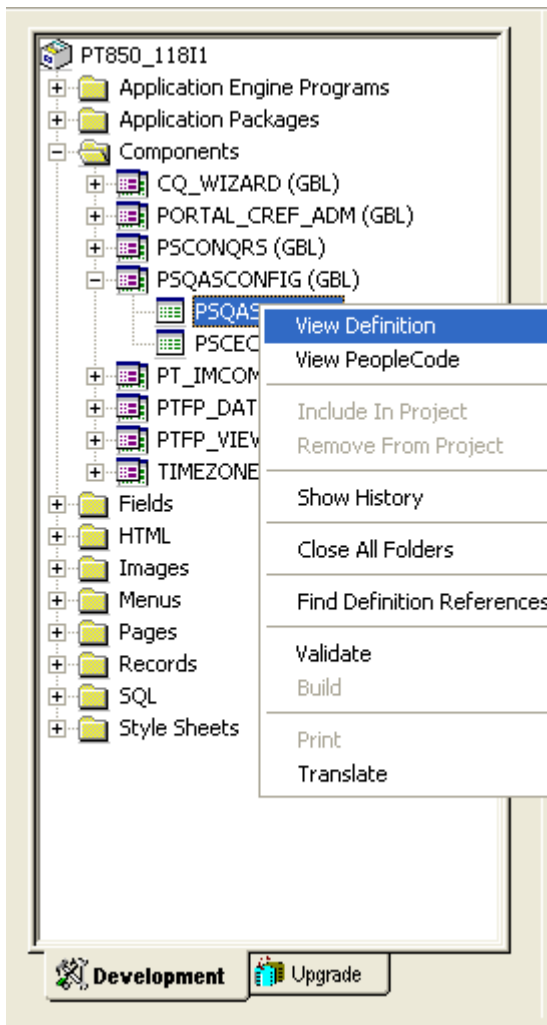
The PeopleSoft Application Designer toolbar and menu changes based on the type of definition that is active. For example, when a page definition is active, the toolbar displays buttons that represent the definitions that you can add to a page. This technique is also known as *morphing*.

The change is so subtle that it can be missed easily. This is because the menu bar names are consistent between definition types. However, if you look at the menu items and toolbars closely, you see a distinct difference. For example, if a page window is active, the menus and toolbars change to include actions and options applicable to a page definition—as in the Show Grid option on the View menu and the introduction of the Layout menu. In addition, page-specific toolbars are also displayed.

In the same respect, if a record window is active, the menus and toolbars morph to include actions and options applicable to a record definition. For example, different options are available on the View menu (such as View, Field Display, and so on). Different toolbar buttons for record definition actions also appear.

## Using Pop-up Menus

Pop-up menus are context-sensitive menus that appear at the current location of the mouse pointer whenever you click the alternate mouse button (typically, the right mouse button). They provide efficient access to numerous PeopleSoft Application Designer commands. Because pop-up menus appear at the current location of the pointer, they eliminate the need to move the pointer up to the menu bar or to a toolbar. A pop-up menu contains only the commands that apply to the selected definition or the current context.



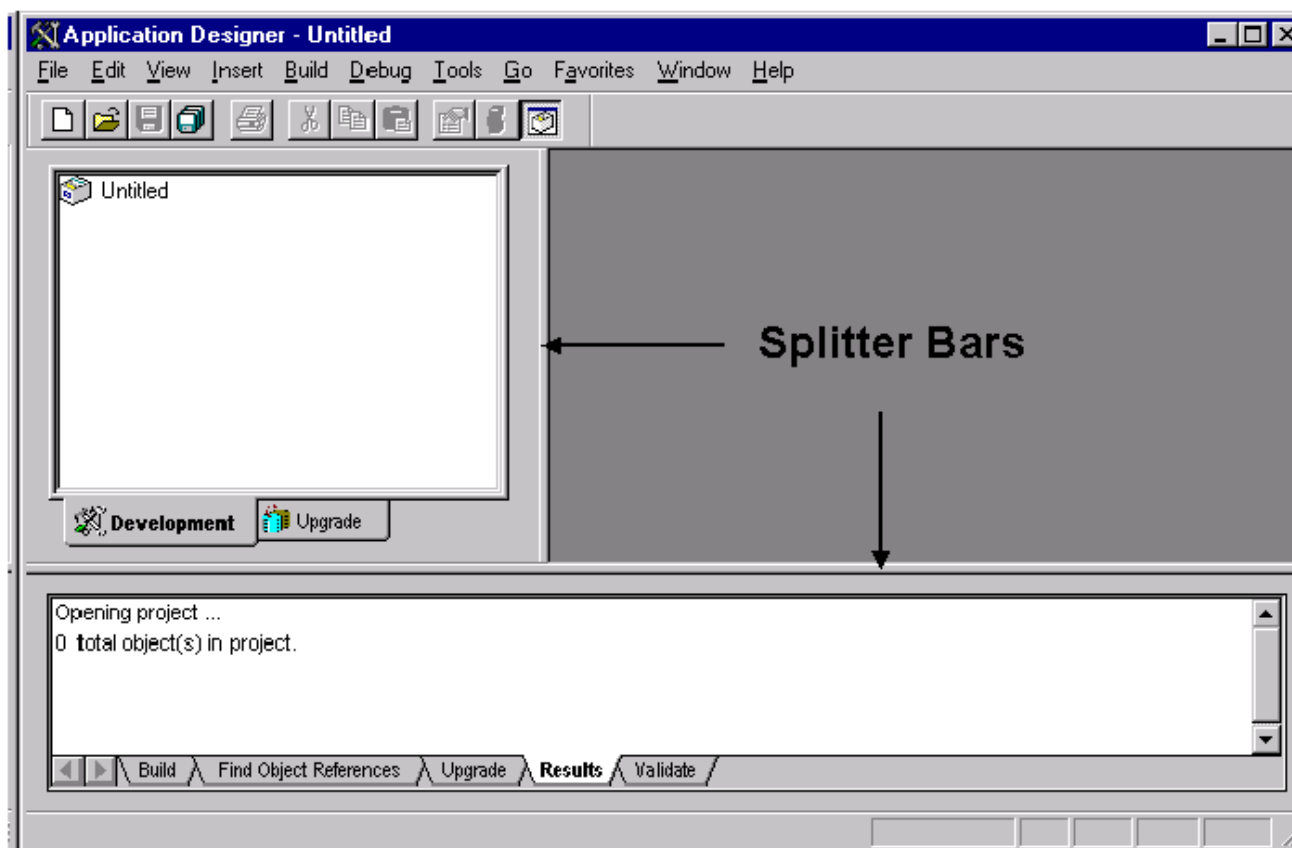
Component definition pop-up menu

## Configuring the Environment

PeopleSoft Application Designer is highly configurable, enabling you to tailor the tool to your needs or preferences.

### ***Splitter Bars***

A splitter bar is a thin vertical or horizontal bar that separates windows inside a parent window. Using the splitter bar enables you to resize two windows at the same time. Splitter bars are used between the project workspace, definition workspace, and output window.



Splitter bars used to resize workspace easily

Drag the splitter bar to change the size of the project workspace, definition workspace, or output window.

### ***Dockable Windows and Toolbars***

You can freely move PeopleSoft Application Designer windows and toolbars to either anchor (or dock) to parts of the main application window, or to make them float freely anywhere that you want them to appear. When you dock a toolbar or window, you anchor it to the top, sides, or bottom of the main window.

To move or dock a PeopleSoft Application Designer window or toolbar, drag the window title bar or the appropriate toolbar to a new location. If you move the window or toolbar near the top, bottom, or sides of the main window, the window or toolbar snaps into place, meaning that it is docked. However, if you press and hold Ctrl and then drag the window or toolbar, you can prevent it from automatically docking when it is near the edge of the window.

---

**Note.** Use caution when running the PeopleCode Debugger with undocked windows.

---

### ***View Menu***

The View menu enables you to display or hide different PeopleSoft Application Designer components (such as the project workspace, output window, and toolbar).

## Working with Generated Files

When working with PeopleSoft Application Designer, you often generate scripts, logs, reports, and so on. When specifying the full path for a file reference in PeopleSoft Application Designer, there is a 100-character (Unicode) limit. This limit applies to all file references, which include build scripts, build logs, compare reports, and so on. For example, a reference to a build log, such as `c:\temp\...\psbuild.log`, can't exceed 100 characters.

---

## Using Miscellaneous Definitions

There are several definition types that are different from the standard definitions described above. These definitions control your work environment or have an impact on the other definitions you create. These miscellaneous definitions are found under the Tools menu and include:

- Field Formats.
- System Colors.
- Styles.
- Access Profiles.
- File Type Codes.

### ***Field Formats***

PeopleTools has formatting provisions at both the field and page levels. The field format specifications affect internal values that are stored in the database, and the page formats affect the visual presentation of values. There are several formatting options for U.S. Social Security Numbers, Canadian Social Insurance Numbers, postal codes, telephone numbers, and various currencies, dates, and times.

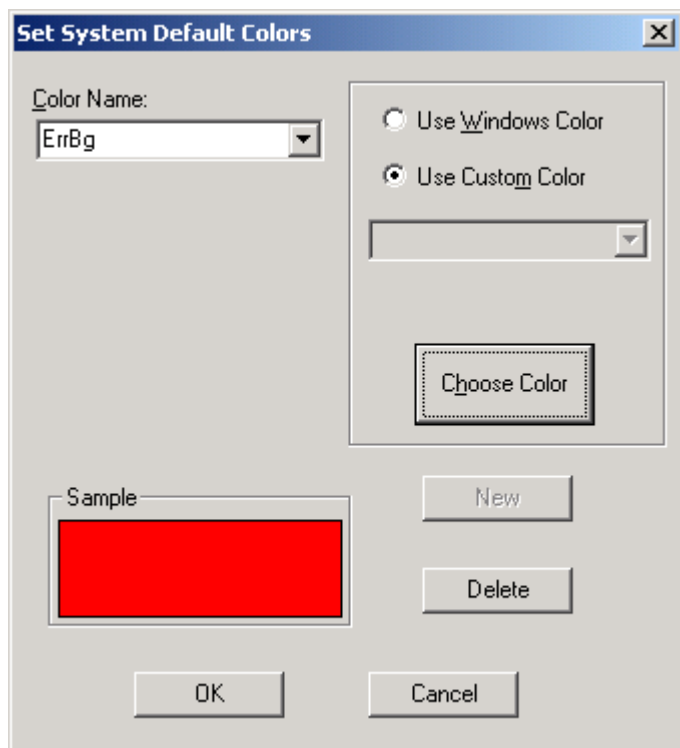
Custom field formats extend these formatting provisions to support the same types of data in other countries. They also support other types of formatted information, such as bank codes, credit card numbers, part numbers, or serial numbers.

Custom field formats enable you to create your own format definitions using format notation and apply them to fields. These formats are organized into format families, which can include one or more unique formats.

See [Chapter 4, "Creating Field Definitions," Changing Format Families, page 83](#).

### ***System Colors***

You can establish definitions for colors you use in your applications. You define colors by associating a color name either with an existing color in the PeopleSoft Internet Architecture or with a custom color that you define. Since taste in color is subjective, PeopleTools allows individual operators to customize their color settings. Users can use the default color definitions you establish in PeopleSoft Application Designer, or they can override them with their own color definitions.



Setting system default colors

To modify the default system colors:

1. Select Tools, Miscellaneous Definitions, System Colors.
2. Select the Color Name you want to modify.
3. To change the color, click the Choose Color button.
4. Choose from the available colors or click Define Custom Colors to create your own color.
5. Click OK.

The new color displays in the Sample area of the Set System Default Colors dialog box.

To create a new system color:

1. Type a new name in the Color Name field and click the New button.
2. Select the Use Custom Color radio button to select a custom color.
3. Click the Choose Color button.
4. Choose from the available colors or click the Define Custom Colors button.
5. Click OK in both dialog boxes to accept the new color.

## Styles

This is a deprecated feature. Styles are set by using style sheet definitions.

See [Chapter 15, "Creating Style Sheet Definitions," Understanding Style Sheets and Style Classes](#), page 407.

### **Access Profiles**

Access profiles are the security profiles users use to connect to the PeopleSoft database. Detailed information about access profiles is discussed in the Security Administration PeopleBook.

See *Enterprise PeopleTools 8.50 PeopleBook: Security Administration*, "Administering User Profiles," Setting Up Access Profiles.

### **File Type Codes**

You may need to set up file type codes when creating change projects. You create change projects to work with PeopleSoft Application Designer's Change Packager tool to package system customizations in preparation for an upgrade. The file type code stores generic information that is applicable to a group of files within the same target directory.

See *Enterprise PeopleTools 8.50 PeopleBook: Change Assistant*, "Working with Change Packages."

---

## **Using Spell Check**

This section provides an overview of the spell check feature in PeopleSoft Application Designer and discusses how to:

- Enable spell check.
- Configure browser options.
- Run spell check on a project.

## **Understanding Spell Check**

The PeopleSoft Application Designer spell check feature checks the spelling for user-facing labels and text in projects and legacy data. Use this feature prior to building your application. Spell check verifies spelling on these items:

- Message Catalog entries.
- Page definitions (label text).
- XLAT tables.
- Component definitions (tab and menu labels).
- Portal registry (labels and attribute labels).
- Record field labels.

## Spell Check Results

The system logs spell check results to the Spell Check tab in the PeopleSoft Application Designer output window. The text "Begin Spell Check" is always the first line on the tab. The first line for each spell check category indicates the total number of entries that were checked for that project. The last line in the output window indicates the total misspelled words encountered during the check.

For each spell check category the results may read differently. This table describes how to read the spell check results for each category:

<b>Output by Error Type</b>	
<p>Message Catalog:</p> <p>Message Set www Nbr xxx (yyy) - zzz</p> <p>For example: Message Set 1 Nbr 1 (Message Text) - Distributed Object Mgr: Xelp Xme=Xangage=abc rtz</p>	<p>Where:</p> <p>www = Message set number.</p> <p>xxx = Message number within the message set.</p> <p>yyy = :</p> <ul style="list-style-type: none"> <li>• Message text, indicating that zzz is the complete message text.</li> <li>• Explain text, indicating that zzz is the complete Explain Text.</li> <li>• Misspelling, indicating that zzz is a misspelled word in either message text or explain text.</li> </ul>
<p>XLAT:</p> <p>[vvv.www.xxx] (yyy) - zzz</p>	<p>Where:</p> <p>vvv = Database field name.</p> <p>www = Field value.</p> <p>xxx = Effective date.</p> <p>yyy =:</p> <ul style="list-style-type: none"> <li>• Long name, indicating that zzz is the complete text for the long name.</li> <li>• Short name, indicating that zzz is the complete text for the short name</li> <li>• Misspelling, indicating that zzz is a misspelled word in either a long or short name.</li> </ul>

<b>Output by Error Type</b>	
<p>Component:</p> <p>[vvv.www] Item Name xxx (yyy) - zzz</p>	<p>Where:</p> <p>vvv = Component name.</p> <p>www = Market code.</p> <p>xxx = Item name for the component.</p> <p>yyy =:</p> <ul style="list-style-type: none"> <li>• Item label, indicating that zzz is the complete text for the component's item label.</li> <li>• Folder tab label, indicating that zzz is the complete text for the component's folder tab label.</li> <li>• Misspelling, indicating that zzz is a misspelled word in either item label or folder tab label.</li> </ul>
<p>Portal Registry:</p> <p>[ttt.uuu.vvv] www xxx (yyy) - zzz</p>	<p>Where:</p> <p>ttt = Portal name.</p> <p>uuu = C for content reference, F for folder.</p> <p>vvv = Portal object name.</p> <p>www =:</p> <ul style="list-style-type: none"> <li>• Empty.</li> <li>• Attribute.</li> <li>• System attribute.</li> </ul> <p>xxx =:</p> <ul style="list-style-type: none"> <li>• Empty if www is empty.</li> <li>• Attribute name if www is attribute.</li> <li>• System attribute name if www is system attribute.</li> </ul> <p>yyy =:</p> <ul style="list-style-type: none"> <li>• Label name, indicating that zzz is the complete text for the Label Name.</li> <li>• Display name, indicating that zzz is the complete text for the display name of an attribute.</li> <li>• Misspelling, indicating that zzz is a misspelled word in either label name or display name.</li> </ul>

<b>Output by Error Type</b>	
Field Label: [www.xxx] (yyy) - zzz	Where: www = Field name. xxx = Field label name. yyy = <ul style="list-style-type: none"> <li>• Long name, indicating that zzz is the complete text for the long name.</li> <li>• Short name, indicating that zzz is the complete text for the short name</li> <li>• Misspelling, indicating that zzz is a misspelled word in either a long or short name.</li> </ul>
Page: [www] Field ID xxx (yyy) - zzz	Where: www = Page name. xxx = Field ID within the page. yyy = <ul style="list-style-type: none"> <li>• Label text, indicating that zzz is the complete text for the field's label text.</li> <li>• Scroll area, indicating that zzz is the complete text for label text within a scroll area.</li> <li>• Misspelling, indicating that zzz is a misspelled word in the label text.</li> </ul>

You can launch an identified definition from the output window directly by double clicking the error line. PeopleSoft Application Designer launches the appropriate content to correct the misspelling:

<b>Error Type</b>	<b>Launches</b>
Message Catalog	n/a
XLAT	Field definition.
Component	Component definition.
Portal registry	Browser to display the Content or Folder Administration page.
Field label	Field definition.
Page	Page definition.

## Enabling Spell Check

The spell check feature is turned off by default. You enable spell check in the Options dialog box, Validate tab. Enabling the spell check option ensures that spell check is run on the designated project when you validate a project.

To enable spell check in PeopleSoft Application Designer:

1. Select Tools, Options from the PeopleSoft Application Designer menu.
2. Select the Validate tab.
3. Select the Validate Spelling check box.
4. (Optional) Select the Log to file check box if you want the spell check results to be saved to a file on your system.

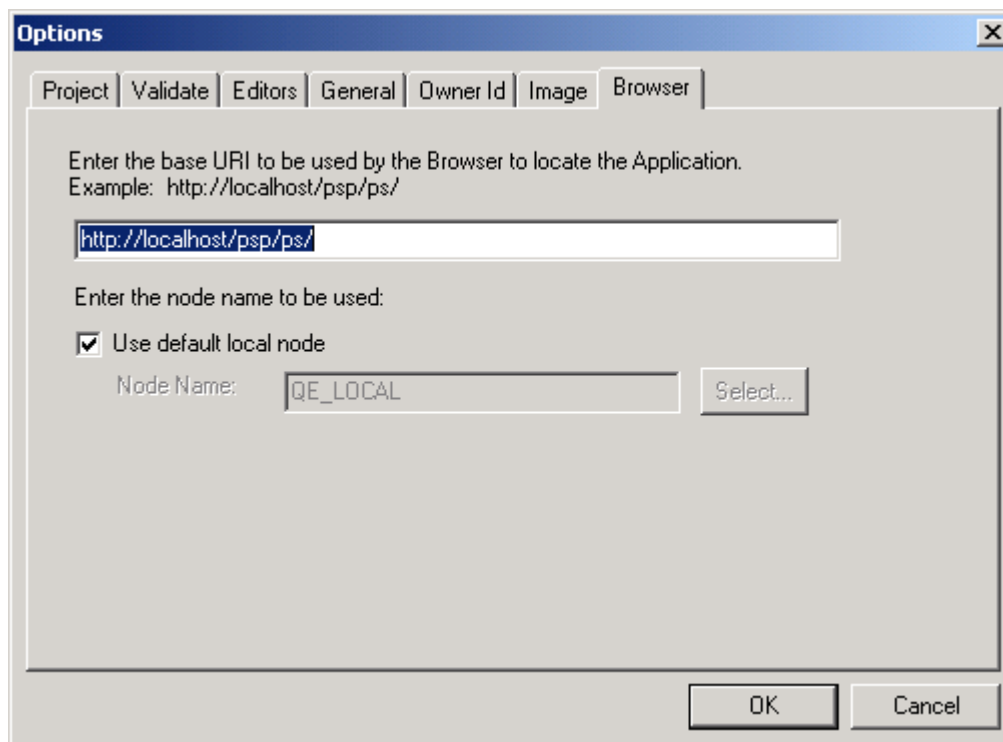
The default file

5. Enter a file name in the File name field or click the browse button to search for a specific file.

Otherwise, the system applies the default file name of *spellcheck.log* and saves the results to the temp directory. This field is required when the Log to file option is selected.

## Configuring Browser Options

The Browser tab on the Options dialog box enables you to specify values that are required to compose a URL for the browser to locate a portal page. Spell check uses the information recorded in this dialog box to launch portal pages from the Spell Check tab in the output window when you double click a portal registry misspelling.



Options dialog box, Browser tab

To configure browser options:

1. Select Tools, Options and click the Browser tab.
2. Enter the base URI or use the default setting.

This is the URI of the node that hosts the portal.

3. Set the node name that is to be used with the base URI.

The Use default local node check box is selected by default. Clearing this check box enables the Select button, which allows you to set the node name in the Select Node dialog box.

The Select Node dialog box displays the list of available nodes by project. You can perform a search for the appropriate node by clicking the Select button.

---

## Building and Maintaining Data

This section discusses:

- Data building and maintenance.
- Creating SQL tables.

## Data Building and Maintenance

The underlying database definitions must be kept in sync with PeopleSoft definitions. Therefore, from PeopleSoft Application Designer, you must:

- Submit SQL Alter and Create commands for all types of database definitions.
- Manage the database index.
- Manage the table space.
- Manage the DDL model for tables, indexes, and table spaces.

## Creating SQL Tables

To access data, you must create a SQL table and then store data in the fields and records. The process of running the SQL necessary to synchronize the database with records, indexes, and DDL is called *Build*.

The Build feature processes definitions at these three levels:

- Current record.
- Selected records in the project workspace.
- All records in the project.

### **See Also**

Chapter 7, "Administering Data," page 147

## Chapter 3

# Working With Projects

This chapter provides an overview of projects and discusses how to:

- View projects.
- Create and save projects.
- Set project properties.
- Insert definitions into a project.
- Remove definitions from a project.
- Merge projects.
- Use change projects.
- Set project options.
- Validate projects.
- Set system IDs.

---

## Understanding Projects

While not one of the eight steps of application development, using projects is an efficient way to organize your definitions as you develop and configure your application. A project simply keeps track of all definition types as a simple list of definition names; it is *not* where the definitions are stored. Development definitions exist outside of the project in your PeopleSoft database.

You are not required to use projects in PeopleSoft Application Designer; you can create and edit individual definitions without associating them with a project. However, using projects can help you:

- Organize related definitions.
- Understand relationships among definitions.
- Coordinate the work of several developers.
- Streamline upgrade tasks.
- Search for fields or records.
- Access related definitions easily.

## Viewing Projects

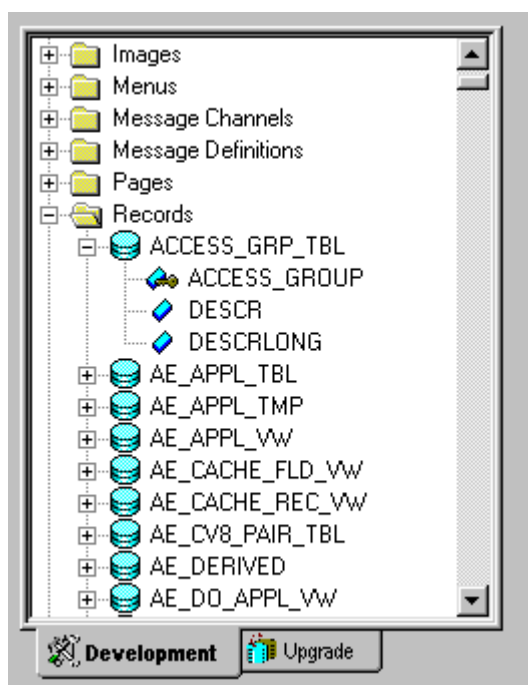
On the left-hand side of your screen, the project workspace displays one project at a time and all of its definitions, making them easily accessible for opening and editing. It includes two views representing the collection of definitions in the project: Development and Upgrade. Switch between the views by selecting the folder tabs at the bottom of the project workspace.

This section discusses how to:

- Access definitions for a project.
- Using the upgrade view.

## Accessing Definitions for a Project

In the Development view, the project folder contains a folder for each definition type included in the project. To open folders and view related definitions, click the expand (+) button. Access the definitions for a project by double-clicking the definition.

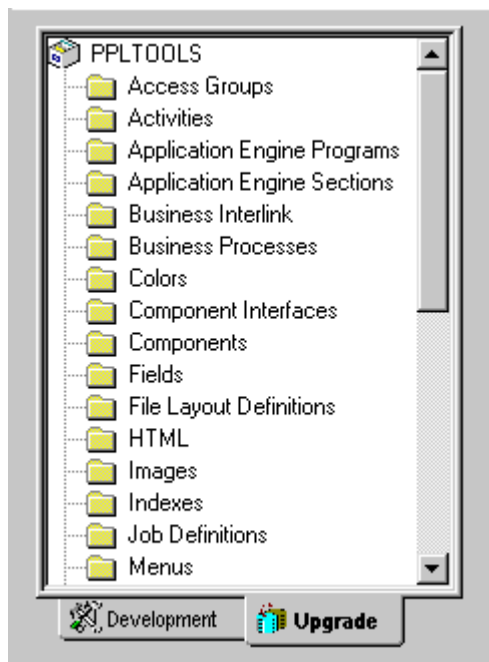


Development view in the project workspace

The definitions appear in a *project tree*, organized by definition type. The project tree enables you to see all definitions associated with the project and their connection, by showing the definitions in an hierarchical format. By expanding definitions that have nested levels, such as records, you'll see the field definitions included in that record.

## Using the Upgrade View

The Upgrade view helps to streamline the migration of definitions—such as records, pages, or PeopleCode—from one PeopleSoft database to another. It displays all of the definitions available for upgrade and attributes related to the upgrade process. When you double-click a definition type in the Upgrade view, an upgrade definition window appears in the definition workspace, displaying the definitions of that type that are available for upgrade and the associated upgrade options.



Upgrade view in the project workspace

This view is important when moving changes into production and when upgrading to new PeopleSoft releases. Definitions in development projects and upgrade projects are interchangeable, meaning that components that were developed in PeopleSoft Application Designer can be copied or compared using the Upgrade view.

### See Also

*Enterprise PeopleTools 8.50 PeopleBook: PeopleSoft Application Designer Lifecycle Management Guide, "Upgrading with PeopleSoft Application Designer"*

---

## Creating and Saving Projects

This section discusses how to:

- Create a new project.
- Save a project.

---

**Note.** Changes that you make to a definition in a project are also reflected outside the project. Before you change a definition in a project, be aware of how it might affect other definitions outside the project. For example, when you rename or delete a field, you do so globally. Although you can enter a project name as selection criteria in certain PeopleSoft Application Designer dialog boxes (such as Open, Rename, and Delete), this is just to narrow the list of definitions returned in the selection list.

---

## Creating a New Project

When you start PeopleSoft Application Designer, an empty project titled "Untitled" always appears, whether you use the project or not. Unlike other definitions, only one project can be open at a time. Although PeopleSoft encourages you to develop in projects, you can hide the project workspace window and ignore the project entirely.

To create a new project:

1. Select File, New.
2. Highlight Project as the new definition type to create, and click OK.

---

**Note.** Another way to create a new project is to open an existing project. Then, select File, Save Project As, and enter a new name for the project. All of the existing definitions in the project are also copied.

---

See [Chapter 2, "Using PeopleSoft Application Designer," Saving Definitions, page 31](#).

To open an existing project, follow the same procedure as you would for any other definition type.

See [Chapter 2, "Using PeopleSoft Application Designer," Opening Definitions, page 28](#).

## Saving a Project

Use a different command when saving an entire project than when saving other types of definitions.

To save a project, select File, Save Project. You can also select File, Save Project As to save the project with a new name. This is a good way to make a copy of a project, because the original project still exists under the original name.

The save toolbar button saves whichever definition is active in the definition workspace, but it cannot be used to save an entire project.

---

## Setting Project Properties

Use the Project Properties dialog box to enter information about the project. Access Project Properties from the File menu.

**Project Properties**

General | Update IDs | Pre-Requisites

PPLTLS84

Description:

Comments:

☒ Change Project

Owner ID:

Last Updated

Date/Time: 05/11/01 12:13:52AM

By User: QEADMIN

Application Upgrade Target

Server:

Database:

User:

OK Cancel

Project Properties dialog box

In the General tab, enter a description of the project and any additional comments you might want others to be aware of when using the project. Select the Change Project check box only if you are creating a change project to be used with the Change Packager. Selecting this option enables the Update IDs and the Pre-Requisites tabs, also used only with the Change Packager.

See *Enterprise PeopleTools 8.50 PeopleBook: Change Assistant*, "Working with Change Packages."

Select the appropriate Owner ID for the project. The Owner ID allows for a way to identify which definitions are owned by which PeopleSoft applications, such as PeopleSoft General Ledger, Accounts Receivables, and so on. The values in the drop-down list box are translate table values associated with the OBJECTOWNERID field.

## Inserting Definitions Into a Project

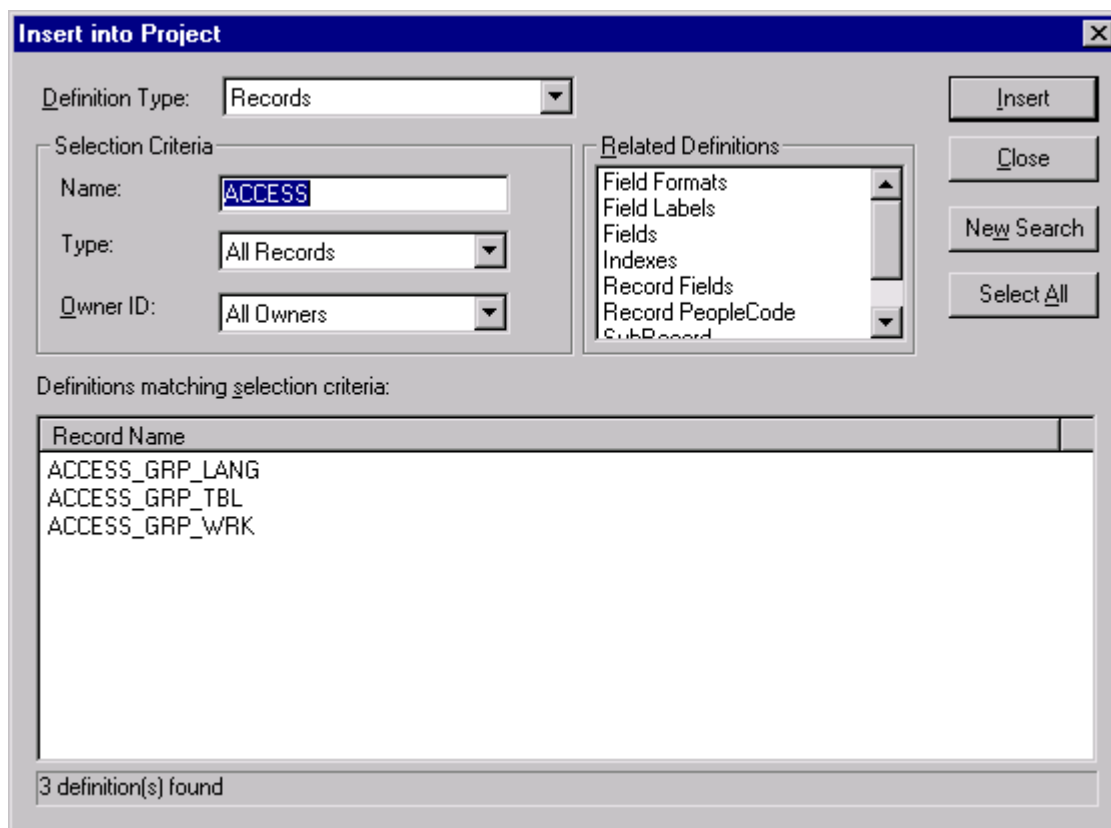
You can add definitions to a project by selecting Definitions into Project from the Insert menu.

To insert a definition into a project:

You can:

- Insert definitions individually.

- Select a number of definitions to insert as a group.



Insert into Project dialog box

### ***Inserting a Single Definition***

To insert a single definition into a project:

1. Open the definition that you want to insert into the project.

Make sure that the definition is the active definition in the definition workspace.

2. Select Insert, Current Definition into Project.

The definition is added to whichever project is currently open.

### ***Inserting a Group of Definitions***

To insert a group of definitions into a project:

1. Select Insert, Definitions into Project.

The Insert into Project dialog box appears.

2. Select the type of definition to insert from the Definition drop-down list box.

To see all available definitions of that type, including upgrade-only definition types, click the Insert button or press Enter.

3. (Optional) Enter the selection criteria.

<b>Name</b>	Enter the definition name or a partial field name and press ENTER.
<b>Type</b>	Narrow your search by selecting a specific definition type, such as View for a record definition.
<b>Owner ID</b>	Narrow your search further by selecting an application with which the definition is used.

4. Select the definitions to insert.

To select multiple definitions, hold down either Shift or Ctrl while clicking the appropriate definitions.

5. (Optional) Specify which related definitions to insert.

To insert related definitions, specify them in the list of related definitions. To select multiple related definitions, hold down either Shift or Ctrl while clicking the appropriate definitions.

---

**Note.** After you insert definitions into a project, the upgrade-only definitions appear in the upgrade view of the project workspace, not the development view.

---

The status bar at the bottom of the Insert into Project dialog box indicates that definitions have been inserted. Additionally, the Results tab on the output window displays the number of definitions inserted each time you perform an insert.

---

## Removing Definitions From a Project

*Removing* a definition from a project is different than *deleting* it from the database. When you remove a definition from a project, the reference to the definition is removed from the project, but the definition still exists in the database and thus may be a part of any other project.

To remove definitions from a project:

1. In the project workspace, select the definition that you want to remove from the project.

To select multiple items, highlight one or more nodes by using the Ctrl key while clicking the definitions to remove.

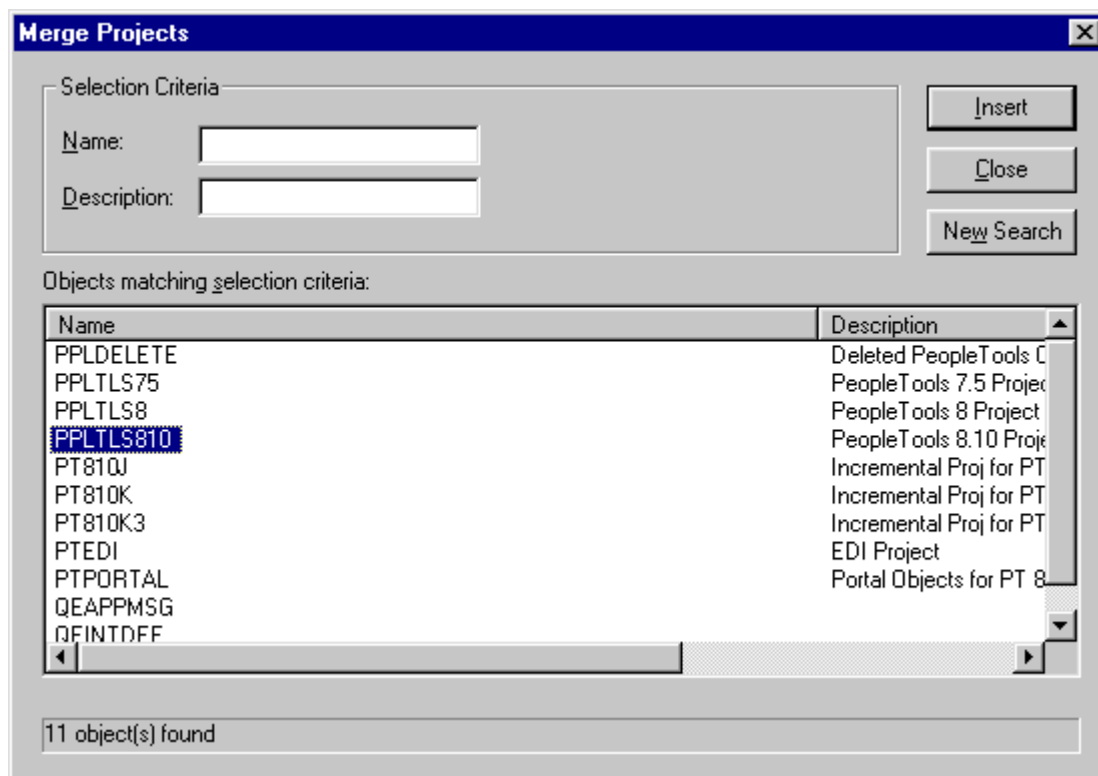
2. Press Delete, or right-click to select Remove from Project from the pop-up menu.

This does not delete the definition; it just removes it from the project.

---

## Merging Projects

You can merge two or more projects by inserting all of the definitions from one project into another.



Merge Projects dialog box

To merge projects:

1. Open the project in which you want to insert another project.
2. Select File, Merge Projects.

The Merge Projects dialog box appears.

3. Enter selection criteria.
  - a. Enter a project name or description (or the beginning characters of either).
  - b. Click Insert or press Enter to display projects matching the selection criteria that you entered.
4. Select a project to insert into your currently open project.

Double-click the project that you want to insert, or highlight the project and click Insert. To select multiple projects, use the Shift or Ctrl keys.

5. Click Insert.

All selected definitions are inserted into the project in one action. After each insert, note the information in the status bar and on the Results tab in the output window.

---

## Using Change Projects

Change projects are used with the Change Packager feature to enable you to group together definitions that are affected by PeopleSoft application updates. Combined with the PeopleSoft Change Assistant, the Change Packager feature significantly reduces the complexity and time required to apply application updates from PeopleSoft.

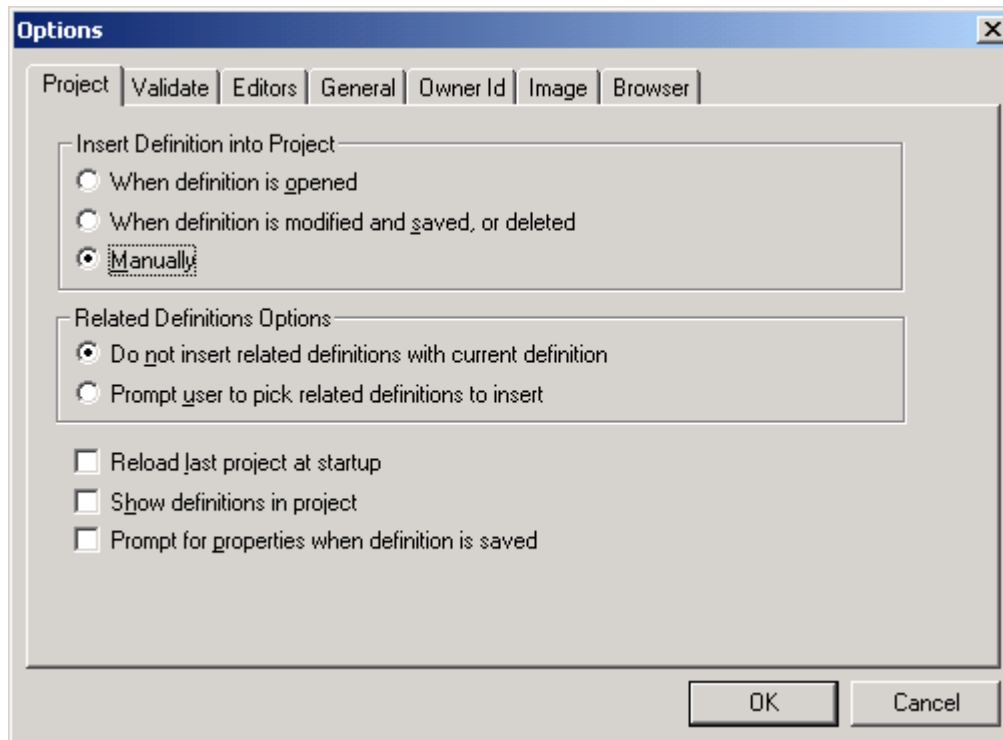
When you create a change project, you must set the appropriate update IDs and pre-requisites.

See *Enterprise PeopleTools 8.50 PeopleBook: Change Assistant*, "Working with Change Packages," Creating a Change Project.

---

## Setting Project Options

Set processing options for all project operations in the Project Options dialog box.



Setting project options

To set project options:

1. Select Tools, Options.

The Options dialog box appears.

2. Select the Project tab in the Options dialog box.

Define when and how definitions are added to the project and how the project appears in the project workspace.

<b>When definition is opened</b>	Automatically inserts any definition that you open into the current project.
<b>When definition is modified and saved, or deleted</b>	Automatically inserts any definition that you save or delete into the current project.
<b>Manually</b>	Does not automatically insert any definition into the current project. This is the default option. Use the Insert menu bar to insert a definition into the project.
<b>Do not insert related definitions with current definition</b>	Does not prompt you to include related definitions. Only the specified definition is inserted, not related definitions.
<b>Prompt user to pick related definitions to insert</b>	After you select Insert, Current Definition into Project, a dialog box appears, prompting you to pick which related definitions to insert.
<b>Reload last project at startup</b>	When starting PeopleSoft Application Designer, the project that was last opened in the previous session opens automatically.
<b>Show definitions in project</b>	Displays a black mark next to icons for definitions that are explicitly in a project (Development view) to indicate that they are in the project. Attribute definitions, which are not explicitly part of the project, are not marked.
<b>Prompt for properties when definition is saved</b>	Prompts you with a definition properties dialog box before you can save the definition.

---

## Validating Projects

An important part of the PeopleSoft upgrade process involves validating your upgrade project. PeopleSoft Application Designer includes a validate utility to make sure that all definitions included in the project actually exist in your database.

To validate a project:

1. Close all active component definitions in the definition workspace.  
Otherwise, the validation occurs in the component, rather than the project.
2. Select Tools, Options.
3. Select the Validate tab in the Options dialog box.
4. Select Validate project integrity.
5. Click OK.

## 6. Select Tools, Validate Project.

The validate utility runs a series of tests on the project or components and sends its results to the Validate tab in the output window. If errors are found, they are listed on this tab.

---

**Note.** The Validate WSRP compliance option applies to projects that contain PeopleSoft content intended for WSRP production.

---

See *Enterprise PeopleTools 8.50 PeopleBook: PeopleTools Portal Technologies*, "Using WSRP to Consume and Produce Remote Portlets," Validating WSRP-Compliance.

---

## Setting System IDs

The system ID functionality makes system ID maintenance easier for developers. This is especially true for those working on mobile applications where the System ID field must be set to SYNCID on the record properties for each record of each component used in mobile applications.

You can set the system ID for an overall project or for the current record definition in the project workspace.

To set system IDs for a project:

1. Open the project.
2. Select Build, Set System IDs for Project

The Build Progress dialog box appears, displaying the progress of the command for the number of objects affected by this command.

### See Also

*Enterprise PeopleTools 8.50 PeopleBook: Global Technology*, "Selecting and Configuring Character Sets and Language Input and Output," Validating Field Length



## Chapter 4

# Creating Field Definitions

This chapter provides an overview of field definitions and discusses how to:

- Create new field definitions.
- Use multiple labels.
- Open an existing field definition.
- View field properties.
- Change field definitions.
- Set custom field formats.
- Use the translate table.

---

## Understanding Field Definitions

Fields are the basic building blocks in your PeopleSoft system; in database terms, they represent columns in a table or view. Field definitions are stand-alone development definitions, defined in PeopleSoft Application Designer, and can be shared across multiple record definitions. Changes to field properties affect all records that include that field.

Each field definition includes attributes, such as data type, field name, long name, short name, field length, and various formatting values.

### ***Fields in Record Definitions***

By grouping fields that share a common theme, you build a record definition. For example, a department table (record definition) might include a department ID field, department name field, location field, and so on.

For each field definition that you add to a record definition, you define additional characteristics, called record field properties, that are specific to the way the field is used in that particular record. Record field properties are *not* shared with other records in which the field is used.

---

## Creating New Field Definitions

This section assumes that you are already familiar with the general terminology and functions of PeopleSoft Application Designer.

This section discusses how to:

- Create a new field definition.
- Specify attributes for a new field definition.
- Specify character field attributes.
- Specify long character field attributes.
- Specify number field attributes.
- Specify date field attributes.
- Specify time field attributes.
- Specify DateTime field attributes.
- Specify image field attributes.
- Specify ImageReference field attributes.

## Creating a New Field Definition

To create a new field definition:

1. Click the New button on the PeopleSoft Application Designer toolbar.

The New dialog box appears.

2. Select Field.
3. Click OK.

A new character field opens in the definition workspace area.

4. Select one of the following field types from the Field Type drop-down list box.

---

**Note.** Make sure to name your fields according to the accepted naming conventions.

---

See [Chapter 2, "Using PeopleSoft Application Designer," Saving Definitions, page 31.](#)

<b><i>Character (Char)</i></b>	Alphanumeric field of fixed length.
<b><i>Long Character (Long)</i></b>	Alphanumeric field of variable length used for textual entries, such as comments or descriptions.
<b><i>Number (Nbr)</i></b>	Positive numeric field of fixed length for which decimals are allowed.
<b><i>Signed Number (Sign)</i></b>	Positive or negative numeric field of fixed length for which decimals are allowed.
<b><i>Date (Date)</i></b>	Date field of constant length. The system edits dates and prohibits incorrect ones, such as day 42 or month 20. Date fields always store a four-digit year.
<b><i>Time (Time)</i></b>	Time field of a constant length. Built-in edits prohibit impossible times, such as hour 26, minute 70, or second 94.
<b><i>DateTime (DtTm)</i></b>	Date and time field of constant length. Built-in edits prohibit impossible dates and times, as defined in the individual Date and Time fields. DateTime fields always store a four-digit year.
<b><i>Image (Img)</i></b>	Image field to store images in a user-defined format, such as JPEG or GIF.
<b><i>ImageReference (Iref)</i></b>	Use this to change an image dynamically at runtime using PeopleCode.
<b><i>Attachment (Att)</i></b>	Maps to a BLOB database type to hold the contents of a file attachment. However, PeopleSoft recommends using the existing field, FILE_DATA, if a BLOB column is needed. For attachments, use the subrecord ATTACH_DETAIL, which contains FILE_DATA.

After you select a field type, a field definition window appears in the definition workspace. Use the field definition window to specify attributes to complete your new field definition.

## Specifying Attributes for a New Field Definition

Access the field definition window for the new field definition.

The following property settings are shared by all field types.

---

**Note.** Multiple views of this dialog box are available, depending on which field type you select. PeopleSoft documents fields that are common to all views first.

---

**Label ID**

Enter a unique identifier for the field that is less than 18 characters, without spaces, and in uppercase. Make this the default label by selecting the Def check box in the last column; otherwise, the first label is the default. You can set only one field label as the default.

---

**Note.** You can specify multiple labels for one field and change them at runtime using PeopleCode.

---

See [Chapter 4, "Creating Field Definitions," Using Multiple Labels, page 73](#).

**Long Name**

Enter up to 30 characters. The name that you enter here appears as the RFT Long option for a page field. This label is also used for the tool tip, or hover text, for links at runtime if it is different from the assigned display value of the link.

---

**Note.** Do not use the \ (backslash) character in the long name of cubes and members. Unexpected behavior might occur.

---

**Short Name**

Enter up to 15 characters as an alternate name to use on pages and reports if you have insufficient space to display the long name. If you leave this field blank, the system automatically copies the first 15 characters of the long name into this field. The name that you enter here appears as the RFT Short option for a page field.

---

**Note.** If you name the field the same as the default label ID, you can identify it more easily when using PeopleCode.

---



---

**Note.** Do not use the \ (backslash) character in the short name of cubes and members. Unexpected behavior might occur.

---

**Def (default)**

When you enter the first label ID, the system automatically selects the Def check box. The system uses this default field if you have multiple labels. You can have only one default field.

**Not Used**

This check box is always read-only in the field definition properties dialog box. You can change its value using PeopleCode, causing the field to be:

- Skipped as part of the unique index.
- Exposed to PeopleCode for read and write operations.
- Ignored in Query.
- Ignored in nVision.
- Unavailable to users in search pages or list boxes if it is a search key or list box item.

See *Enterprise PeopleTools 8.50 PeopleBook: PeopleCode Language Reference*, "PeopleCode Built-in Functions," SetDBFieldNotUsed.

<b>Chart Field</b>	Read-only in the field definition properties dialog box. It is used for character fields and by financial applications to specify a usage pattern. You can change the value using PeopleCode.
<b>Encrypt</b>	Select this option to encrypt the field using Oracle Transparent Data Encryption (TDE). <hr/> <b>Note.</b> The Encrypt option is disabled for field types that do not support encryption, for database platforms other than Oracle, as well as for Oracle database versions where TDE is not supported. <hr/> <i>See Enterprise PeopleTools 8.50 PeopleBook: Data Management, "Administering PeopleSoft Databases on Oracle," Encrypting Fields.</i>

## Specifying Character Field Attributes

Character fields are used for names, codes, and anything with letter values. You also use them for numbers for which you want to contain formatting attributes and the data itself. Examples of such fields are those in which you enter telephone numbers and zip or postal codes. You cannot use character fields in calculations. You can enter decimal numbers in character fields, but you cannot use them mathematically until you convert them back to numeric fields.

You can specify character field attributes at design time, and you can access most of them at runtime with PeopleCode.

<b>Field Length</b>	Enter a whole number between 1 and 254 for the length of the field.
---------------------	---

**Format Type**

Select one of the following values:

*Uppercase*: Converts the field value to uppercase and signifies that no other formatting options apply to this field. Use this option for code values, such as department ID, for which it doesn't matter whether the user enters the value in uppercase or lowercase. This is the default format.

*Mixedcase*: Stores uppercase and lowercase characters as entered. Use this option for fields that contain textual data, such as a department or company division name. For example, *Accounting – Receivables* is more readable than *ACCOUNTING – RECEIVABLES*.

*Numbers Only*: Forces entries to be numeric. This is useful for enforcing numeric values without redefining the field as a number field. This option automatically populates fields with leading zeros. For example, if the user enters *1* in a three-digit field, the system changes this to *001*.

*SSN*: Formats the entry in U.S. Social Security Number format (999-99-9999). Define 9 as the field length, even though the display length is 11. The system automatically adds dashes when formatting the field for display, but the number is stored in the database without the dashes.

*SIN*: Formats the entry in Canadian Social Insurance Number format (999-999-999) and performs the standard check-digit verification for SIN. Define 9 as the field length, even though the display length is 11. The system automatically adds dashes when formatting the field for display, but the number is stored in the database without the dashes.

*Raw Binary*: Allows character fields containing embedded NULLs, such as encrypted values.

*Name*: Indicates that the field entry uses a PeopleSoft standard name convention:

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

The entry can contain alphanumeric characters, spaces, periods, hyphens, and apostrophes. Uppercase and lowercase characters are preserved as entered—in other words, mixed case formatting is included automatically. Valid entries might include:

- O'Brien,Michael.
- Jones IV,James.
- Phillips MD,Deanna Lynn.
- Reynolds Jr.,Dr. John Q.
- Phipps-Scott,Ms. Adrienne.
- Knauft,Günter.

---

**Note.** For names that include Chinese, Japanese, or Korean (CJK) ideographic characters, different standard name formats can be used.

---

See *Enterprise PeopleTools 8.50 PeopleBook: Global Technology*, "Selecting and Configuring Character Sets and Language Input and Output," Using CJK Ideographic Characters in Name Character Fields.

*Postal Code North America:* Formats the entry to U.S. zip code format or Canadian postal code format. Canadian postal codes are edited to ensure that alpha and numeric characters are entered in the correct positions.

The following entry formats apply. Stored formats appear in parentheses following the entry format:

- 99999 (99999)
- 99999-9999 (99999-9999)
- 999999999 (99999-9999)
- A9A9A9 (A9A 9A9)
- A9A 9A9 (A9A 9A9)

*Postal Code International:* Allows the entry of international postal codes. The field length must be at least 7 digits. If 9 digits are entered, the system assumes that the value is a U.S. zip code; if the entry is in A9A9A9 format, the system assumes that the value is a Canadian postal code. Then, the value is formatted accordingly. For all other entries, no formatting is applied.

*Phone Num North America (phone number North America):* Formats the entry in North American telephone number format.

The following entry formats apply. Stored formats appear in parentheses following the entry format:

- 9999999 (999-9999)
- 999-9999 (999-9999)
- 9999999999 (999/999-9999)
- 999/999-9999 (999/999-9999)

*Phone Number International:* Allows the entry of international telephone numbers. If you enter 7 or 10 digits (and no other characters), the system assumes that the value is a North American phone number, and formats it accordingly. For all other entries, no formatting is applied.

*Custom:* If you select this value, the Family Name and Display Name fields become available for entry. These selections enable you to apply additional formatting attributes that affect how the field appears on a page.

## Specifying Long Character Field Attributes

Depending on your database environment, the maximum length of a long character field ranges from several thousand characters to 32,767 characters (64,000 bytes). Using a long character field instead of a regular character field enables the user to insert a tab at runtime by pressing ctrl+tab.

**Maximum Length**

To control the length of a long character field, enter the maximum number of bytes that you want the system to write to this field. If you don't specify a maximum length, or specify 0 (zero), you can enter an unlimited number of characters, depending on the capacity of your database system.

When working with fields, be aware that your chosen database system limits the amount of data contained in a field. For Oracle the field size is truncated to 1333 characters. Consider this limitation when choosing to use fields of a Long Character data type.

For Oracle, if the maximum length of a long field is less than 4001 bytes, the field can be stored more efficiently as a VARCHAR2(4000). To support PeopleSoft's Unicode implementation on the Oracle platform, PeopleSoft defined "long" columns with a specified length of 1330 or less as VARCHAR2 columns. If the length specified was greater than 1330, the column was generated as a long VARCHAR or long raw, depending on the setting of the Raw Binary check box. This allows for the maximum possible usage of a tripled VARCHAR2 character length for Unicode-defined databases.

**Raw Binary**

Select to store data that is not converted when moving data between different systems. Raw binary is intended for binary data and byte strings. For example, raw binary can store graphics, sound, documents, and arrays of binary data.

**Specifying Number Field Attributes**

When you create a new field, notice that there are two field types for numbers:

- Number fields.
- Signed number fields.

Both field types are fixed in field length and allow the entry of positive numbers. Only signed numbers, however, allow the entry of negative numbers. Otherwise, both number field types are the same.

Unlike character fields, a number field can contain decimals but does not contain special formatting, such as the formatting for telephone numbers. Use number fields for calculations. If you use number fields for codes, you sacrifice flexibility to change your coding structure to alphanumeric in the future.

Field Type: **Number** ☐ Signed

Integer Positions: **3** Decimal Positions: **0**

Field Labels

	Label ID	Long Name	Short Name	Def
1				<input type="checkbox"/>

Number Field type

**Integer Positions**

Specify the number of integer positions to the left of the decimal. If the number contains only decimal positions, leave this field blank.

**Decimal Positions**

Specify the number of decimal positions to the right of the decimal. If a number contains only integer positions, leave this field blank. In calculations, the system rounds up the result to the number of decimal positions defined here.

---

**Note.** Number fields cannot exceed 31 positions. Therefore, the sum of integer and decimal positions cannot exceed 31.

The field length notation differs between PeopleSoft Application Designer and Structured Query Language (SQL). If you specify a field length of 8 integer positions and 3 decimal positions in PeopleSoft Application Designer, SQL processes that as a length notation of 11.3.

---

SQL describes field length in terms of *precision* and *scale*. Precision is the total number of integer and decimal positions. Scale is the number of decimal positions. Thus, 11.3 means 8 integer positions and 3 decimal positions; 11.0 means 11 integer positions and 0 decimal positions.

When you enter 0 in the Decimal Positions field and enter 5 or 10 in the Integer Positions field, the RawBinary format option appears. The RawBinary format converts a number field value to a full-length 16- or 32-bit integer. See the example in the table below.

**Signed**

Select this check box for a field that is used for arithmetic calculations that might contain a negative value.

This table shows how the integer position affects the RawBinary storage format:

<i>Integer Position</i>	<i>Maximum Value</i>	<i>Number Stored As</i>
5	65535	16-bit integer
10	4294967295	32-bit integer

## Specifying Date Field Attributes

Date fields contain calendar dates. A date field has a field length of 10 and is maintained by the system. The default format of a date field is defined by the database and can be overridden by your browser settings.

**Default to 2000 if year less than or equal to**

Specify the last two digits of a year, which is typically what users enter in a date field. If you enter 50, the default century becomes 2000 when someone enters 0 through 50 as the year in a date field. If someone enters a number between 51 and 99 in a date field, the default century becomes 1900. The default value is 50.

---

**Note.** When working with date and time fields on pages and search pages, carefully consider search criteria. Assume the date, time, and time zone fields are separate with the condition that the time is time zone-controlled by the time zone field and the time is associated with the date field. In this case, you need to provide date, time and time zone on the search page if any one of those search criteria is required.

---

## Specifying Time Field Attributes

Time fields are fixed-length fields (15 positions, format of HH:MI:SS.999999) that contain the time of day. The maximum time precision varies, depending on your database.

**Time Formatting** Specify the formatting for this field. HH represents hours, MI represents minutes, SS represents seconds, and 999999 represents microseconds.

---

**Note.** You can use %CLIENTDATE as the default constant for the Time field. %CLIENTDATE adjusts the date as appropriate to the time zone of the browser.

---

### See Also

*Enterprise PeopleTools 8.50 PeopleBook: PeopleCode Language Reference*, "System Variables," %ClientDate

*Enterprise PeopleTools 8.50 PeopleBook: PeopleCode Language Reference*, "System Variables," %Date

## Specifying DateTime Field Attributes

The DateTime field is a fixed-length field (26 positions, format of YYYY-MM-DD-HH-MI-SS.999999) that holds dates and times. The maximum date and time precision depends on your database.

---

**Note.** You can use %CLIENTDATE as the default constant for the DateTime field. %CLIENTDATE adjusts the date as appropriate to the time zone of the browser.

---

### See Also

*Enterprise PeopleTools 8.50 PeopleBook: PeopleCode Language Reference*, "System Variables," %ClientDate

*Enterprise PeopleTools 8.50 PeopleBook: PeopleCode Language Reference*, "System Variables," %Date

## Specifying Image Field Attributes

Use Image fields to store application data that takes the form of images. Image fields are functionally no different from any other type of field. Just as end-users employ character fields to store names and addresses, they can employ image fields to keep track of all types of digital pictures. This field enables you to store pictures of assets, company logos (in SQL tables), or scanned images of employees (as part of their personal data rows).

**Maximum Length**

Specify the maximum number of bytes available to store an image. If you don't specify a maximum length, the maximum length is determined by your database platform. If a user attempts to cut and paste an image that is larger than the maximum length or database capacity, an error message appears.

The following databases apply (maximum length appears in parentheses):

- DB2/zOS (up to 32 kilobytes\*).
- Oracle (2 gigabytes).
- Sybase (2 gigabytes).
- SQL Server (2 gigabytes).
- Informix (2 gigabytes).
- DB2 UDB for Linux, Unix, and Windows (32 kilobytes).

\* Depends on the page size, the buffer pool, and the other data that is being stored in that row.

**Image Format**

Select an image format. Format determines the type of image that you can select and write to the database for this field. PeopleSoft supports BMP, DIB, GIF, CUT, EPS, JPG, PCX, PCT, RLE, and TGA image types. At runtime, the user can upload BMP, DIB, GIF, and JPG images to the database.

## Specifying ImageReference Field Attributes

Use the ImageReference field to store static images that can be changed at runtime with PeopleCode. By associating image definitions with an ImageReference field on a page, you can display images dynamically. For example, to have different images appear on an employee profile depending on the status of the employee's current review, use the ImageReference field to reference the current review field and display the appropriate image.

## Working With Additional Oracle Data Type Mappings

If your site has upgraded to a PeopleSoft 9.x application and performed the appropriate conversion process during the application upgrade, there are different data type mappings between PeopleSoft field types and the underlying Oracle system.

See The upgrade documentation and release notes for your release for information regarding the conversion process.

The following table describes how the PeopleSoft field types involved map to the underlying Oracle data types.

<b>PeopleSoft Field Type</b>	<b>Pre Conversion Oracle Data Type Mapping</b>	<b>Post Conversion Oracle Data Type Mapping</b>
Character	ANSI: VARCHAR2(n) Unicode: VARCHAR2(n*3)	ANSI: VARCHAR2(n) Unicode: VARCHAR2(n)  <b>Note.</b> For Unicode the VARCHAR2 column is interpreted as the number of Unicode characters due to init.ora parameter NLS_LENGTH_SEMANTICS is set to "CHAR".
Long Character	If $1 \leq n \leq 2000$ ANSI: VARCHAR2 Unicode: VARCHAR2(n*3) CHECK (Length <field name>) <=n) If $n = 0$ or $n > 2000$ ANSI: LONG VARCHAR Unicode: LONG VARCHAR	ANSI: VARCHAR2(n) when $1 \leq n \leq 4000$ CLOB when $n = 0$ , or $n > 1333$ Unicode: VARCHAR2(n) when $1 \leq n \leq 1333$ CLOB when $n = 0$ , or $n > 1333$  <b>Note.</b> For Unicode the VARCHAR2 column is interpreted as the number of Unicode characters due to init.ora parameter NLS_LENGTH_SEMANTICS is set to "CHAR".
Long Character with Raw Binary attribute	ANSI: LONG RAW Unicode: LONG RAW	ANSI: BLOB Unicode: BLOB
Image	ANSI: LONG RAW Unicode: LONG RAW	ANSI: BLOB Unicode: BLOB
Attachment	ANSI: LONG RAW Unicode: LONG RAW	ANSI: BLOB Unicode: BLOB
Date	DATE	DATE
Datetime	DATE	TIMESTAMP
Time	DATE	TIMESTAMP

## Working With Additional Microsoft SQL Server 2005 Data Type Mappings

If your site has upgraded to a PeopleSoft 9.x application and performed the appropriate conversion process during the application upgrade, there are different data type mappings between PeopleSoft field types and the underlying Microsoft SQL Server 2005 system.

See The upgrade documentation and release notes for your release for information regarding the conversion process.

The following table describes how the PeopleSoft field types involved map to the underlying Microsoft SQL Server data types.

<b><i>PeopleSoft Field Type</i></b>	<b><i>Pre Conversion Microsoft SQL Server Data Type Mapping</i></b>	<b><i>Post Conversion Microsoft SQL Server Data Type Mapping</i></b>
Character	<i>ANSI: CHAR</i> <i>Unicode: NCHAR</i>	<i>ANSI: VARCHAR(n)</i> <i>Unicode: NVARCHAR(n)</i>
Long Character	<i>ANSI: TEXT</i> <i>Unicode: NTEXT</i>	<i>ANSI: VARCHAR(MAX)</i> <i>Unicode: NVARCHAR(MAX)</i>
Long Character with Raw Binary Image Attachment	IMAGE (ANSI and Unicode)	VARBINARY(MAX) (ANSI and Unicode)

---

## Using Multiple Labels

This section provides an overview of multiple labels and discusses how to:

- Add a new label.
- Delete a label.
- Select record field labels.

## Understanding Multiple Labels

You can define additional label pairs and an identifier for any field. Multiple fields are useful when you want different labels to appear on different pages. For example, on one page you can set the label to Begin Date such as to mark the beginning of a vacation, while on another page the label might be Class Start Date to mark the beginning of a training course.

Multiple labels are displayed in alphabetical order and sorted by label ID. You can sort the labels in a field by double-clicking the column header. The sorting is not saved, but it is helpful for quickly viewing the labels.

## Adding a New Label

Open the field definition in the definition workspace.

To add a new label, enter the label ID, long name, and short name in the last row. Each label can have related language labels. After the multiple labels are defined, you can select appropriate labels when designing a record or page. You can also specify different labels in PeopleCode. To add a line feed for Long Name and Short Name, enter *n* (backslash and the letter *n*). For example, *work \n month* adds a line feed between the words *work* and *month*.

When you create multiple labels, the first label ID remains the default label. You can change the default label by selecting the Def check box for a different row. There can be only one default label.

---

**Note.** Each label ID must be unique so that each field definition label can be identified separately, and it must be entered using uppercase letters with no spaces.

---

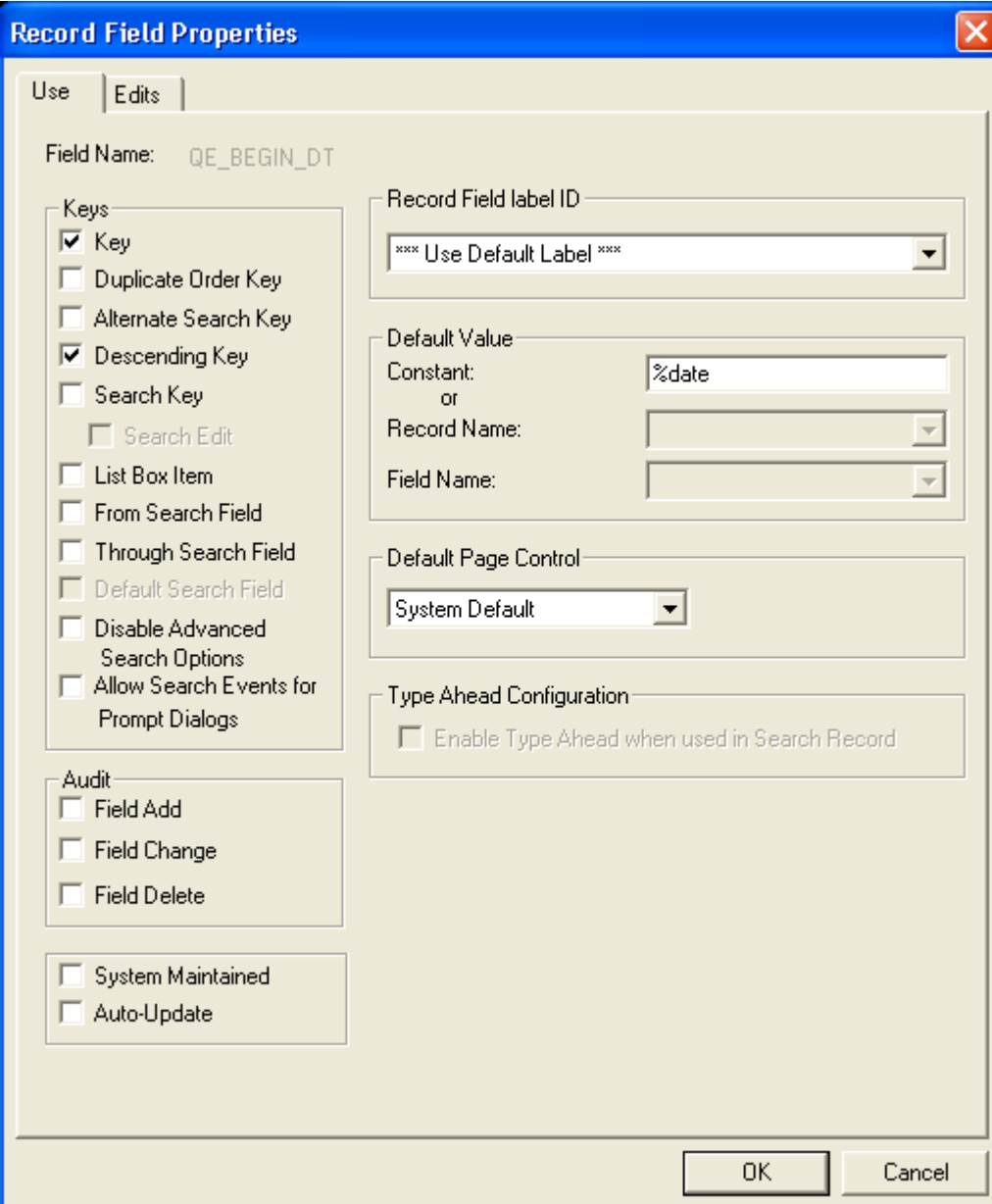
## Deleting a Label

Open the field definition in the definition workspace.

To delete a label, right-click a field label row and select *Delete Label* from the pop-up menu. Labels that are currently in use cannot be deleted. To determine where the label is used, select the label and select *Find Definition Reference* from the pop-up menu. The results appear in the Find Definition Reference output window.

## Selecting Record Field Labels

Access the Record Field Properties dialog box.



The image shows the 'Record Field Properties' dialog box with the 'Use' tab selected. The 'Field Name' is 'QE\_BEGIN\_DT'. The 'Keys' section has 'Key' and 'Descending Key' checked. The 'Record Field label ID' is set to '\*\*\* Use Default Label \*\*\*'. The 'Default Value' section has 'Constant' set to '%date'. The 'Default Page Control' is 'System Default'. The 'Type Ahead Configuration' section has 'Enable Type Ahead when used in Search Record' unchecked. The 'Audit' section has 'Field Add', 'Field Change', and 'Field Delete' unchecked. The 'System Maintained' and 'Auto-Update' options are also unchecked. The 'OK' and 'Cancel' buttons are at the bottom right.

Record Field Properties dialog box

For the record definition, the Record Field label ID drop-down list box is available on the Use tab of the Record Field Properties dialog box. The drop-down list box displays the label ID, short name, and long name that you set in the field definition. *\*\*\* Use Default Label \*\*\** is the default selection. The record field label changes automatically whenever you change the default label in the field definition.

## Opening an Existing Field Definition

There are several ways to open a field definition:

- Select File, Open from the menu.

From this dialog box, you can view all available fields in the database. You can open all fields of a certain type or from a specific project, or you can open a specific field or set of fields.

- Clone an existing definition.

To create a field definition with attributes similar to an existing one:

1. Open an existing field definition.
2. Select File, Save As.
3. Enter a new name and click the OK button.

- Open the field from the project workspace.

Double-click a field from the project workspace to open a field definition window in the definition workspace.

- Open the field from the record.

Open an existing record definition. When the record definition window opens, you can:

1. Select a field.
2. Right-click the field.
3. Select *View Definition* from the pop-up menu.

---

## Viewing Field Properties

There are several ways to view field properties:

- While in the field definition workspace, click the Properties button.
- Select File, Definition Properties.
- While in a record definition:
  1. Select the field.
  2. Right-click the field name.
  3. Select *View Field Properties* from the pop-up menu.

The General tab in the Field Properties dialog box provides a place for you to enter comments documenting the purpose of the field definition. The Owner ID drop-down list box enables you to select a valid owner, as in application project or role. The Last Updated group box contains information—date, time, and user ID—about the last time that the field definition was modified.

The Translate Values tab is available only for character fields.

## See Also

[Chapter 4, "Creating Field Definitions," Using the Translate Table, page 89](#)

---

## Changing Field Definitions

This section provides an overview of the effects of changing field definitions and discusses how to:

- Change field types.
- Rename field definitions.
- Delete field definitions.
- Print field definitions.

## Understanding the Effects of Changing Field Definitions

When you develop new record definitions or modify existing ones, you might need to change characteristics or attributes for a field. If you change attributes in a field definition, the change affects every occurrence of the field in every record definition. So, before you change basic attributes in a field definition, consider how it affects all of the record definitions in which the field occurs. If the change isn't appropriate for every occurrence of this field, consider defining a new field instead.

If you still find that a change is needed, PeopleSoft recommends that you determine the extent of the change by finding out how many records are affected. Do this by using the Find Definition References tool. Open the field definition and select Edit, Find Definition References, or right-click and select the option from the pop-up menu. If you're in a record definition window, another method is to select the field, right-click the field name, and select *Find Definition References--Field* from the pop-up menu.

The Find Definition References tab in the Output window displays a list of all projects, record definitions, page definitions, messages, channels, file layouts, and PeopleCode programs in which the field is used. Double-click a line of output to open the corresponding definition in the definition workspace.

---

**Note.** Changing the name or length in a field definition requires modification of the underlying SQL table. Do this either by running the build process and specifying Create Tables or Alter Tables, or by carrying out an action issued by your system administrator. For example, if you change the length in a field definition and 30 record definitions contain the same field name, you have 30 records that must be created or altered.

---

## Changing Field Types

Change a field type by opening a field and selecting another field type from the drop-down list box. However, this feature is intended as a design-time change—that is, before your tables have been created in the database. While some field type conversion can be done, results vary, depending on your database platform.

---

**Note.** If you change a field from one type to another, all records containing that field must be rebuilt by running the Alter process. Use Find Definition References to find all incidences of the changed field.

---

See [Chapter 7, "Administering Data," Altering Tables, page 173.](#)

## Renaming Field Definitions

You can use either the Rename Definition dialog box or the Open Definition dialog box to rename a field definition.

### ***Using the Rename Definition Dialog Box***

To rename a field definition from the Rename Definition dialog box:

1. Select File, Rename to open the Rename Definition dialog box.
2. Select *Field* as the definition type.
3. Enter the name of the field.

If you aren't sure of the name or the correct spelling, use the search criteria fields to locate the field to rename. The list displays field names that match your search criteria.

4. To rename a field in the list, select the field and click Rename, or double-click the field.

The field name becomes available for entry, and you can type the new name.

5. When you finish typing the new name, press enter or click the Rename button.

The Rename Field dialog box appears. Select the records in which you want the field to be renamed. The De-select All button toggles to Select All.

6. Click OK.

### ***Using the Open Definition Dialog Box***

To rename a field definition from the Open Definition dialog box:

1. Select File, Open to access the Open Definition dialog box.

This dialog box enables you to set up search criteria for locating field names.

2. Select Field as the definition type.

If you have additional information, use the Name field. You can further narrow your search criteria by selecting a field type or the name of the project that contains the field to delete.

3. Click the Open button or press enter when you have completed your search criteria.
4. You can rename a field in the Open Definition dialog box by selecting the field, right-clicking the field name, and selecting *Rename*.

You can also select the field name and click it a second time. The field name becomes available for entry and you can enter a new name.

5. When you finish typing the new name, press enter or click the Rename button.

### ***Results of Renaming Field Definitions***

When you rename a field, the system automatically changes all occurrences of it, including data on tables with columns named FIELDNAME where the data matches the field being renamed.

The only references that are *not* renamed are the text portions of SQL functions, such as SQLExec and Scroll Select. If you have already used SQL Create to create the underlying tables for the record definitions that contain the field that you renamed, you must re-create or alter those tables. If the tables contain data that you want to preserve, use the SQL Alter function to rename the database table fields.

## **Deleting Field Definitions**

You can use either the Delete Definition dialog box or the Open Definition dialog box to delete a field definition.

### ***Using the Delete Definition Dialog Box***

To delete field definitions by using the Delete Definition dialog box:

1. Select File, Delete to open the Delete Definition dialog box.  
Set up search criteria for locating field names.
2. Select *Field* as the definition type.
3. If you have additional information, use the Name field.
4. Select a field type or the name of the project that contains the field that you want to delete.

This further narrows your search criteria.

5. When your search results appear in the output window, delete the field definition by selecting the field and clicking the Delete button or double-clicking the field name.

---

**Note.** You cannot delete a field that is currently used in any records. Before you delete a field definition, you must first remove it from records in which it appears.

---

### ***Using the Open Definition Dialog Box***

To delete field definitions by using the Open Definition dialog box:

1. Select File, Open to access the Open Definition dialog box.  
This dialog box enables you to set up search criteria for locating field names.
2. Select *Field* as the definition type.
3. If you have additional information, enter it in the Name field.
4. You can further narrow your search criteria by selecting a field type or the name of the project that contains the field that you want to delete.
5. When you have completed your search criteria, click Open or press enter.

6. Right-click the field that you want to delete from the list of field names displayed.
7. Select *Delete* from the pop-up menu.

## Printing Field Definitions

If you plan to make changes to your field definitions, PeopleSoft recommends viewing your field definition before you proceed. You can view it on your screen or print it to a one-page report that combines information from several resources. This can also serve as a paper audit trail to document your database.

To print a field definition, select File, Print from the field definition window (or click the Print button in the toolbar) to open the Print dialog box, where you specify printing options. You can also print field definitions from the Open Definition dialog box. On the list resulting from your search, right-click the name of the field definition that you want to print, and select *Print* from the pop-up menu to view your printing options.

---

## Setting Custom Field Formats

This section provides overviews of custom field formats and format notation and discusses how to:

- Change format families.
- Edit formats.
- Use family options.
- Test formats.

## Understanding Custom Field Formats

PeopleTools has formatting provisions at both the field and page levels. The field format specifications affect internal values that are stored in the database, and the page formats affect the visual presentation of values.

There are several formatting options for U.S. Social Security Numbers, Canadian Social Insurance Numbers, postal codes, telephone numbers, and various currencies, dates, and times. Custom field formats extend these formatting provisions to support the same types of data in other countries. They also support other types of formatted information, such as bank codes, credit card numbers, part numbers, or serial numbers.

## Understanding Format Notation

Stored and display formats are defined by strings that contain *lexical*, *literal*, and *meta* characters:

- Lexical characters delimit character expressions and designate interpretation rules.
- Literal characters represent only themselves.
- Meta characters represent a class of characters.

Use the format symbols in the following tables to create the format notation for your own custom field formats:

<b>Meta Symbol</b>	<b>Description</b>
#	<p>Digit placeholder.</p> <p>If the number has more digits to the right of the decimal point than there are # symbols to the right in the format, the system <i>truncates</i> the number to as many decimal places as there are # symbols to the right.</p> <p>If the number has more digits to the left of the decimal point than there are # symbols to the left in the format, the system displays the extra digits.</p> <p>If the number has fewer digits to the right of the decimal point than there are # symbols to the right of the decimal point in the format, the system adds spaces.</p> <p>If the number has fewer digits to the left of the decimal point than # symbols to the left of the decimal point in the format, the system also adds spaces.</p> <p>Example format: ###.##</p> <p>Input: 1234.567 - Matches? Yes - Output: 1234.56</p>
0 (zero)	<p>Digit placeholder.</p> <p>Follows the same rules as for #, except that if the number has fewer digits than there are 0s in the format, the system displays the extra 0s.</p> <p>Example format: 000.00</p> <p>Input: 1.2345 - Matches? Yes - Output: 001.23</p>
(period)	<p>Decimal point.</p> <p>This symbol determines how many digits (0 or #) appear to the right and left of the decimal point.</p> <p>If the format contains only # symbols to the left of this symbol, numbers less than one appear with a decimal point.</p> <p>This symbol has meaning only in conjunction with the # and 0 symbols.</p> <p>Example format: ###.##</p> <p>Input: .12345 - Matches? Yes - Output: .12</p>
9	<p>Required numeric placeholder.</p> <p>If the number does not have the same number of digits as there are 9s, the system displays an error message.</p> <p>Example format: 999</p> <p>Input: 123 - Matches? Yes - Output: 123</p> <p>Input: 12 - Matches? No</p>
A	<p>Matches any alphabetic character: A–Z or a–z.</p> <p>Example format: AA</p> <p>Input: Sd - Matches? Yes - Output: Sd</p> <p>Input: 4A - Matches? No</p>

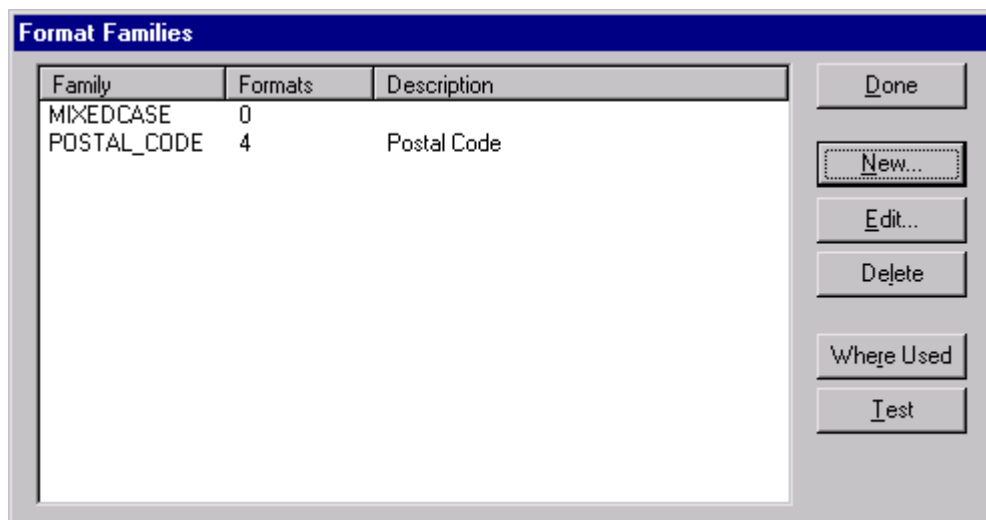
<b>Meta Symbol</b>	<b>Description</b>
<b>Z</b>	Matches any alphabetic or numeric value: <i>A-Z,a-z</i> , or <i>0-9</i> . Example format: <i>ZZ</i> Input: <i>3g</i> - Matches? Yes - Output: <i>3g</i> Input: <i>A3C</i> - Matches? No
<b>@</b>	Matches any character. Example format: <i>@@@</i> Input: <i>1q?</i> - Matches? Yes - Output: <i>1q?</i>

<b>Lexical Symbol</b>	<b>Description</b>
<b>*</b>	Matches zero or more occurrences of this character expression. Example format: <i>a*b</i> Input: <i>b</i> - Matches? Yes - Output: <i>b</i> Input: <i>ab</i> - Matches? Yes - Output: <i>ab</i> Input: <i>aab</i> - Matches? Yes - Output: <i>aab</i>
<b>+</b>	Matches occurrences of this character expression. Example format: <i>a+b</i> Input: <i>b</i> - Matches? No Input: <i>ab</i> - Matches? Yes - Output: <i>ab</i> Input: <i>aab</i> - Matches? Yes - Output: <i>aab</i>
<b>[ ]</b>	Denotes a character expression that matches the input character if the input character is the same as any character in the list enclosed by square brackets. If the expression matches, the input character appears. The system interprets all characters enclosed in square brackets as literal characters. Example format: <i>[ab]c</i> Input: <i>ac</i> - Matches? Yes - Output: <i>ac</i> Input: <i>bc</i> - Matches? Yes - Output: <i>bc</i>

<b><i>Lexical Symbol</i></b>	<b><i>Description</i></b>
{ }	<p>Denotes a character expression that matches the input character if the input character matches any character in the list enclosed by curly braces.</p> <p>If the expression matches, the first character in the list (not the input character) is copied to output. All characters enclosed in curly braces are interpreted as literal characters.</p> <p>Example format: {ab}c</p> <p>Input: <i>ac</i> - Matches? Yes - Output: <i>ac</i></p> <p>Input: <i>bc</i> - Matches? Yes - Output: <i>ac</i></p>
?	<p>This expression is optional. It is copied to output only if it appeared in input.</p> <p>Example format: a?b</p> <p>Input: <i>ab</i> - Matches? Yes - Output: <i>ab</i></p> <p>Input: <i>b</i> - Matches? Yes - Output: <i>b</i></p>
!	<p>This expression is optional. It is copied to output regardless of whether or not it is matched.</p> <p>Example format: a!b</p> <p>Input: <i>ab</i> - Matches? Yes - Output: <i>ab</i></p> <p>Input: <i>b</i> - Matches? Yes - Output: <i>ab</i></p>
~	<p>This expression is optional. It is copied to output regardless of whether or not it is matched.</p> <p>Example format: a~b</p> <p>Input: <i>ab</i> - Matches? Yes - Output: <i>b</i></p> <p>Input: <i>b</i> - Matches? Yes - Output: <i>b</i></p>
\	<p>Example format: a\b</p> <p>Input: <i>ab</i> - Matches? No</p> <p>Input: <i>a?b</i> - Matches? Yes - Output: <i>a?b</i></p>
( )	<p>Groups expressions.</p> <p>Example format: (abc)!99</p> <p>Input: <i>abc12</i> - Matches? Yes - Output: <i>abc12</i></p> <p>Input: <i>12</i> - Matches? Yes - Output: <i>abc12</i></p>

## Changing Format Families

Custom field formats enable you to create your own format definitions using format notation and apply them to fields. These formats are organized into format families, which can include one or more unique formats.



Format Families dialog box

To change format families:

1. Select Tools, Miscellaneous Definitions, Field Formats to access the Format Families dialog box.

The Format Families dialog box lists the defined format families and enables you to edit, delete, or create new families. You can also determine where each family is used in your applications and perform tests.

2. To define a new format family, click the New button.

The New Family dialog box appears.

3. Enter a name for your new format family, and click OK.

The Formats dialog box appears.

4. In the Stored group box, enter a name and format.

This defines how the format family is to be stored in the database and printed in reports.

5. Enter the format notation that you want to use for storing the data in the Format field.

Because a stored format cannot exist alone, a display format is always required. If you are adding to a family that has existing stored formats, a display name and format pair is optional. PeopleSoft Application Designer automatically places all of the current display formats in the new stored format. The display format default is the stored format that you can redefine later.

6. In the Display group box, enter a name for the format.

This defines how the format family appears online.

7. Enter the format notation that you want to use for displaying the data in the Format field.

8. Click the Add button to add the new format family, and then click OK when you are finished.

## Max Display Length

Max Display Length enables you to override the default length that PeopleSoft Application Designer uses to determine the length of the edit field in custom format. This is useful when the displayed data is longer than the defined field length. For example, you might store data as 999 (field length in the database is 3 characters), but the display format of the data is 9-9-9 (5 characters).

PeopleSoft Application Designer uses the field length of 3, by default, for the edit field, which is not long enough to hold the 5 characters of the reformatted data. By entering five in the Max Display Len (maximum display length) field, the page uses 5 characters for the edit field without requiring that the field length in the database be 5 characters long.

## Editing Formats

The Edit Formats dialog box presents a tree where you define the stored and display formats for a family and all options and attributes. This dialog box enables you to:

- Add, update, and delete stored and display formats.
- Enter a description for the family.
- Specify options, such as uppercase and smart punctuation, for the family.

Click OK to save changes.

Formats dialog box

This section discusses how to:

- Add a new stored format.

- Add a new display format to a stored format.
- Update a display format.
- Delete stored formats.

### ***Adding a New Stored Format***

To add a new stored format:

1. Open the format family to which you want to add a format.
2. Select the format family name.
3. Enter the stored and display names and format notation.
4. Click Add.

The new format appears in the tree below the existing formats.

### ***Adding a New Display Format to a Stored Format***

To add a new display format to a stored format:

1. In the tree, select the stored format name to which you want to add a display format.
2. In the Display group box, enter a new name and format notation.
3. Click the Add button to add the display format.

The new display format appears in the tree, attached to the stored format to which it belongs.

### ***Updating a Display Format***

To update a display format:

1. Select the display format in the tree view.

---

**Note.** Clicking the Update button applies only to display formats.

---

2. Select a display format and enter the enabled fields.
3. Click Update to update the display.

### ***Deleting Stored Formats***

To delete a stored format:

1. Select the display format.
2. Click the Delete button.

PeopleSoft Application Designer displays a warning message indicating that deleting a display format might strand stored data. Select Yes to continue with the deletion.

---

**Note.** Stored formats cannot be deleted directly. Therefore, all display formats for a specific stored format must be deleted first. This ensures that data is not stranded in the database under an invalid format. When the last display format is deleted for a stored format, the stored format is automatically deleted.

---

## Using Family Options

Family options include:

- Smart punctuation
- Make uppercase

### **Smart Punctuation**

Smart punctuation automatically adds or removes punctuation characters to the data that users enter. For instance, if you enter a phone number like *8005551212* and press tab to exit the entry field, it is reformatted as *800-555-1212*.

The system uses the punctuation characters specified in the punctuation list to strip out unwanted input and to modify the pattern specified before applying the pattern to the data.

---

**Note.** Smart punctuation and the standard set of characters are enabled by default.

---

You can modify the standard set of characters by selecting *Custom* in the options drop-down list box and typing another set of punctuation characters to use. Smart punctuation processes the characters defined in the standard set or custom set as follows:

- The input has all characters in the punctuation set stripped out—similar to ignoring white spaces.
- The pattern has all characters in the punctuation set expanded if they are not found in the input.

This is done by enclosing each punctuation character that is found in the punctuation set with "[ *puncchar* ]!"

- The formatter is run on the modified data to apply the modified pattern.

---

**Warning!** Use meta characters and lexical characters in the punctuation sets with caution. Step two of the transformation process modifies the format pattern so that PeopleSoft Application Designer might sometimes convert special characters to literals (when they're placed inside the left or right brace).

---

For example:

- Input: *(800)555-1212*.
- Display pattern: *999-999-9999*.
- Punctuation set: *()-*.
- Stored pattern: *9999999999*.
- The *"(", ")",* and *"-"* are stripped out of the input because they appear in the punctuation set.

Now the input looks as follows: *8005551212*.

The display pattern is modified to `999[-]/!999[-]/!9999` because the `"-"` is in the punctuation set. The transformed data (`8005551212`) is matched against the display pattern of `999[-]/!999[-]/!9999`. The match is successful, so the raw data is transformed into `800-555-1212`.

Now, the input data has successfully matched the expected display format. The next step is to transform this data into what we want stored. In this case, the stored format has no dashes, as in `9999999999`.

The transformed data (`800-555-1212`) is stripped of the smart punctuation characters (`8005551212`), and the pattern of the stored format is `9999999999`. A match is made so that the system stores `8005551212` in the database.

Finally, the stored data is reformatted back to the display using `8005551212` and the display pattern. Output to the display is reformatted to `800-555-1212`.

### ***Make Uppercase***

Custom format fields support mixed case. With uppercase activated, PeopleSoft Application Designer transforms the data into uppercase when you press tab to exit the field.

## **Testing Formats**

The Test Format dialog box enables you to test a specific format or unformatted path for a family without having to build a page.

Test Format dialog box

### **Stored Format Name and Display Format Name**

Select a stored and display format pair to test.

### **DB->Display**

Click this button to process the input field like data from the database. The system applies the stored format pattern to the input, then applies the display format pattern to the result. The output appears in the Transform field, which is the same output displayed for a field using this format.

**Display->DB**

Click this button to process the input field like user input from a page. The system applies the display format pattern to the input, then applies the stored format pattern to the result. The transformed output appears in the Transform field. This output reflects what is stored in the database if the operation is successful. On an actual page, when the user presses tab to exit the field, the system calls the DB->Display function to reformat the user input for the display.

---

## Using the Translate Table

This section provides an overview of the Translate table and discusses how to:

- Add values to the Translate table.
- Change translate values.
- Delete translate values.
- Save the Translate table.

## Understanding the Translate Table

The Translate table is a prompt table, which is similar to an all-purpose data dictionary, to store values for fields that don't need individual prompt tables of their own. As a general rule, store field values in the Translate table if the field meets the following criteria:

- Field type is *Character*.
- Field length is 1 to 4 characters.
- Field values consist of a relatively small, static set of values that are not maintained by the user.
- No other fields relate to this field.

---

**Note.** If the only values for a field are *Y* (yes) and *N* (no), you don't need to enter them in the Translate table. The Translate table comes with a field for Y and N named PSYESNO. When you select the Yes/No Table Edit, the system automatically points to the PSYESNO field in the Translate table.

---

### ***Example of When to Use the Translate Table***

Consider the relative attributes of a department ID field and a gender field in a company database. In the following table, Department ID is not a good case for using the Translate table, because it requires its own prompt table. However, the Gender field is a good case for using the Translate Table:

<b><i>Characteristic</i></b>	<b><i>Department ID</i></b>	<b><i>Gender</i></b>
Field type.	Character.	Character.

<b>Characteristic</b>	<b>Department ID</b>	<b>Gender</b>
Field length.	Longer than 3 characters.	1 character.
Field values.	Could be many values, 20 or more.	Two possible values.
Other related fields.	Related to other fields, like department manager.	Not related to other fields.
Requires maintenance.	Yes, users might have to add departments.	No, gender is a static value.

### **Translate Table Attributes**

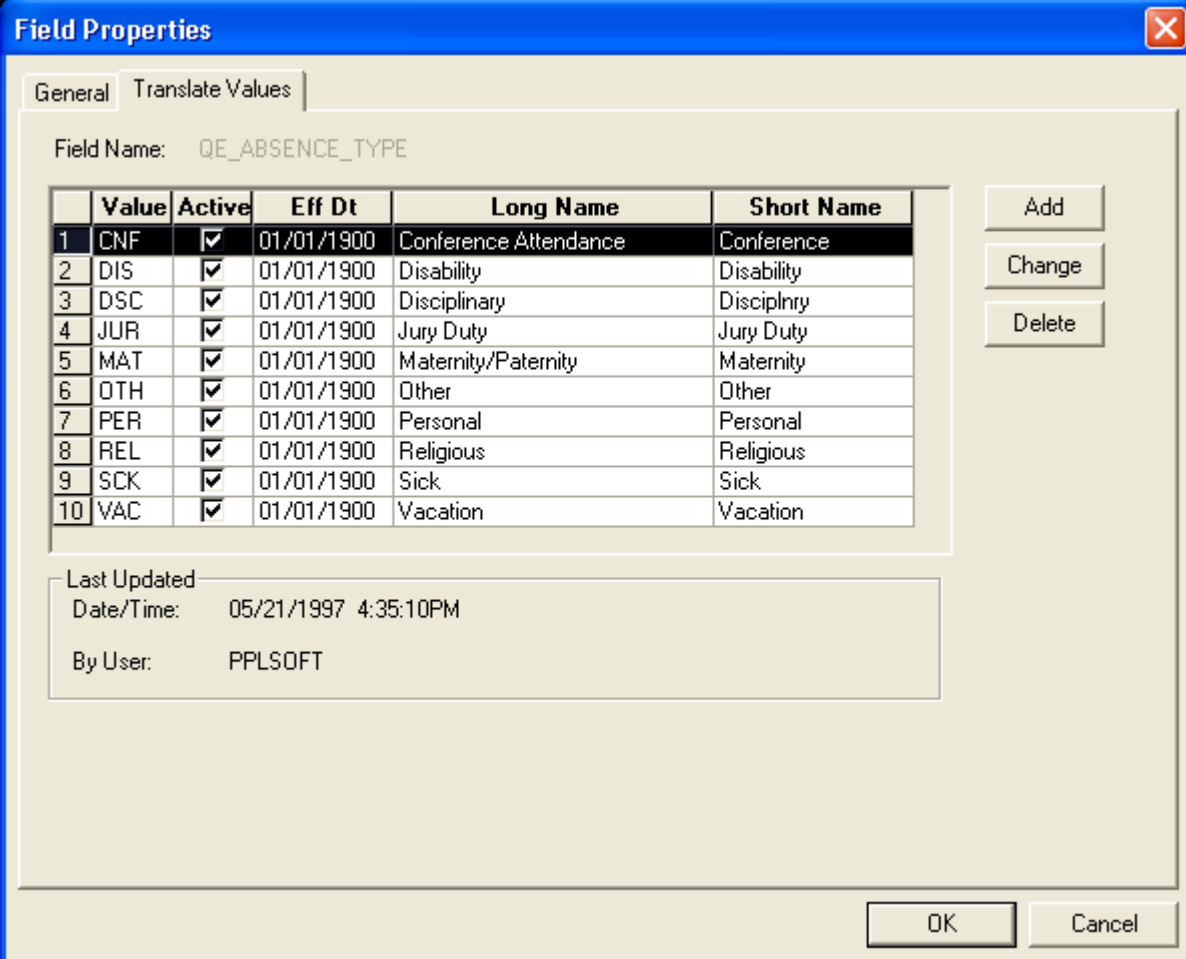
The system maintains the structure of the Translate table and you supply the data. This table contains the following fields:

<b>Field Name</b>	<b>Description</b>
FIELDNAME	Field name, such as ABSENCE_TYPE.
LANGUAGE_CD	Language code.
FIELDVALUE	Value for the field.
EFFDT	Effective date.
VERSION	Internal version number (system-maintained).
EFF_STATUS	Status—active or inactive.
XLATLONGNAME	Thirty-character description; used as a label on pages and reports.
XLATSHORTNAME	Ten-character description; used as a label on pages and reports.
LASTUPDDTTM	DateTime field showing the last time that a field was updated.
LASTUPDOPRID	User ID of the user who most recently updated the field.

If you must define more information about a field other than the preceding attributes, create a separate prompt table for the field instead of adding it to the Translate table.

When the user presses the prompt button to prompt for valid values in a field on a page, the system displays a list of the translate values for a field.

Each value in the Translate table has an effective date, and the date must be earlier than the effective dates of any rows that reference the value. The translate values delivered by PeopleSoft all have an effective date of January 1, 1900. Remember this if you add new translate values for a field.



The image shows a screenshot of the 'Field Properties' dialog box, specifically the 'Translate Values' tab. The dialog box has a blue title bar with the text 'Field Properties' and a close button. Below the title bar, there are two tabs: 'General' and 'Translate Values', with 'Translate Values' being the active tab. The 'Field Name' is set to 'QE\_ABSENCE\_TYPE'. Below this, there is a table with 6 columns: 'Value', 'Active', 'Eff Dt', 'Long Name', and 'Short Name'. The table contains 10 rows of data, each representing a different type of absence. To the right of the table are three buttons: 'Add', 'Change', and 'Delete'. Below the table, there is a section for 'Last Updated' information, including 'Date/Time' and 'By User'. At the bottom of the dialog box are 'OK' and 'Cancel' buttons.

	Value	Active	Eff Dt	Long Name	Short Name
1	CNF	<input checked="" type="checkbox"/>	01/01/1900	Conference Attendance	Conference
2	DIS	<input checked="" type="checkbox"/>	01/01/1900	Disability	Disability
3	DSC	<input checked="" type="checkbox"/>	01/01/1900	Disciplinary	Disciplinary
4	JUR	<input checked="" type="checkbox"/>	01/01/1900	Jury Duty	Jury Duty
5	MAT	<input checked="" type="checkbox"/>	01/01/1900	Maternity/Paternity	Maternity
6	OTH	<input checked="" type="checkbox"/>	01/01/1900	Other	Other
7	PER	<input checked="" type="checkbox"/>	01/01/1900	Personal	Personal
8	REL	<input checked="" type="checkbox"/>	01/01/1900	Religious	Religious
9	SCK	<input checked="" type="checkbox"/>	01/01/1900	Sick	Sick
10	VAC	<input checked="" type="checkbox"/>	01/01/1900	Vacation	Vacation

Last Updated  
 Date/Time: 05/21/1997 4:35:10PM  
 By User: PPLSOFT

Field Properties dialog box: Translate Values tab

## Adding Values to the Translate Table

Access the Add Translate Table Value dialog box.

To add translate values to the Translate table:

1. From a field definition window, select File, Definition Properties.

You can also right-click and select *Field Properties*, or press alt+enter.

2. From a record field definition window, right-click the field, and select *View Translates* from the pop-up menu.
3. Select the Translate Values tab.

The Translate Values dialog box displays existing values for the field and enables you to add, change, or delete values. In the Last Updated box, you see information—date, time, and user ID—about the last update for the selected translate value.

4. Click Add to define a new value.
5. Enter the field value, effective date, long name, and short name that your users enter on the page.

<b>Field Value</b>	Enter the translate value for the field. The system automatically sorts values in ascending order as you enter them. If you enter three translate values with field values <i>A</i> , <i>T</i> , and <i>C</i> , they appear in the Translate Values dialog box as <i>A</i> , <i>C</i> , <i>T</i> .
<b>Effective Date</b>	Enter the date on which you want this value to take effect. The default value is today's date. If you want the effective date to predate all rows on your database, enter <i>01011900</i> (January 1, 1900).
<hr/> <p><b>Note.</b> The Add button only allows you to add a new, distinct translate value. To add a different effective date for the same translate value, select the translate value and use the Change button. As long as the effective date is changed, the system saves the modified entry.</p> <hr/>	
<b>Inactive</b>	When a value for a field becomes obsolete, PeopleSoft recommends that you select the Inactive check box rather than deleting the value. Deactivating a value enables fields in the database that still contain the value to use the correct long and short names. If you delete an obsolete code from the Translate table and you still have records in the database that contain that value, you must change all of those values to active values.
<b>Long Name</b>	Enter up to 30 characters.
<b>Short Name</b>	Enter up to 10 characters. If you leave this field blank, the system automatically copies the first 10 characters of the long name into this field. PeopleSoft recommends that you use mixed case to improve readability.

## Changing Translate Values

Sometimes the meaning of a translate value changes, but you still need to retain both values in the Translate table. For example, suppose that your PROFICIENCY field has a value of *E* (extremely high). You've been in production for five years, when upper management decides to change the meaning of the value to *Exceptional*, as of January 1, 1998.

You can accommodate this type of change by creating a second active entry for the same translate value. The first value is valid in the time range between the effective dates of January 1, 1990 and December 31, 1997. The second entry is valid from January 1, 1998 onward. You don't want to delete the old entry because the database might contain rows that predate January 1, 1997 and contain this value. You might also maintain the old definition for historical reporting.

To change an existing translate value, select the value and click Change, or double-click the value. The system displays the Change Translate Table dialog box, in which you can make the necessary changes.

## Deleting Translate Values

Use caution when deleting a translate value, unless you are removing an invalid value that was entered by mistake. To deactivate a value because it is no longer used or its meaning has changed, either change the long and short names, or change the status to *Inactive*. To delete the value, select it and click the Delete button. The row disappears.

## Saving the Translate Table

After you have made all of your changes to the Translate table, you must save the *entire field definition*. There is no save option specifically for translate values. Click OK from the Field Properties dialog box, and then save the field definition by clicking the Save button or selecting File, Save.

To save the Translate table, you must be authorized to modify field definitions. There is a special *Translates Only* security access level for fields that allows translate values to be updated, but not other field attributes.



## Chapter 5

# Planning Records, Control Tables, and TableSets

The first step in the application development process is planning. You must consider as a system designer how to store, retrieve, manipulate, and process data that is stored in the tables in your application database.

This chapter provides an overview of the planning process and discusses how to share tables.

---

## Understanding the Planning Process

This section discusses:

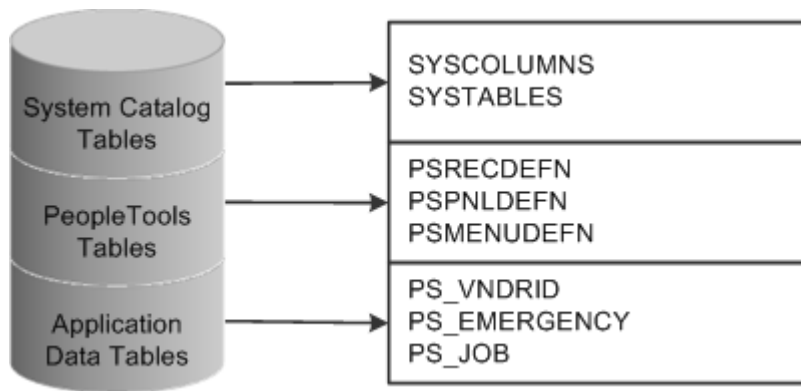
- Table-based systems.
- Normalized relational databases.
- Record definition planning.
- Effective dates.
- Control tables.
- TableSets.

## Table-Based Systems

PeopleTools-based applications are table-based systems. A database for a PeopleTools application contains three major sets of tables:

- System Catalog tables store physical attributes of tables and views, which your database management system uses to optimize performance.
- PeopleTools tables contain information that you define using PeopleTools.
- Application Data tables house the actual data that your users enter and access through PeopleSoft application pages.

This figure shows PeopleSoft database tables and sample names:



Tables in a PeopleSoft database

Like a spreadsheet, each of these tables contain columns and rows. Columns define the structure of how the data is stored. Rows represent the actual data that is stored in the database.

Every time that you create a new definition using PeopleTools, the system inserts rows of data into various PeopleTools tables. The entries in these tables determine the online processing of the system and what happens during imports. PeopleSoft maintains the structure of these tables. You maintain data in the PeopleTools tables related to definitions that you create or adapt using PeopleTools. The PeopleTools tables can be viewed in the PPLTOOLS project using the PeopleSoft Application Designer.

To create the application data tables that store the rows of data that your *users* manipulate:

1. Create a record definition.

In doing so you determine the structure of the table, the characteristics of the fields, and any online processing that you want to occur behind the scenes when a user enters data.

2. Apply the SQL Create option to build the SQL table in which your application data will reside based on a subset of parameters in your record definition.

During this process, the system automatically gives the application data table the same name as your record definition, prefaced with *PS\_*.

## Normalized Relational Databases

To better understand the structure of your PeopleSoft system, you should be familiar with the concept of a normalized relational database. A *normalized* table adheres to standards that are designed to improve the productivity of the database user. Normalization makes the database more flexible, allowing data to be combined in many different ways.

The standards for a normalized database are called *forms*, such as *first normal form*, *second normal form*, and so on.

### First Normal Form

The first normal form requires that a table contain no repeating groups of nonkey fields. In other words, when you're setting up a record definition, if you encounter a field that could have multiple occurrences, put that field in a separate record that is subordinate to the primary record definition (a child record). This allows unlimited occurrences of a repeating field rather than a specified number. Each row of data is uniquely identified by a primary key, which can be a single field or a group of fields that, when concatenated together, form a unique key.

For example, look at the record definition structure of the tables that we use to schedule exam times for different locations in our training database. Here are the necessary fields, in order of importance:

- LOCATION
- EXAM\_DT
- EXAM\_TIME

You know that you have multiple exam dates and times per location. You could set up record definitions to accommodate this data as follows:

<b>Record Definition</b>	<b>Fields</b>	<b>Key?</b>
LOCATION	LOCATION	Yes
EXAM	LOCATION EXAM_DT	Yes Yes
EXAM_TIME	LOCATION EXAM_DT EXAM_TIME	Yes Yes Yes

Because multiple exam dates per location might exist, we added exam dates to the second record definition (child record) that is subordinate to the first (*parent*) record. Similarly, because there can be multiple exam times per date, exam times are located in a third record definition that is subordinate to the second.

### Second Normal Form

The second normal form dictates that every nonkey field in a table must be completely dependent on the primary key. If two fields make up the key to a table, every nonkey field must be dependent on both keys together. For example, if a table has the Employee ID and the Department ID fields as keys, you wouldn't put the Department Name field in the table because this field is dependent only on the Department ID field and not on the Employee ID field.

### **Third Normal Form**

The third normal form is a corollary to the second; it requires that a nonkey field not be dependent on another nonkey field. For example, if a table is keyed by the Employee ID field, and Department ID is a nonkey field in the table, you wouldn't put the Department Name field in the record because this field is dependent on a nonkey field (Department ID). This is why the Department Name field would be found only on the table that is keyed by the Department ID field, not on any other that contains Department ID.

With the third normal form, you store shared fields in tables of their own and reference them elsewhere. For example, you wouldn't put the Department Name field in every record definition in which the Department ID field appears. Instead, you would create a prompt table of department IDs and department names. Similarly, you would create a prompt table of job codes and job titles instead of putting a job title in every employee's record.

---

**Note.** When designing record definitions, adherence to the third normal form is recommended to increase flexibility and reduce data redundancy.

---

## **Record Definition Planning**

Before you begin to create record definitions, you should have a clear picture of how you plan to use the record definition, the fields that it will contain, special edits that you would like to see performed on the record definition, or specific fields in the definition.

You actually define two layers of information:

- Record level
- Field level

At the record level, determine the ultimate purpose of the record definition and how it will be used in the system. Is it destined to define an underlying SQL table to hold data? Are you building a view to join or retrieve information from other tables? Do you need a temporary work record where you can store derived data?

You can audit record-level changes, as opposed to individual fields contained in the record definition—an efficient alternative if you plan to audit several fields. More sophisticated use of record definitions, such as sharing information in TableSets and multilanguage controls, are also established at the record level.

At the field level, plan the details of what types of fields to add. Should they be character fields or number fields? Should automatic formatting be used? What are the keys to the data stored in the database? Which fields should you audit? Do you want to specify prompt tables so that users can select from lists of valid values that are stored elsewhere in the database?

In most cases, if you are creating a record definition for a SQL table, you don't have to worry about record-level definitions for parameters and conditions. Unless you change how a record definition is used, the system automatically assumes that you are defining a record definition for an underlying SQL table.

## **Effective Dates**

Effective dates enable you to keep historical, current, and future information in tables. You can use the information to review the past and plan for the future. There are three types of effective dates:

<b>Future</b>	Data rows that have effective dates that are after the system date, which is usually today's date.
<b>Current</b>	Data row with the most recent effective date that is closest to today's (system) date, but not a future date. Only one row is the current row.
<b>History</b>	Data rows that have effective dates before the current data row.

The EFFDT (effective date) field has special properties related to the processing of effective dates on rows and should be used only when needed.

Unlike regular date fields, which you can use anywhere in the system, use the EFFDT field only on record definitions for which you want to maintain data history—future, current, and past—to store rows of data in sequence. This enables you to store multiple occurrences of data based on when it goes into effect.

For effective-dated rows, you can have multiple occurrences of future and history, but only one current row of data.

EFFDT is almost always a key and almost never a list item. Activate the Descending Key attribute so that the row with the most recent effective date appears first on pages. You might enter %DATE (current system date) as the default constant for this field.

---

**Note.** Alternatively, you can use %CLIENTDATE as the default constant for the date field. %CLIENTDATE adjusts the date as appropriate to the time zone of the browser.

---

To enable you to track an accurate history of your effective-dated information, the system invokes special logic when you access a record definition that contains EFFDT. The action that you select dictates whether you can access the row type and what you can do with each type of row:

<i><b>Action Type</b></i>	<i><b>View</b></i>	<i><b>Change</b></i>	<i><b>Insert New Rows</b></i>
Update/Display	Current, Future	Future only	Effective Date Greater Than the Current Row
Update/Display All	History, Current, Future	Future only	Effective Date Greater Than the Current Row
Correction	History, Current, Future	All Existing Rows	Add New Rows with No Effective Date Restrictions

---

**Note.** For records that do not contain EFFDT, all actions (Update/Display, Update/Display All, and Correction) operate the same way—they retrieve all existing rows for the specified keys.

---

When you're running a page with effective-dated records and you insert a row, the system copies the contents of the prior row into the new row to save you keying time. On a large effective-dated table, you don't want to reenter all of the data when only a single field changes. Also, anytime you insert an effective-dated row using PeopleCode, the same copying of the prior row's contents takes place.

**Effective Status**

In prompt tables, EFF\_STATUS (Effective Date Status) usually accompanies EFFDT. When used with EFFDT, it's part of the mechanism that enables the system to select the appropriate effective-dated rows.

You can also use EFF\_STATUS by itself as a simple status field, but don't change the translate values. They must be *A* (active) and *I* (inactive) for EFFDT to work properly. If you need a status field with different values, use or define a different field.

**Effective Sequence**

The EFFSEQ (Effective Sequence) field serves different purposes, depending on whether it's paired with EFFDT. If EFFSEQ isn't paired with EFFDT, EFFSEQ has no special function and can be used as a simple sequencing field wherever you need one.

If EFFSEQ is paired with EFFDT, it enables you to enter more than one row with the same effective date. You assign a unique sequence number to each row that has the same effective date. Do not make EFFSEQ a required field—*unrequired* allows the first EFFSEQ to be zero. Select Display Zero in the page definition to have zeros appear on the page.

For example, suppose that you want to enter both a transfer and a pay rate change for an employee, and both actions are effective on the same day. Enter the transfer on the job data pages as usual, and leave the Effective Sequence Number field as *0* (zero). Then, insert a row to enter the change in pay rate. This time, the effective date is identical to the previous row, but enter *1* in the Effective Sequence Number field.

**See Also**

*Enterprise PeopleTools 8.50 PeopleBook: PeopleCode Language Reference*, "System Variables," %ClientDate

*Enterprise PeopleTools 8.50 PeopleBook: PeopleCode Language Reference*, "System Variables," %Date

**Control Tables**

Control tables store information that controls the processing of an application. This type of processing might be consistent throughout an organization (in which case the entire organization shares the same control information), or it might be used only by portions of the organization for more limited sharing of data.

**Sharing One Set of Common Values**

The first type of sharing is to create one table that everyone shares; it stores common information that is valid for *all* users, such as a country table to store country codes or a department table to store department codes. Such control tables are ordinarily maintained centrally because the data is shared throughout the entire organization.

### Sharing Common Values in Overlapping Plans

What do you do if the codes that are stored in a table are valid only for *some* users? Consider benefit plans, for example. Typically, you store information for benefits plans in a plan table. However, not all plans are valid for all employees. It might depend on whether they are full-time or part-time, union or nonunion. Some plans might overlap; some might be appropriate for all employees and others only for some. In a relational database, you don't want to define the same plan value—and associated data—more than once.

In this case, you can easily resolve the problem by using two tables. The first is the plan table, which stores the relevant data for each plan. The second table defines which plans are valid for various benefit programs or groups of plans. For example, one benefit program might be valid for nonunion employees, and another benefit program might contain the plans as negotiated with a union.

This table shows how you might set up the two tables to reflect the benefits offered:

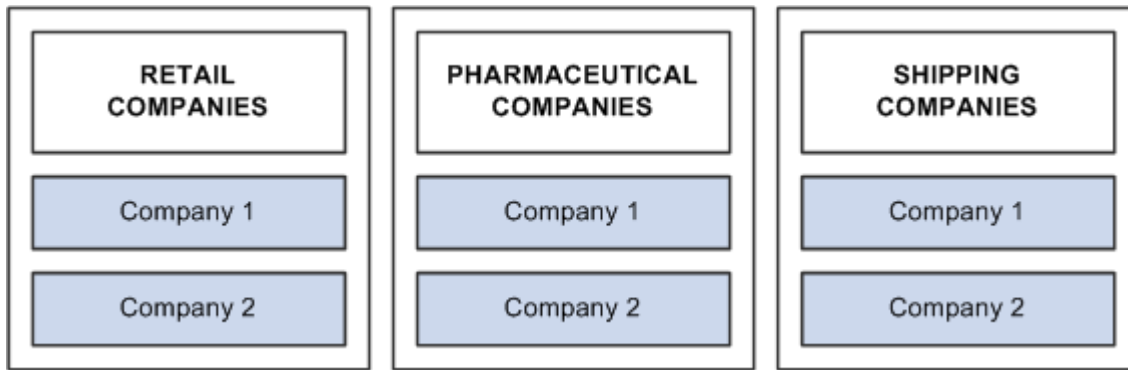
<b>Benefit Program Table</b>		<b>Benefit Plan Table</b>			
Key	Valid Values	Key	Description	Field	Field
Nonunion Program	Plan 1	Plan 1	Health	...	...
	Plan 2	Plan 2	Life	...	...
Union Program	Plan 2	Plan 3	Savings	...	...
	Plan 3	Plan 4	Health-Union	...	...
	Plan 4			...	...

These tables are ordinarily centrally maintained, because the data is being shared by various groups in the organization.

## TableSets

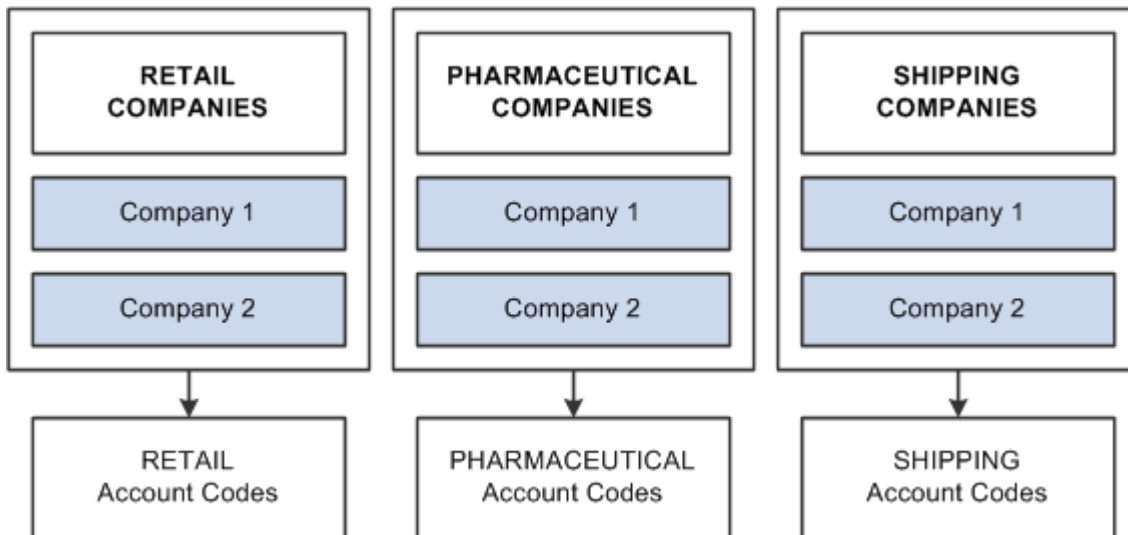
When *none* of the information stored in control tables is valid for all users, but the structure of these common tables is the same, you can set up a way to share multiple sets of values. For example, a multicompany organization must store completely different sets of accounting codes for its various operating entities, and the data for these accounting codes is maintained in a set of relevant control tables. The actual data values differ, but the structure of the control tables remains the same. PeopleTools enables you to share sets of values in a control table through *TableSets*.

To better understand TableSets, consider an organization that has two retail stores with common accounting codes, two pharmaceutical firms with another set of accounting codes, and two shipping firms with yet another set of codes.



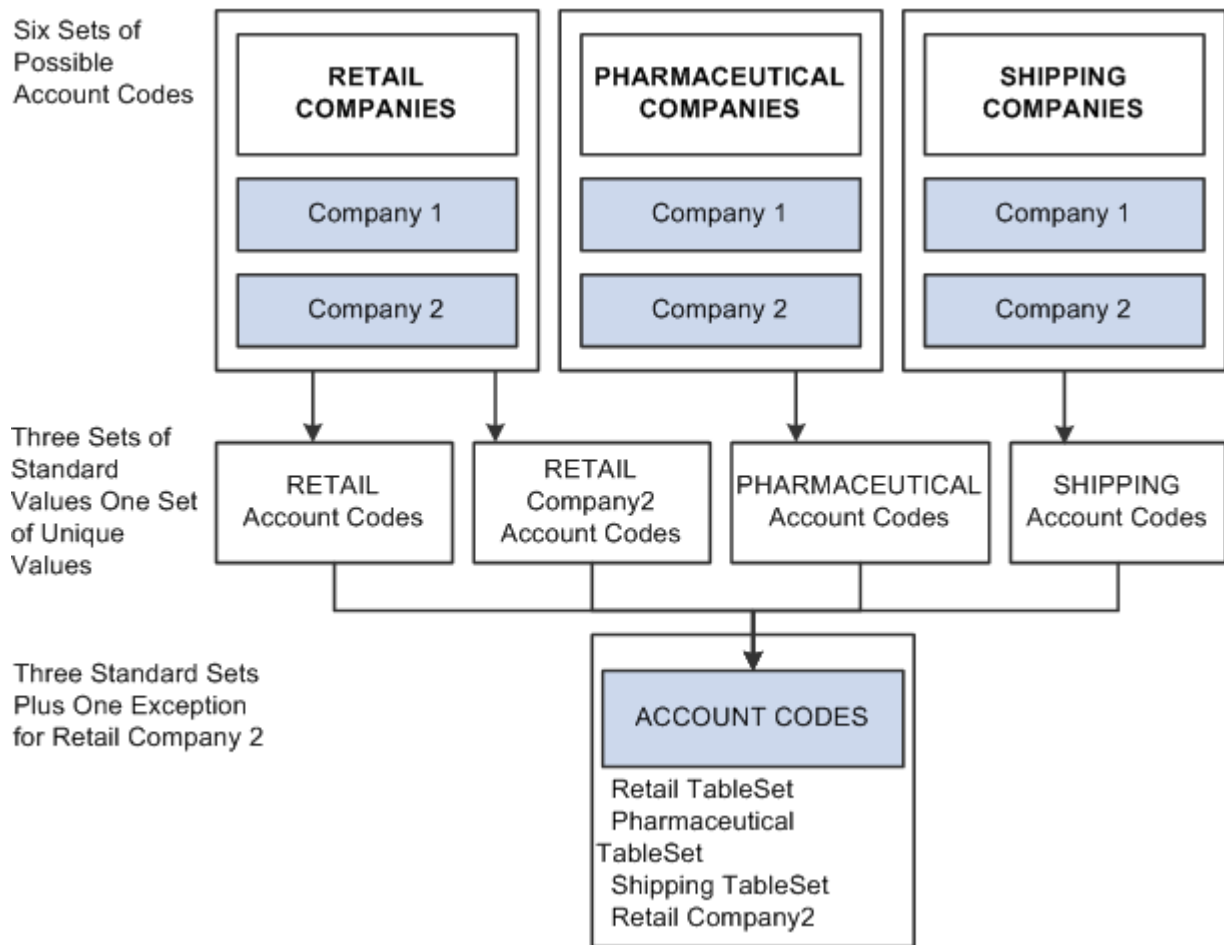
Maintaining multiple account codes for multiple companies

If each of these companies has completely different accounting codes, you can establish six different sets of account codes to be maintained by each company. If they all have exactly the same accounting codes, you can limit them to one set of values. However, the reality is usually somewhere in between. That is, there is one set of account codes for each type of business: retail, pharmaceutical, and shipping. Rather than having six different companies maintaining separate copies of this common data, you can reduce the number to three sets.



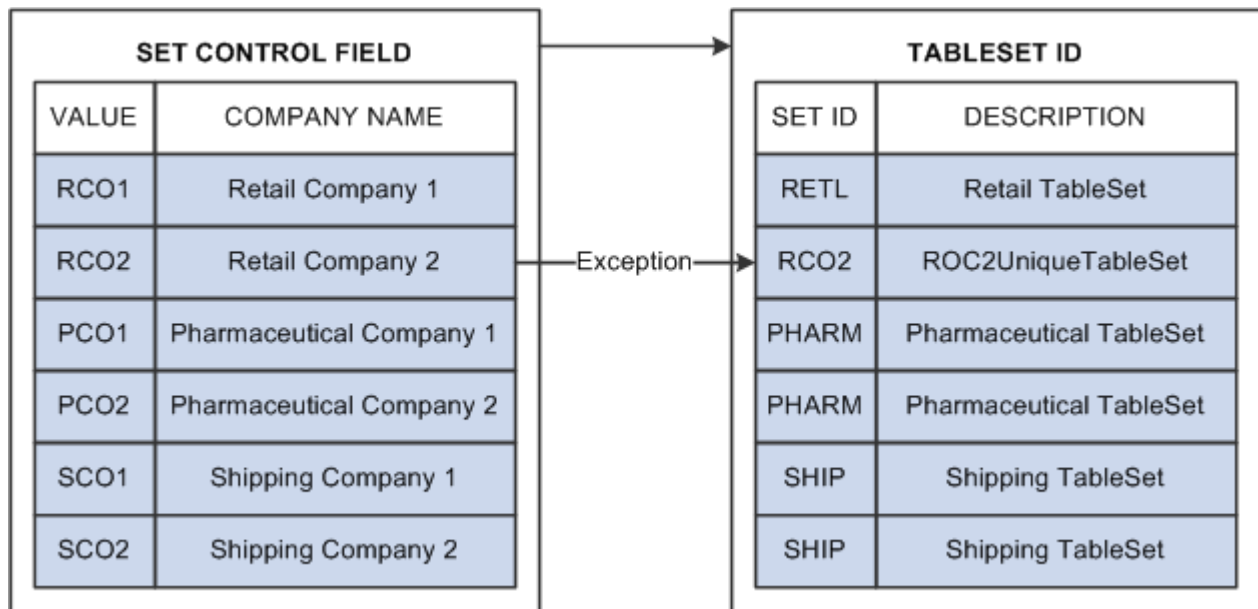
Sharing multiple account codes among companies

You can also handle exceptions. Suppose that Retail Company 2, a recently acquired company, has its own unique set of account codes. A separate set of values should be maintained for this company as an exception to the retail rule.



#### Sharing account codes among companies with exceptions

When you share tables in PeopleTools applications, you add the `setID` field as an additional key or unique identifier to the table that you want to share. This key identifies the sets of information in the table that are shared by multiple companies or business units under your corporate umbrella. You then specify a *set control* field, which identifies which fields map between the original key and the TableSets. You can specify any field that logically identifies the TableSet. In this example, you might assign the `Company` field as the set control.



Linking set controls and TableSets

### Sharing Groups of Record Definitions

While this example illustrates how you might share data values for a single table—Account Codes—you typically share data that is stored in many tables that are based on the same TableSets. To minimize the overhead of defining TableSets, you can define *record groups* that share table data in a similar manner. For example, rather than use the TableSets that you establish for accounting codes solely for the Accounting Code table, you can group all accounting-related tables into one record group.

### TableSets and PeopleSoft Applications

Some PeopleSoft applications already take full advantage of TableSets and table sharing. Throughout the PeopleSoft Financials and HRMS product lines, TableSets are used extensively, in most cases triggered by business unit.

## Sharing Tables

This section provides an overview of table sharing and discusses how to:

- Add the SetID field to the record definition.
- Define a set control field as the field controlling the assignment of TableSets.
- Modify the set control field.
- Create setIDs.
- Define record groups to identify the tables and subordinate (child) tables that are affected.
- Define TableSet controls.

- Share trees.

## Understanding Table Sharing

To share tables, you must modify the record definitions for both the table that you want to share and the one that contains the set control field. Using the online PeopleTools Utilities for TableSets, you define the terms or controls for sharing.

For example, in the PeopleTools database, there are three companies: one U.S.-based parent company (PST) and two sister companies (CCB in the U.S. and VNB in Canada). In this organization, all of the U.S.-based companies share one set of accounting codes, and all of the Canadian-based VNB companies share another.

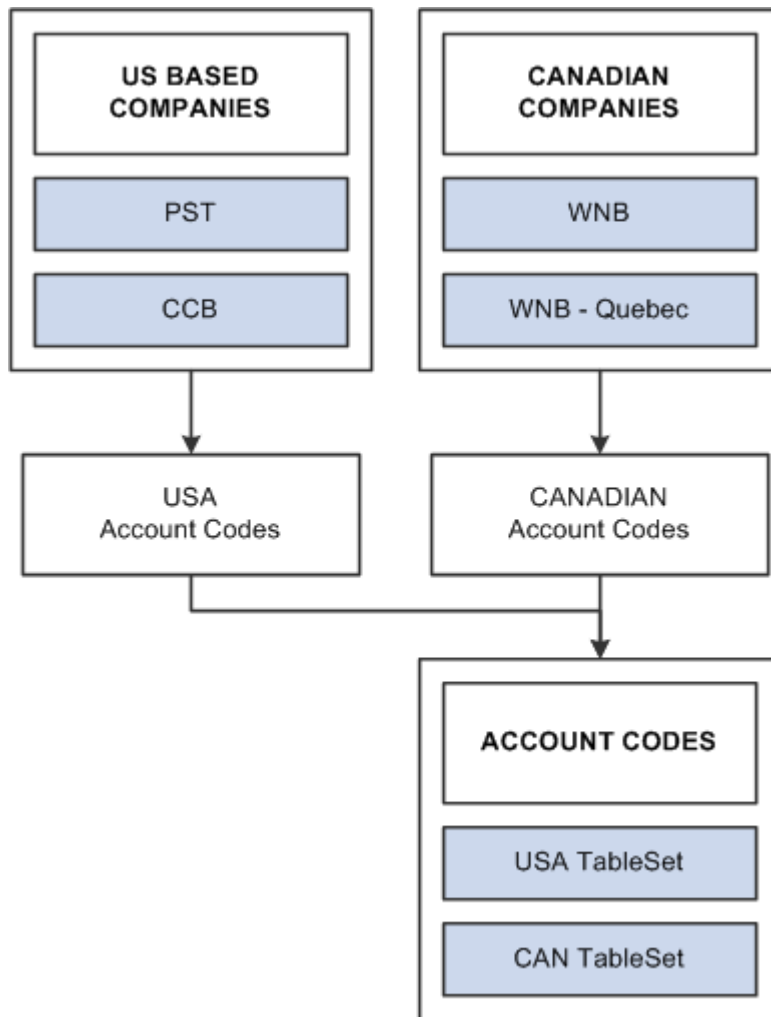
<i><b>U.S. Account Codes</b></i>	<i><b>Canadian Account Codes</b></i>	<i><b>Description</b></i>
123456789	123456789	Teller
987654321	987600000	Customer Service
CCB-4476-EXTSAL-USA	VNB-4476-EXTSAL-CDN	Extraordinary Salary Employees
CCB-4476-REGSAL-USA	VNB-4476-REGSAL-CDN	Regular Salary Employees

These two groups of companies must maintain parallel sets of accounting codes, and in some cases, they use the same code, uniquely identified for U.S. or Canadian companies.

We must set up the Account Code table so that users who access that page or prompt for valid values see only the values for their respective companies, depending on whether they're located in the U.S. or Canada.

To do this:

1. Modify the Account Code table, which will be shared, and the Company table, the values of which (CCB, PST, VNB) control the TableSets that are used.
2. Create two TableSet IDs:
  - USA: For U.S. companies.
  - CAN: For Canadian companies.



TableSets and set control fields

### Assigning Set Control Fields

If you plan to use the Table Sharing feature to add an additional high-level key to identify common sets of values and handle exception values, you enter a set control field. The set control field determines which set of values appears, based on how you define table sharing.

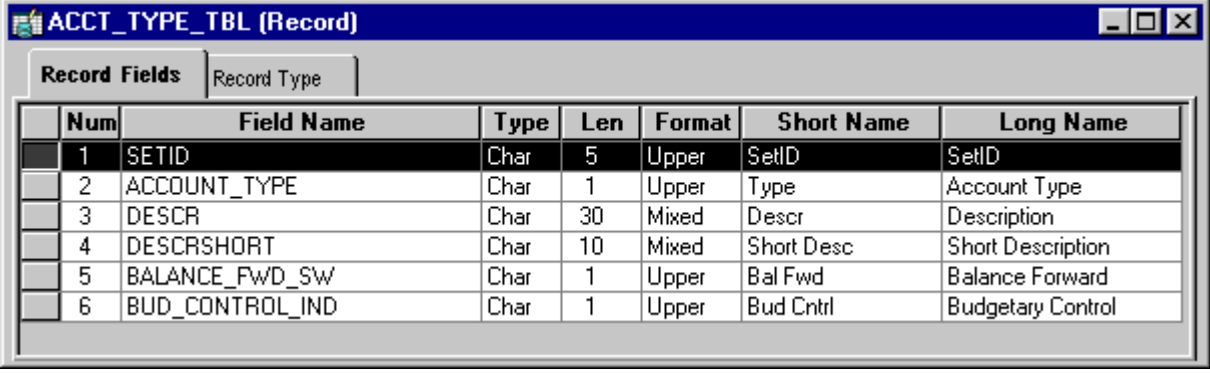
Table sharing is usually applied in multicompartment environments, in which you might want to share certain tables in some operating entities while allowing others to retain control over the contents of their own. Use the feature to maintain a single table in which you can store multiple sets of values to be used by different entities in your organization.

When you create a record definition that will be shared, you add the field `setID` as a high-level key. The `setID` value is controlled by the key that you select in the Set Control Field drop-down list box in the Record Properties dialog box. These `setIDs` identify groups of tables to be shared and enable each business entity to identify whether it wants to use the corporatewide table or maintain its own. Enter the name of the key as your set control field for each record definition that will be shared.

For example, if you have several companies in your organization, many of which share the same set of account codes, add a SetID field to the Account table record definition. In this definition, enter *Company*. as the set control field. This enables different companies to access the account values stored in the same Account table. As each company defines its set controls, it has the option to use the shared Account table or maintain control of its own.

## Adding the SetID Field to Record Definitions

Add the setID field to the record definition for table sharing purposes.



	Num	Field Name	Type	Len	Format	Short Name	Long Name
	1	SETID	Char	5	Upper	SetID	SetID
	2	ACCOUNT_TYPE	Char	1	Upper	Type	Account Type
	3	DESCR	Char	30	Mixed	Descr	Description
	4	DESCRSHORT	Char	10	Mixed	Short Desc	Short Description
	5	BALANCE_FWD_SW	Char	1	Upper	Bal Fwd	Balance Forward
	6	BUD_CONTROL_IND	Char	1	Upper	Bud Cntrl	Budgetary Control

Adding setIDs to record definitions

To add the setID:

1. Open the record definition for the table that you want to share.
2. Select New, Field, Character.
3. Enter the special field that PeopleTools provides for table sharing:

*SETID*

SetID is a 5-character field. Add it to the top of your record definition list, and define it as a required key, search and list item, with a prompt table edit against the PeopleTools SETID\_TBL.

## Defining Set Control Fields

After you add the setID field to your record, identify the set control field in Record Properties.

To define the set control field:

1. Select Edit, Definition Properties from the menu.
2. Select the Use tab.
3. Select the field that identifies the appropriate setID in the Set Control Field drop-down box.

See [Chapter 5, "Planning Records, Control Tables, and TableSets," Understanding Table Sharing, page 105.](#)

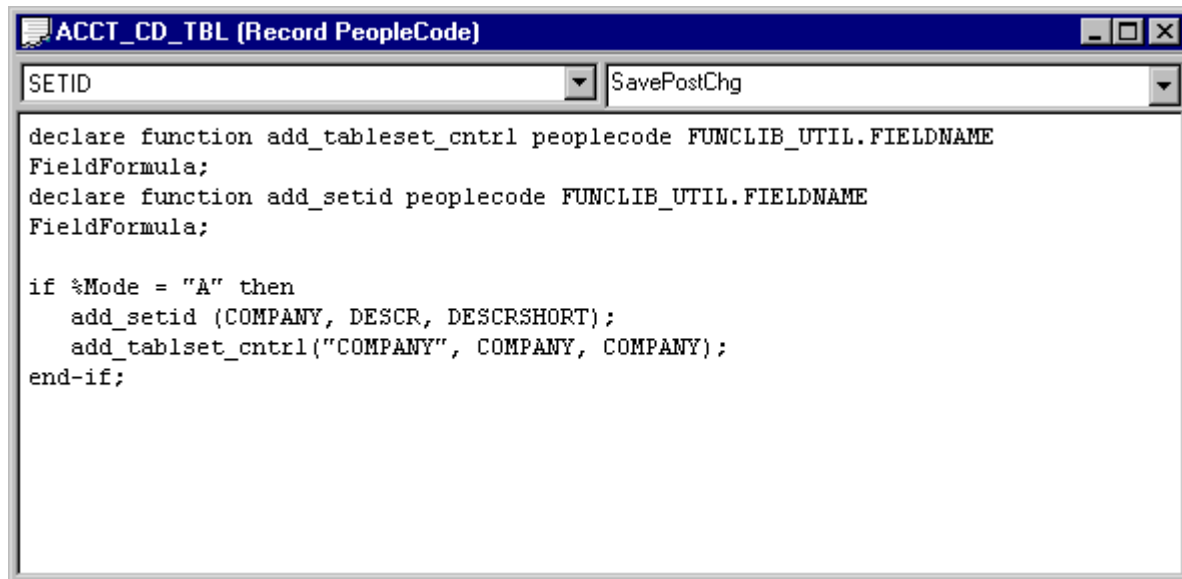
4. Click OK to save your changes.

5. Use the SQLAlter process to alter your underlying SQL table.
6. When the system prompts you for a default setID, enter the most common setID for your company.

## Modifying the Set Control Field

If you are sharing tables or using a set control field that is not already defined as such in your PeopleSoft application, you must make some modifications to the record definition for the table in which you store values for your specified set control field. For example, all PST company codes are stored in the COMPANY\_TBL.

Here, you must add PeopleCode to the set control field so that each time you add a new value, the system populates the set controls in PeopleTools utilities, where you assign the appropriate TableSets.



Attaching PeopleCode to your set control field

This program assigns a default setID with the same value as the Set Control Field value, for each new COMPANY row that you add to the COMPANY\_TBL. For example, when you add a row for CCB in the Company table, it creates a default setID of CCB. This way, each value that you add is assigned a unique TableSet until you assign shared TableSets in the Utilities window.

You can easily clone and modify this program to change the ADD\_SETID and ADD\_TABLESET\_CNTRL statements to reflect your table sharing objectives. Alternatively, if you're working with existing TableSet functionality, you can point to a model setID that is delivered with your PeopleSoft application.

### See Also

[Chapter 7, "Administering Data," Altering Tables, page 173](#)

## Creating SetIDs

After you prepare for sharing your table, you define the setIDs that logically group information. When you define setIDs, you create groups that might naturally share more than one table. For example, although CCB and VNB are planning to share only the ACCT\_CD\_TBL now, the setIDs that you set up might define logical divisions in the organization that serve as the basis for sharing all accounting-related tables. Describe your sets to give them as broad an application as is practical in your organization.

If possible, use descriptions that denote the shared table and set control field. For example, for CCB and VNB, we added two TableSet IDs, identifying them as related to accounting codes for companies.

<i>setID</i>	<i>Description</i>	<i>Short Description</i>
USA	Accounting Codes - USA Co.s	AcctCd USA
CAN	Accounting Codes - CAN Co.s	AcctCd CAN

You create SetIDs on the TableSetID page in PeopleTools, Utilities, Administration, TableSet IDs.

### See Also

*Enterprise PeopleTools 8.50 PeopleBook: System and Server Administration*, "Using PeopleTools Utilities," TableSet IDs

## Defining Record Groups

Record groups are a set of logically and functionally related control tables and views. They help enable TableSet sharing, which eliminates redundant data entry, and ensure that it is applied consistently across all related tables and views.

In the record group table, you group the record definitions for the tables that you want to share and any dependent record definitions. If you're adding a table to a PeopleSoft application, an appropriate record group might already be defined. However, if you're adding new business functions, you might need to add a new record group for the tables that you're defining. You access the record groups table in PeopleSoft Pure Internet Architecture on the Record Group page.

### See Also

*Enterprise PeopleTools 8.50 PeopleBook: System and Server Administration*, "Using PeopleTools Utilities," Record Group

## Defining TableSet Controls

As you add values for your set control, the system automatically populates the TableSet Controls 1 table with default values. You define TableSet controls in PeopleTools, Utilities, Administration, TableSet Control using the browser. Here you can enter the set control value for which you want to assign a TableSet.

For example, we use Company as our set control field and PST, CCB, and VNB are values that we're adding to the Company table. The system automatically assumes that each new set control value maintains its own set of tables and does not enable table sharing. So, the default values are the same as the set control value. In this case, when we added the value CCB to the Company table, the system populated the TableSet controls with a default setID of CCB.

Because CCB is a U.S.-based company that should share the same U.S. accounting codes, we must change the defaults to USA.

### See Also

*Enterprise PeopleTools 8.50 PeopleBook: System and Server Administration*, "Using PeopleTools Utilities," TableSet Control

## Sharing Trees

To share trees as well as tables and views, complete TableSet Controls 2 in the same way that you did TableSet Controls 1. For example, if PST and CCB want to share one department tree for organizational security and VNB another, you assign the appropriate setID for each value.

### Defining TableSet Controls for Trees

To define TableSet controls for trees:

1. Select PeopleTools, Utilities, Administration, TableSet Control in PeopleSoft Pure Internet Architecture.
2. Enter the name of the set control value for which you want to assign a TableSet in the search page.

For example, we use Company as our set control field and PST, CCB, and VNB are values that we're adding to the Company table.

Because you already assigned a default setID in TableSet Controls 1, the system displays the default setID that you assigned to this field value. If you create another TableSet for sharing trees, you can change this value.

3. Specify the tree name.

Click the prompt button next to the Tree Name field for a list of only the tree definitions with the same set control field (which has PostSave PeopleCode that recognizes it as a TableSet control field).

This prompt list comes from a SQL view of tree definitions with that set control field that haven't already been associated with a tree group.

4. In each tree row, specify the appropriate setID.
5. Save your changes.

### ***Viewing All Record Definitions Associated With a Set Control***

You can view all record definitions that are associated with a set control in the Record Group tab of the TableSet Controls page.

### ***Example of Set Controls***

After you set up and define all of your set controls, you can see the results by looking at pages on which you reference codes that are stored in a shared table. For example, on Administer Workforce, Use, Job Data 1 and 2 pages, Account Code is an optional field that is edited against the ACCT\_CODE\_TBL. In this component, the system identifies the company based on the department to which an employee is assigned. For example, Simon Schumacher is in department *10100*, which is associated with the company *CCB*.

Select the Job Data 2 tab and enter *8001* or *Simon Schumacher* in the search record dialog box to retrieve a CCB row, for which the company is defined in the department table. Here, you can see that Simon works in department *00001*, which identifies him as a CCB employee. If you click the prompt button next to the Account Code field, the system displays only those rows in the Account Code Table that are associated with USA, the setID for CCB.

Alternatively, if you access the job row for Joan Avery, an employee of VNB, and click the prompt button for the same Job Data 2 Account Code field, you retrieve only values that are associated with CAN, the setID for VNB.



## Chapter 6

# Creating Record Definitions

Fields that are grouped together as a unit are *record definitions*. A record definition represents what the underlying Structured Query Language (SQL) database tables look like and how they process data.

This chapter discusses how to:

- View record definitions.
- Save record definitions.
- Name record definitions.
- Create a new record.
- Open an existing record.
- Manipulate fields in record definitions.
- Set record properties.
- Set record field properties.

---

## Viewing Record Definitions

You can see four views of the record by selecting View from the main toolbar: Field Display, Use Display, Edits Display, and PeopleCode Display.

This section lists common elements and discusses how to:

- View basic field definitions.
- View key-related characteristics and default values.
- View editing options.
- View PeopleCode program types.
- Reorder fields.
- Size and sort columns.

## Common Elements Used in This Section

<b>Num</b> (number)	The number of the field in the order in which it is defined in the record.
<b>Field Name</b>	The name of the field in the SQL database.
<b>Type</b>	The data type of the field, such as character, number, or date.

## Viewing Basic Field Definitions

Field Display mode shows the basic field definition characteristics for fields in the record definition. Field definition characteristics are global—they affect all record definitions in which the field is used. To open the associated field definition, right-click while in Field View and select View Definition.

<b>Len</b> (length)	Indicates the maximum length of the field, including decimal places.
<b>Format</b>	Notes special formatting for the field, such as mixed case, date, international phone number, or RawBinary.
<b>Short Name</b> and <b>Long Name</b>	Displays the short and long names of the field as users see it on pages.

## Viewing Key-Related Characteristics and Default Values

Use Display mode shows key-related characteristics and default values for fields that determine how fields are used in a record. The use characteristics might differ for fields that are used on more than one record definition. Double-click the field to access the Record Field Properties dialog box in which you define these parameters.

<b>Key</b>	<p>Indicates whether the field is a key to the record definition. Key fields are included in an index that is automatically created during the table build process.</p> <p>The word "Not" in this column indicates that this field cannot be used as a chart field. This can only be set through PeopleCode using the SetDBFieldNotUsed function.</p> <p>See <i>Enterprise PeopleTools 8.50 PeopleBook: PeopleCode Language Reference</i>, "PeopleCode Built-in Functions," SetDBFieldNotUsed.</p>
<b>Ordr</b> (order)	Indicates the order of key fields in the index. This order can be configured from the Index dialog box and is reflected in this column.
<b>Dir</b> (direction)	Indicates the order in which the key field indexes are created in the database: ascending or descending.

<b>CurC</b> (currency control)	The currency format for this field is controlled by the currency code in another field.
<b>Srch</b> (search)	Indicates a search key—a field for which one is prompted in a search record dialog box.
<b>List</b>	Indicates a list box item—that is, whether the values for the field appear in the search record list box.
<b>Sys</b> (system)	Indicates that the field is generated and maintained by the system.
<b>Audt</b> (audit)	Specifies whether additions, changes, or deletions to data in this field are written to the standard PeopleTools Audit Table (PSAUDIT).
<b>InTypAhd</b>	Indicates that the field is used as a Type Ahead field and appears in the type ahead window.
<b>EnTypAhd</b>	Specifies whether Type Ahead has been enabled in a search record.
<b>Default</b>	Represents the default value that is used to initialize the field.

## Viewing Editing Options

Edits Display mode shows all editing options (edit as a validation rule) that are available for fields in a record. Edits on a field vary from one record definition to another. To define parameters, access the Record Field Properties dialog box by double-clicking the field.

<b>Req</b> (required)	Indicates whether the field is required. A user must enter a value before the record can be written to the database.
<b>Edit</b>	<p>Specifies that the values for this field are validated against a table.</p> <p>The translate table stores the codes and translate values:</p> <p><i>Y/N</i> (yes or no): Only two values, Y or N, are valid.</p> <p><i>Prompt</i>: Values reside in a designated prompt table.</p>
<b>Prompt Table</b>	Indicates the prompt table for a field. When a user clicks the prompt button or presses the ALT+5 key combination on a page, the values that are stored in this table are retrieved.
<b>Set Control Field</b>	If you plan to use the Table Sharing feature to add an additional high-level key to identify common sets of values and handle exception values, add the name of the key here. The set control field determines which set of values appears, based on how you define table sharing.



## Sizing and Sorting Columns

You can change column lengths in any of the displays by dragging them to the appropriate size with your cursor. For example, you might make the short name column smaller so that the entire long name for each field appears. The default sizing of all columns returns after you close the record and reopen it.

You can also sort the rows in columns by double-clicking the column heading. For example, double-clicking the Num column heading returns the list of record fields to the default sort order.

---

## Saving Record Definitions

PeopleSoft recommends that you save your work every time that you define a new record definition. As soon as you add or change one element in the new definition, save your work and name the record. You cannot save a record definition until you make at least one change to the record definition by changing record properties, adding or deleting at least one field, or changing the record field properties.

You cannot add PeopleCode to a field until you save the record definition.

To save a new record definition, select File, Save or File, Save As. If you haven't named the definition, the system prompts you to enter a record name. The system also prompts you to set the tablespace with the Change Space dialog box.

See [Chapter 7, "Administering Data," Setting the Tablespace, page 187](#).

---

## Naming Record Definitions

Use these guidelines for record definition names:

- The name length can be up to 15 characters, with the exception of the Temporary Table type, which has a maximum length of 13.
- The name must begin with a letter and can contain underscores to make it more readable.
- Avoid special characters, such as # or \$, which can cause problems in some database environments.

See [Chapter 2, "Using PeopleSoft Application Designer," Saving Definitions, page 31](#).

### ***Record Naming Conventions***

To help identify the purpose of different types of record definitions, PeopleSoft recommends that you adopt these naming conventions for record definition names, and use these suffixes:

<b>_TBL</b>	Identifies an edit or prompt table that contains data that is used for validation, as opposed to data that is maintained by the application. Prompt tables store commonly used values. They include, but are not limited to, control tables, which store company-wide values. For example, the location table (LOCATION_TBL) stores values for all operating locations in which your company does business; the country table (COUNTRY_TBL) stores values for all valid countries.
<b>_VW</b>	Identifies a record definition that is physically implemented by defining a SQL view.
<b>_DVW</b>	Identifies a dynamic view.
<b>_WRK</b>	Identifies derived work records.
<b>_SBR</b>	Identifies subrecords.
<b>_QVW</b>	Identifies a query view.
<b>_WL</b>	Identifies the record as a worklist record definition.

In some cases, PeopleSoft also uses these prefixes to identify special types of record definitions:

<b>R_</b>	Identifies work record definitions for Structured Query Report reports. The remainder of the record name consists of the program or report ID.
<b>AUDIT_</b>	Identifies record definitions that store audit information for other record definitions in the database.
<b>WEBLIB_</b>	Identifies record definitions that store internet scripts. Internet scripts are generally located in FieldFormula PeopleCode events. You must grant access in the Security component for a WEBLIB record with an internet script before it can be run in a PeopleCode program.
<b>FUNCLIB_</b>	Identifies record definitions that contain written PeopleCode functions, as opposed to built-in functions. You can include these records in the component and call them as functions. These self-developed functions are generally located in FieldFormula events, and the records are usually derived.
<b>DERIVED_</b>	Identifies shared record definitions (across an application module or group) that have fields for PeopleCode events.

---

## Creating a New Record

This section discusses how to create new record definitions.

To create a new record definition:

1. Select File, New.

2. Select Record.
3. Click OK.

The object workspace appears so that you can build a list of fields in a record definition.

4. Select the Record Type tab to define the type of record definition.

<b>SQL Table</b> (Structured Query Language table)	Select to define a record definition that has a corresponding physical SQL table in the database. Create this table when you run the Build Operation from the Build menu. This is the default setting.
<b>SQL View</b> (Structured Query Language view)	<p>Select to define a record definition that corresponds to a SQL view, which is not a physical SQL table in the database, but rather fields from one or more SQL tables that are reorganized into a different sequence. This provides an alternate view of information that is stored in tables.</p> <p>To create the SQL view, click the Click to open SQL Editor button, enter a SQL Select statement, and then run the Build process.</p> <p>See <a href="#">Chapter 6, "Creating Record Definitions," Creating SQL View and Dynamic View Select Statements, page 145.</a></p>
<b>Dynamic View</b>	<p>Select to define a record definition that can be used like a view in pages and PeopleCode, but is not actually stored as a SQL view in the database. Instead, the system uses the view text as a base for the SQL Select that is performed at runtime. Dynamic views can provide superior performance in some situations, such as search records and in PeopleCode Selects, because they are optimized more efficiently than normal SQL views.</p> <hr/> <p><b>Note.</b> Keys for dynamic views should not be effective date fields.</p> <p>The dynamic view should contain only fields whose names match exactly the fields comprising the SQL query.</p> <hr/>
<b>Derived/Work</b>	Select to define the record definition as a temporary workspace to use during online page processing. A derived or work record is not stored in the database, so you do not build it.
<b>SubRecord</b>	Select to define the record definition as a subrecord—a group of fields that are commonly used in multiple record definitions—that you can add to other record definitions. This way, you can change a group of fields in one place, as opposed to changing each record definition in which the group of fields is used.
<b>Query View</b>	Select to define the record definition as a view that is constructed using the PeopleSoft Query tool. Before you can create the view, PeopleSoft Application Designer prompts you to save the definition.
<b>Temporary Table</b>	Select to define the record definition as a temporary table. Temporary images of the table can be specified on the PeopleTools Options page. Temporary tables are used for running PeopleSoft Application Engine batch processes. Temporary tables can store specific data to update without risking the main application table.

**Non Standard SQL  
Table Name**  
(nonstandard Structured  
Query Language table  
name)

Specify the SQL table name that you are defining, to override the standard convention of prefixing PS\_ to the record name.

### See Also

Chapter 6, "Creating Record Definitions," Manipulating Fields in Record Definitions, page 121

---

## Opening an Existing Record

Open a record definition in the same manner you open other definitions in PeopleSoft Application Designer. In the open dialog box you can narrow the search by selecting a record type or project. You can also open an existing record from the project tree view by double-clicking a record name.

---

## Manipulating Fields in Record Definitions

This section provides an overview of fields in record definitions and discusses how to:

- Insert fields into records.
- Insert subrecords into records.

## Understanding Fields in Record Definitions

You create record definitions by adding field definitions to a new record definition or by cloning and modifying an existing record definition. You can add fields in any order and reorder them at any time. Keys should be located at the top of the record definition, in order of importance.

Each field has basic attributes that are shared across all records that contain the field. These field properties include data type, field name, long name, short name, field length (or integer and decimal positions), formatting, help context number, and translate values. If you change any of these attributes for a field, the change affects *every* occurrence of the field in every record definition. If the change isn't appropriate for every occurrence of this field, consider defining a new field instead.

---

**Note.** Changing a field name or length requires modification of the underlying SQL table, either by running the SQL Build or SQL Alter menu items or by a system administrator action. For example, if you change a field length on one record definition and 30 other record definitions contain the same field name, you have 31 records that must be built or altered. If the records are all within one project, you can run the build process on the project instead.

---

## Inserting Fields Into Records

You can insert a field into a record in one of several ways:

- Using the project workspace tree.

To insert a field into a record definition directly from the project workspace by using drag and drop:

1. Locate a field using the PeopleSoft Application Designer project workspace tree.
2. Drag the field from the project workspace tree to the object workspace.

The field and its attributes are automatically added to the list of fields in the record definition.

- Using the Insert menu (Insert, Field).

Use the Insert, Field menu selection to search for fields with selection criteria, and then add them to the record using the Insert button. You can also double-click a selected field definition and select Insert the Field Definition. Fields are inserted below the selected field in the record definition.

- Dragging fields from existing records.

To drag a field from one record definition to another:

1. Open a record that contains the fields that you need.
2. Select the field.
3. Drag the field to a new record.

## Inserting Subrecords Into Records

Select Insert, SubRecord to search, select, and insert subrecords into a record. A subrecord enables you to add a group of fields that are commonly used in multiple record definitions. A subrecord must be defined before it can be inserted into a record definition.

### ***Viewing a Subrecord***

Any open record in which a subrecord is inserted can have the subrecord expanded into the same record definition window by selecting View, Expand All Subrecords. After selecting this option, the parent record definition expands to show the fields in the subrecord. Subrecord fields are shaded.

When the subrecord is expanded, you cannot insert, cut, delete, paste, reorder, and sort fields. If you reordered the display of the fields, you cannot expand the subrecords.

The expanded subrecord fields are read-only, which means that you cannot see the properties of these fields. To access the properties, first open the subrecord. The shortcut to opening a subrecord is to click the expanded subrecord and select *View Definition*. This opens a subrecord definition from which you can view the properties of the fields.

Collapse the subrecord by selecting View, Collapse All Subrecords.

### ***Nesting Subrecords***

Nested subrecords are fully supported to any level. Expanding a record toggles the record field list to show all of the fields from all levels of nesting. Changes to subrecords are immediately reflected in expanded records.

### ***Editing PeopleCode in Subrecords***

You can edit PeopleCode that is attached to a subrecord field by double-clicking the appropriate field as it appears in the expanded record. Any PeopleCode changes that you make apply to all records that contain that subrecord.

---

## **Setting Record Properties**

After you create a record definition, you'll want to apply attributes so the record functions. Access the dialog box by selecting Record Properties, then either the General or Use tab. This section discusses how to:

- Set general properties.
- Set use properties.
- Create user-defined audit record definitions.

### **Setting General Properties**

Access the Record Properties, General tab.

<b>Description</b>	Enter a descriptive name for the record.
<b>Record Definition</b>	Enter appropriate comments in this area, including details about the record type, use, parent and child relationships, or other information that is important to other application designers.
<b>Owner ID</b>	Displays a list of applications with which this record is used. This list is helpful to identify the applications with which the record is associated during application development.
<b>Last Updated</b>	Displays the date and time of the last modification that was made to the record and the name of the user who made the modification.

### **Setting Use Properties**

Access the Record Properties, Use tab.

These properties control the way the record definition is used in the system.

Specify the field with which the system maps between the original key and the tablesets in the Set Control Field drop-down list box.

See [Chapter 5, "Planning Records, Control Tables, and TableSets," Defining Set Control Fields, page 107.](#)

### ***Record Relationship Settings***

<b>Parent Record</b>	If this is a subordinate or child record, specify the parent record.
<b>Related Language Record</b>	<p>Specify a related language record:</p> <ul style="list-style-type: none"> <li>• The master record definition, which contains the appropriate key and nonkey field definitions.</li> <li>• A clone of the master record definition, to which you add an additional key for language code.</li> </ul> <p>This is the related language record definition. This definition should contain only those nonkey fields for which contents vary by language. Link the two record definitions by specifying the name of the related language record definition in this field on the master record definition.</p> <p>At runtime, the system checks the user's language preference and retrieves the data value from the appropriate related language record definition.</p>
<b>Query Security Record</b>	Specify a query security record if you want secure access to a record using a security view.
<b>Optimization Delete Record</b>	Specify a delete record to use for optimization.
<b>Record Name</b>	Specify the user-defined audit record.

### ***Audit Options***

While you can audit individual fields at the field level, you might find it more efficient to have the system audit the entire row whenever a user adds, changes, or deletes information. With record-level audits, the system focuses on rows of data, instead of specific fields. Consequently, a record-level audit writes a single row of audit data, rather than writing multiple rows for each insert, change, or delete transaction at the field level.

<b>Add</b>	Inserts an audit table row whenever a new row is added to the table underlying this record definition.
<b>Change</b>	Inserts one or two audit table rows whenever a row is changed on the table underlying this record definition.
<b>Selective</b>	Inserts one or two audit table rows whenever a field that is also included in the record definition for the audit table is changed.

<b>Delete</b>	Inserts an audit table row whenever a row is deleted from the table underlying this record definition.
---------------	--

### ***Record-Level Auto-Update Options***

These settings are used exclusively for mobile applications, enabling developers to display derived values on a mobile page. They are not available for subrecords.

---

**Important!** PeopleSoft Mobile Agent is a deprecated product. These features exist for backward compatibility only.

---

<b>System ID Field</b>	Set this field to assign a unique way to identify the record for mobile synchronization purposes. This field must have a user defined index created for it.
<b>Timestamp Field</b>	Set this field to automatically update this field with the date and time when there's a change to the record for mobile synchronization purposes.

### ***Record Information***

The Record Information contains options used during PeopleSoft upgrades.

<b>Tools Table</b>	Identifies SQL tables owned by Enterprise PeopleTools development, which is typically the same ownership as indicated by the PPLTOOLS project.
<b>Managed</b>	Identifies SQL tables owned by Enterprise PeopleTools development that contain information which is used by managed (cached) objects.
<b>Upgrade Job Not Needed</b>	Identifies SQL tables that <i>do not</i> require an upgrade job to be defined. This option applies only to "A to B" upgrades.

---

**Note.** The Append ALL (Dynamic Views) option applies to dynamic views. If it is set, at runtime the system appends the WHERE clause generated at runtime to all the SELECT statements in a dynamic view with a UNION.

Also, if it is set, at runtime the system generates FILL as the table alias in the WHERE clause condition. For example, FILL.<record field name> LIKE '%O%'. You need to specify which table corresponds to the FILL in the dynamic view definition. For example, FROM tableA FILL. This is because the same record field name may exist in two different records.

---

### ***See Also***

PeopleSoft upgrade documentation

## Creating User-Defined Audit Record Definitions

To audit at the record level, you must create a record definition and SQL table in which you store audit information. When you create a new audit record definition, name it with an *AUDIT\_* prefix. Some processes, such as the Employee ID Change and Employee ID Delete processes in the PeopleSoft Human Resources Management product line, change fields, such as EMPLID (employee identification). These processes do not affect record definitions that begin with the *AUDIT\_* prefix, leaving your audit data secure.

The easiest way to create an audit table is to open the record definition that you want to audit, and save it as a new record definition, prefaced with *AUDIT\_*. Audit record definitions can't contain key fields. Therefore, if you clone a record definition to create an audit record definition, you must remove all key attributes.

PeopleSoft recommends that you also use the audit-specific fields that are already defined for the PeopleTools audit table (PSAUDIT), which PeopleSoft uses to track field-level audits. Place these audit fields at the top of the audit record definition, as you would keys. If you are creating several user-defined audit record definitions, you might consider creating and using a subrecord definition with these audit fields instead of adding them individually to each audit record definition.

This table lists audit field names and their purpose:

<i><b>Audit Field Name</b></i>	<i><b>Purpose</b></i>
AUDIT_OPRID	Identifies the user who caused the system to trigger the audits—either by performing an add, change, or delete to an audited field.
AUDIT_STAMP	Identifies the date and time the audit was triggered.
AUDIT_ACTN	<p>Indicates the type of action that the system audited. Possible actions include:</p> <ul style="list-style-type: none"> <li>• A: Row inserted.</li> <li>• D: Row deleted.</li> <li>• C: Row changed (updated), but no key fields changed. The system writes old values to the audit table.</li> <li>• K: Row changed (updated), and at least one key field changed. The system writes old values to the audit table.</li> <li>• N: Row changed (updated), and at least one key field changed. The system writes new values to the audit table.</li> </ul>
AUDIT_RECNAME	Identifies the name of the record definition that was audited.

In most cases you should include AUDIT\_OPRID, AUDIT\_STAMP, AUDIT\_ACTN. The AUDIT\_STAMP must be given the attribute AUTOUPDATE. You might also add AUDIT\_RECNAME if you are creating an audit table to audit more than one record definition.

---

**Note.** Select the Auto-Update check box in the Record Field Properties dialog box; otherwise, the audit record does not receive a date and time stamp.

---

---

## Setting Record Field Properties

This section provides overviews of record field properties and record key considerations, and discusses how to:

- Set record field use properties.
- Add from and through logic to the search page.
- Set record field edit properties.
- Move fields in the same record.
- Move fields to another record definition.
- Delete a field from a record definition.
- Rename record definitions.
- Delete record definitions.
- Print record definitions.
- Create SQL view and dynamic view Select statements.

## Understanding Record Field Properties

When you add a field to a record definition, you must define attributes for how the field is used in that record. These include key settings, default values, table edits, and PeopleCode programs.

The properties of record fields are not shared among records; they are specific to a single record definition and are stored with the record. Therefore, even though you might add the same field to multiple records, each record stores a unique set of record field properties while the primary field definition remains the same.

When you access *record* field properties from a *record* definition, you set the *record field* properties—not field definition properties or record definition properties.

To edit record field properties from a record definition, select Edit, Record Field Properties from the PeopleSoft Application Designer menu.

## Understanding Record Key Considerations

When creating key fields in your record definition there are several guidelines to keep in mind.

## Records With Multiple Keys

You can specify more than one field as a key to a record definition—that is, a record definition can have a compound key.

	Num	Field Name	Type	Key	Ordr	Dir	CurC	Srch	List	Sys	Audt	Default
	1	EMPLID	Char	Key	1	Asc		Yes	Yes	No		
	2	ABSENCE_TYPE	Char	Key	2	Asc		No	No	No		'CNF'
	3	BEGIN_DT	Date	Key	3	Desc		No	No	No		%date
	4	RETURN_DT	Date					No	No	No		
	5	DURATION_DAYS	Nbr					No	No	No		
	6	DURATION_HOURS	Nbr					No	No	No		
	7	REASON	Char					No	No	No		
	8	PAID_UNPAID	Char					No	No	No		'U'
	9	EMPLOYER_APPROVED	Char					No	No	No		'N'
	10	COMMENTS	Long					No	No	No		

Defining compound key combinations

For example, the keys to the ABSENCE\_HIST record definition are EMPLID, ABSENCE\_TYPE, and BEGIN\_DT. This means that an absence history can be created only once.

## Keys on Parent and Child Tables

In some cases, you have a field in a table for which you want to allow multiple occurrences, in which case you create a subordinate or child table. For example, for employee reviews, an employee can be reviewed for performance in multiple categories—organization skills, interpersonal skills, and so on. These categories and ratings are stored in a separate child table, EE\_REVIEW\_RT, which is directly related to REVIEW\_DT, the parent table that stores information about employee reviews.

The keys that you establish in a parent record definition determine which keys are required in child record definitions. The child must have the same keys as the parent, plus one or more keys that uniquely identify each row:

<i>Item</i>	<i>Parent Table</i>	<i>Child Table</i>
Record Definition	REVIEW_DT	EE_REVIEW_RT
Key Fields	EMPLID REVIEW_DT	EMPLID REVIEW_DT CATEGORY

Most record definitions have either one primary key, multiple keys that comprise independent, or parent and child key combinations. There are, however, exceptions, such as record definitions without keys and duplicate order keys.



---

**Note.** Records with duplicate order keys should not have related language tables associated to them. If the record requires a related language record you must remove the order key(s). Do not create a related language record with a duplicate order key.

---

### **Ordering Keys**

The position of keys relative to one another is critical in a record definition; always list them in order of importance. The primary key must be the first field in the record, followed by the next most important key, and so on.

Key fields are followed by duplicate order keys, in order of importance, then by fields that are not keys. You can scatter alternate key fields anywhere among regular fields, in any order. They do not need to be grouped.

Key, Duplicate Order Key, and Alternate Search Key are mutually exclusive.

---

**Note.** The display of key fields in record definition views does not always match the order in the record. Always check the Num column to see the actual order.

Each alternate search key that you establish creates a database index when you SQL Create the table. While database indexes are important, they consume disk space and, when the system must support the alternate key, processing time. Therefore, don't add alternate search keys unless you really need them.

For both records and subrecords, you can only add 10 alternate search keys.

---

## **Setting Record Field Use Properties**

As you add fields to a record definition, you must decide which fields uniquely identify each row; these become the record keys. The nature of the data that you are storing should naturally determine the "keys" to the information in the database.

The record field properties options that are available for field types might vary slightly—for example, some of the properties that you assign might be relevant only to a character field in a record and not to a date or time field. However, all of the options are described in this section.

The image shows the 'Record Field Properties' dialog box with the 'Use' tab selected. The 'Field Name' is 'DESCR'. The 'Keys' section has 'List Box Item' checked. The 'Record Field label ID' is 'DESCR "Descr, Description"'. The 'Default Value' section has 'Constant' checked. The 'Default Page Control' is 'System Default'. The 'Type Ahead Configuration' has 'Enable Type Ahead when used in Search Record' unchecked. The 'Audit' section has 'Field Add', 'Field Change', and 'Field Delete' unchecked. The 'System Maintained' and 'Auto-Update' options are also unchecked.

**Record Field Properties**

Use | Edits

Field Name: DESCR

**Keys**

- ☐ Key
- ☐ Duplicate Order Key
- ☐ Alternate Search Key
- ☐ Descending Key
- ☐ Search Key
- ☐ Search Edit
- ☒ List Box Item
- ☐ From Search Field
- ☐ Through Search Field
- ☐ Default Search Field
- ☐ Disable Advanced Search Options
- ☐ Allow Search Events for Prompt Dialogs

**Record Field label ID**

DESCR "Descr, Description"

**Default Value**

Constant:

or

Record Name:

Field Name:

**Default Page Control**

System Default

**Type Ahead Configuration**

☐ Enable Type Ahead when used in Search Record

**Audit**

- ☐ Field Add
- ☐ Field Change
- ☐ Field Delete

☐ System Maintained

☐ Auto-Update

OK Cancel

Record Field Properties dialog box: Use tab

## Keys

### Key

Select to identify the field as the search criteria that uniquely identifies each row. You cannot have duplicate values for primary keys. For example, EMPLID is the only key to the PERSONAL\_DATA record definition. Therefore, EMPLID must be a unique value for each employee and there can be only one PERSONAL\_DATA row per employee.

### Duplicate Order Key

Select to indicate that duplicate values can occur. The order in which you place duplicate order keys in a field list determines the order in which duplicate keys are retrieved.

<b>Alternate Search Key</b>	Select to identify the field as a key that provides an alternate path into the table data. Duplicate values are allowed in an alternate search key field. If you define a field as an alternate search key in a search record, when you bring up a page, the system prompts you to enter a key or alternate search key values.
<b>Descending Key</b>	<hr/> <p><b>Note.</b> This option no longer produces an index with descending order.</p> <hr/> <p>Select to identify the field as descending if you want rows of data to be retrieved in reverse alphanumeric order (for example, 3, 2, and 1). If you clear this check box, the key is ascending, meaning that rows are stored and retrieved in alphanumeric order (for example, 1, 2, and 3). This option applies only to a field that is specified as a key, duplicate order key, or alternate search key. PeopleSoft uses descending order primarily for effective date fields; most character keys are ascending.</p>
<b>Search Key</b>	Select to make the field available on the basic and advanced search or lookup pages. A search key is valid only for keys and should be used only in search and prompt records. If you select this check box, the system automatically selects List Box Item.
<b>Search Edit</b>	Enabled only if Search Key is selected. Selecting this option enforces the required property and table edits on the search page. It also enforces these edits under circumstances where the search page would normally be bypassed. With this option, the user no longer has the ability to perform partial searches on this field.
<b>List Box Item</b>	Select if you want the field to appear in the list box preceding a page. If a field has values in the translate table and you designate it as a list box item, the list box automatically shows the translated value instead of the code.
<b>From Search Field and Through Search Field</b>	<p>Select for fields that are used as search records. If you select From Search Field, the displayed list contains rows in which the field value is greater than or equal to a value that the user enters. If you select Through Search Field, the displayed list contains rows in which the field value is less than or equal to a value that the user enters.</p> <p>If you do not want these fields to be in a search list box, clear the List Box Item check box, even if the field is an alternate search key.</p>
<b>Default Search Field</b>	Select to control which field name appears in the Search By drop-down list box on the basic search page.
<b>Disable Advanced Search Options</b>	Select to prevent runtime search pages from displaying advanced search features for this field.
<b>Allow Search Events for Prompt Dialogs</b>	<p>Select to enable the SearchSave and SearchInit PeopleCode events on a prompt dialog for this field. The default value is <i>off</i>.</p> <hr/> <p><b>Note.</b> The use of these events should be limited in this context. The amount and complexity of the PeopleCode affects the speed of the data retrieval.</p> <hr/>

## **Audit**

The audit options apply only to data that is manipulated on a normal PeopleSoft application page through component processing. These options do *not* apply to data that is added using SQLExec in PeopleCode or by some other means. PSAUDIT logs the user ID, the date and time the field was modified, and the old and new values.

To audit certain fields, regardless of the record definitions in which they're included, you want to control when the fields are audited. To audit several fields in a record definition, you might want to consider specifying audits at the record definition level.

<b>Field Add</b>	Audits this field whenever a new row of data is added.
<b>Field Change</b>	Audits this field whenever the contents are changed.
<b>Field Delete</b>	Audits this field whenever a row of data is deleted.
<b>System Maintained</b>	Select to indicate that the field value is system-generated. This option is for documentation purposes only and doesn't affect processing.
<b>Auto-Update</b>	Select to have the field updated with the server's current date and time whenever a user creates or updates a row. Any user entries—even if permitted on a page—are overwritten by the server time.
<b>Currency Control Field</b>	Enables the user to specify where to find the currency code that is used to display the currency symbol, decimal digits, and scale of a number field. To use this option, the multicurrency option must be set. You might also want to enable the current display when you define the field on the page.

## **Record Field Label ID**

Specify which label, long name or short name, to use for the record. The default value is *\*\*\*Use Default Label\*\*\**. This enables you to change labels of record fields dynamically anytime that the default label on the field definition is changed.

For example, if a field definition has three labels:

- Label1 (Long Name1, Short Name1), marked as default.
- Label2 (Long Name2, Short Name2).
- Label3 (Long Name3, Short Name3).

When the label ID in a record field is set to *Use Default Label*, initially the long name and short name are Long Name1 and Short Name1. If the default label is changed to Label3 in the field definition, then the long name and short name automatically become Long Name3 and Short Name3.

## Default Value

Select the most commonly used value as the default. You can always enter a different value if the default is inappropriate. The more defaults that you provide, the more data entry time you'll save your users.

For a field, you can enter a default value as a system variable or as a combination of record and field names.

<b>Constant</b>	<p>Specify a default value.</p> <p>The value is case-sensitive. If the record has translate values for this field, the values appear in a drop-down list.</p> <p>You can specify a system variable only when you want to display the value of the current date or time. The accepted system variables to be used as constants are:</p> <ul style="list-style-type: none"> <li>• %DATE for DATE, TIME, or DATETIME fields.</li> <li>• %TIME for TIME fields.</li> <li>• %DATETIME for DATETIME fields.</li> <li>• %CLIENTDATE for DATE, TIME, or DATETIME fields. Note that in this case %CLIENTDATE is only relevant for a DATE field. Otherwise it behaves the same as %DATE.</li> </ul>
-----------------	---

<b>Record Name and Field Name</b>	Enter the record and field names of the default value.
-----------------------------------	--

<b>Default Page Control</b>	Specify the default appearance of a field as it appears on the page that corresponds with the record field that you are creating.
-----------------------------	---

## Type Ahead Configuration

<b>Enable Type Ahead when used in Search Record</b>	<p>Select to make this field appear in the type ahead list when the containing record is used as a search record.</p> <p>This field is enabled for Character, Number, and Signed Number fields if the following condition are met:</p> <ul style="list-style-type: none"> <li>• The Default Page Control field value is <i>Edit Box</i> or <i>System Default</i>.</li> <li>• The field is a search key or an alternate search key.</li> </ul>
---	---

## Time Zone and Related Date Fields

These fields are enabled only if the current field is a time or date and time field. They determine whether the field is displayed or entered in a specified time zone.

**Specified Time Zone**

Times are always stored in a database base time zone, but when you place a time field on a page, you can display the time in the base time zone or another time zone.

If you clear this check box, the time appears in the database base time zone. If you have users in multiple time zones, you can reduce confusion by showing the time zone along with the time.

If you select this check box, the system converts the time according to the time zone specified in Time Zone Control Field. This control field must be a field in the current record. Set an appropriate default value for the time zone control field.

**Date Control Field**

Select which related date field in the current record stores the calendar date to which this field should be adjusted.

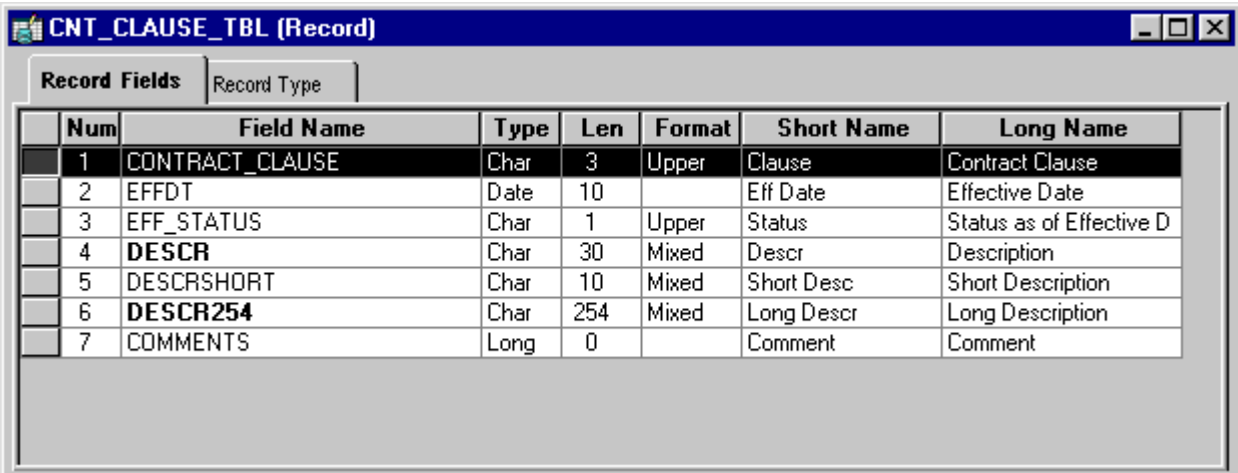
Typically, you want to make a date field a descending key so that the row with the latest and most current time appears first. If you want the default value to be the current system date, enter the value %date (or %time) as the constant in the Default Values group box.

In general, each Time record field that is related to the Time Zone field should also be related to a Date field on that same record. Otherwise, the time data that is displayed or saved may not properly reflect the appropriate state of daylight savings in that time zone. Furthermore, each Time record field should be associated with a *different* Date record field.

For applications where two Time record fields need to share a common Date record field (as in, records with a start time and an end time *always* occurring on the same date), you can relate each Time record field to the same Date record field. However, keep in mind that the Date record field will only consider itself associated with *one* Time record field—the related Time record field that has the earliest position in the record field ordering for that record.

## Adding From and Through Logic to the Search Page

Refer to the following example when applying from and through logic to a search page.



CNT_CLAUSE_TBL (Record)							
Record Fields				Record Type			
	Num	Field Name	Type	Len	Format	Short Name	Long Name
	1	CONTRACT_CLAUSE	Char	3	Upper	Clause	Contract Clause
	2	EFFDT	Date	10		Eff Date	Effective Date
	3	EFF_STATUS	Char	1	Upper	Status	Status as of Effective D
	4	DESCR	Char	30	Mixed	Descr	Description
	5	DESCRSHORT	Char	10	Mixed	Short Desc	Short Description
	6	DESCR254	Char	254	Mixed	Long Descr	Long Description
	7	COMMENTS	Long	0		Comment	Comment

CNT\_CLAUSE\_TBL record

To apply the from and through logic to a search page:

1. Determine which component you want to change.

You must modify the search record for a particular page, so you must first identify the appropriate component.

2. Determine on which field to use the from and through search logic.

Open the component in PeopleSoft Application Designer and examine the search record that is associated with the component. In the CNT\_CLAUSE\_TBL record, the search record is CNT\_CLAUSE\_TBL, and the field to which you want to apply the from and through search logic is CONTRACT\_CLAUSE.

3. Create a new view that contains the same fields as the original search record.

To use the from and through search logic, the search record *must be a view*. If the search record is already based on a view, you can modify the existing view instead of creating a new view.

4. Select the From Search Field property for the field.

In the new view, select the from search field and apply the following record field properties: Key, Search Key, and From Search Field (List Box Item is optional).

5. Create a new field with exactly the same attributes as the from field.

For example, if CONTRACT\_CLAUSE represents the *field* from which you want to search, create a field called CONTRACT\_CLAUSE\_TO representing the field *through* which you want to search.

6. Insert the new field directly below the original field.

7. Select the Through Search Field property for the field.

Select the new through search field and apply the following record field properties: Key, Search Key, and Through Search Field only.

8. Update the SQL view text.

Update the SQL view Select statement in the new view to reflect the new column. That is, the view text should select the same field twice in a row (as shown in the example that follows this procedure). This is because the same field is used for the from logic and the through logic.

9. Save and build the new view.

10. Update the component properties.

You must update the search record so that the component uses the new view.

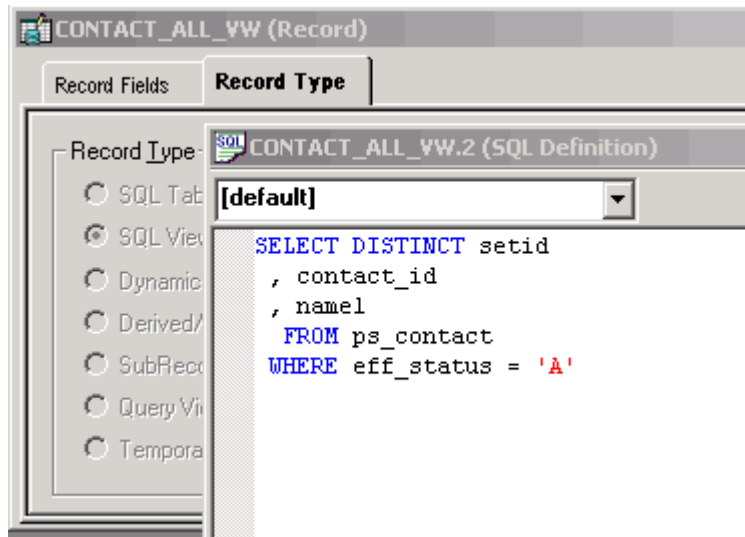
11. Test the new search page.

The search page should now include the new through field so that the user can specify a range of values for the field. Because you want to search on both fields, you must use an advanced search.

The results in the list box should reflect a range of values between the from and through fields that are specified. The results are inclusive—they include the minimum and maximum values that the user specified.

### ***Example of Updating the SQL View Select Statement***

Once you define a search field as a Through Search Field, you'll need to update the SQL view text.



Updating the SQL view Select statement

## **Setting Record Field Edit Properties**

The Edits tab enables you to specify whether the system is to perform special edits or validations on a record field. You also name the record definition for the prompt table that stores values.

The image shows the 'Record Field Properties' dialog box with the 'Edits' tab selected. The 'Field Name' is 'DESCR'. The 'Required' checkbox is unchecked. Under 'Edit Type', 'Table Edit' is selected. The 'Table Edit' section contains three dropdown menus: 'Type' (set to 'Prompt Table Edit'), 'Prompt Table', and 'Set Control Field'. The 'Display in Type Ahead Window' checkbox is also unchecked. 'OK' and 'Cancel' buttons are at the bottom right.

Record Field Properties dialog box - Edits tab

**Required**

Select if you don't want users to skip a field or leave it blank. Users are unable to save their work until they complete all of the required fields on a page.

---

**Note.** The system only checks required fields when the page field is changed or the record buffer is changed.

---

**Edit Type****No Edit**

Select if you don't want to edit the contents of this field against a table. This is the default, which makes the options in the Table Edit group box unavailable.

<b>Table Edit</b>	Select to edit the contents of the field against the values that are maintained in the specified table. When you select Table Edit, the Type field becomes available in the Table Edit group box.
<b>Display in type ahead Window</b>	<p>Select to make this field appear in the type ahead auto-suggest list.</p> <p>This property is enabled for date, datetime, time, character, number, and signed number fields if the following conditions are met:</p> <ul style="list-style-type: none"> <li>• The field is a List Box Item.</li> <li>• The field is not a search item.</li> </ul>

### Table Edit

<b>Type</b>	<p>Select from the drop-down list box:</p> <p><i>Prompt Table with No Edit:</i> Provides users with a list of suggested values, but does not edit the contents of the field against the prompt table. Users can enter any value. Selecting this option makes the Prompt Table field available.</p> <p><i>Prompt Table Edit:</i> Edits the contents of the field against the values that are maintained in the specified prompt table. Selecting this option activates the Prompt Table field. When you enter a prompt table name and exit the field by pressing tab, Set Control Field becomes activated.</p> <p><i>Translate Table Edit:</i> Edits the contents of the field against the translate table. The translate table stores values for fields that must be validated but don't need individual tables of their own. If you select this option, Prompt Table and Set Control Field become unavailable.</p> <p><i>Yes/No Table Edit:</i> Makes the values for this field Y (yes) and N (no) only. If you select this option, Prompt Table and Set Control Field become unavailable.</p> <p>This is a 1/0 table edit where 1=True and 0=False.</p>
-------------	---

---

**Note.** The list of available table edit types is dependent on the field type.

If you add a field (other than with the paste action) to a record and the field name is already defined in the database with previously defined translate values, the system automatically selects Table Edit and selects *Translate Table Edit* as the table type. For both *Translate Table Edit* and *Yes/No Table Edit*, PeopleSoft recommends that you follow its design standards, select Required, and specify a default value for the field in the Record Field Properties dialog box.

---

<b>Prompt Table</b>	<p>Enter the name of the record definition that you want to use as the prompt table for this field.</p> <p>If you want the prompt table to vary depending on the context of the field, indicate a field in the derived or work record (DERIVED) that contains the name of the prompt table at runtime. Then, in this field enter <i>%FieldName</i>.</p> <p>The % is required, and indicates that you're referencing a derived or work record definition named DERIVED. <i>FieldName</i> is the name of the field in that DERIVED record definition.</p>
---------------------	---

**Set Control Field**

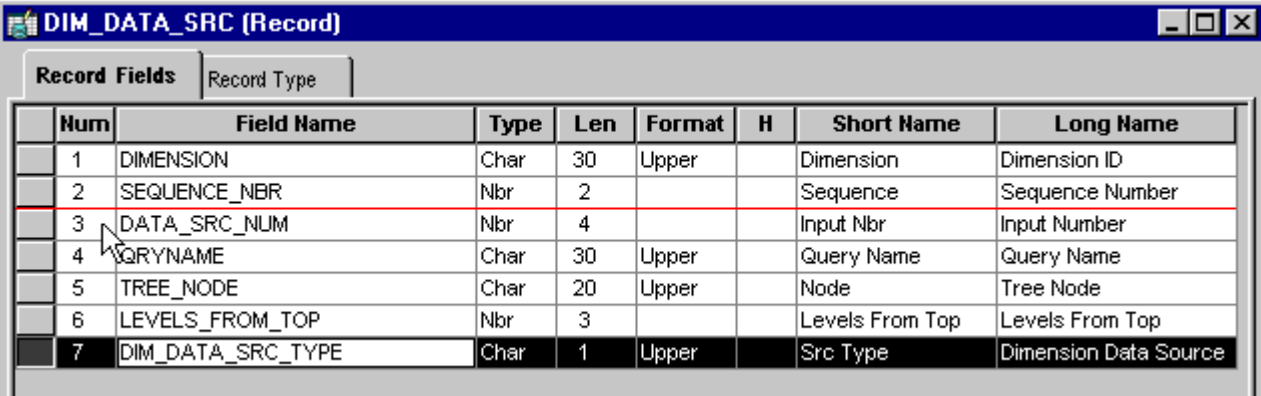
Select a set control field that overrides the set control field of the record definition specified in the prompt table. If you don't specify a name in this field, the default is the set control field of the record definition specified in the prompt table.

**Reasonable Date**

Select if you want the system to test the field value to determine whether it is within 30 days of the current date. If the date is out of range, a warning message appears when the user exits the field.

## Moving Fields in the Same Record

You can move fields in a record by dragging a field to another place in the open record definition. In addition, you can change the visible order of fields by clicking any of the grid headings.



Num	Field Name	Type	Len	Format	H	Short Name	Long Name
1	DIMENSION	Char	30	Upper		Dimension	Dimension ID
2	SEQUENCE_NBR	Nbr	2			Sequence	Sequence Number
3	DATA_SRC_NUM	Nbr	4			Input Nbr	Input Number
4	QRYNAME	Char	30	Upper		Query Name	Query Name
5	TREE_NODE	Char	20	Upper		Node	Tree Node
6	LEVELS_FROM_TOP	Nbr	3			Levels From Top	Levels From Top
7	DIM_DATA_SRC_TYPE	Char	1	Upper		Src Type	Dimension Data Source

Moving field to new position

To move a field in the same record definition:

1. Select the field that you want to move.

Use the Ctrl and Shift keys to select multiple fields at once.

2. Drag the selected field number to the new position in the record definition window.

A red line between rows indicates the new placement options as you drag the field. The Num column automatically renumbers the fields in the new order.

---

**Note.** The fields are reordered only in the visible display of the record definition, not in the actual table.

---

## Moving Fields to Another Record Definition

To move fields from one record definition to another, you can cut and paste the fields. You can also drag a field from the project workspace into a record definition in the object workspace or between open record definitions.

To move a field from one record definition to another:

1. Open the two records between which you want to move the field.
2. Select the field that you want to move.

Move multiple fields by using the Ctrl or Shift keys.

3. Select Edit, Cut.

If this is the correct field to cut from the record, click Yes.

4. Select the new position for the field in the destination record.
5. Select Edit, Paste to paste the field into the new record definition.

---

**Note.** PeopleCode that is associated with fields is not carried over with cut and paste operations. The same is true for delete and undo with field deletions in records. RecordField attributes, such as key, search key, and so on, *are* retained.

---

## Deleting a Field From a Record Definition

To delete a field from a record definition:

1. Select the field that you want to remove and press the del (delete) key.

This deletes the field completely and doesn't copy it to the clipboard, unlike a cut operation, which *does* copy to the clipboard.

2. When the system prompts you to confirm the deletion, click Yes.

If you've already SQL Created the underlying table for the record definition from which you are deleting the field, re-create the table, or use the SQL Alter function to alter the table.

---

**Important!** If you delete a field from a record definition, you must also delete it from any pages on which it appears. When you delete a field, the system does not automatically delete references to the field in PeopleCode, so you must do this manually. To determine where the field is referenced in PeopleCode, use the Find Object References feature.

You can also reference the following two reports: "Fields and Records" (XRFFLRC) shows which records contain the field; "Fields Referenced by PeopleCode Programs" (XRFFLPC) shows PeopleCode that refers to the field in the record.

---

Modify or remove PeopleCode when you find references to the deleted field.

## Renaming Record Definitions

To rename a record definition:

1. Select File, Rename.

The Rename Definition dialog box appears.

2. Select *Record* from Definition Type.

3. Click Rename.
4. Select the record and click Rename.  
A rectangular box appears around the name.
5. Enter the new name and press enter.

### ***Results of Renaming Record Definitions***

When you rename a record definition, the system automatically renames all references to it, including data on tables with columns named RECNAME where the data matches the record being renamed.

The only references that are *not* renamed are the text portion of SQL functions, such as SQLExec and Scroll Select, and the record names in the view text. To find the text portion of SQL functions in PeopleCode or record names in view text, select Edit, Find In to search for the matching text.

If you have already SQL Created the underlying tables for the record definition that you renamed, re-create that table. To preserve data in the tables, use the SQL Alter function to rename the database tables.

## **Deleting Record Definitions**

To delete a record definition:

1. Select File, Delete.
2. Select the record definition to delete.
3. Click Delete.

---

**Warning!** When you delete a record definition, the system automatically deletes any PeopleCode that is associated with the record.

---

4. Click Yes if you really want to delete the record definition.

---

**Note.** Notify your database administrator about which record definition you deleted, so that the administrator can drop the underlying SQL table and its contents from the database.

---

## **Printing Record Definitions**

You can print your record definitions as references identifying all of the fields and their various attributes—any special use, edits, or PeopleCode that you've applied.

To print a record definition, it must be open in the object workspace.

To print a record definition:

1. Select File, Print Setup to change any of the print record defaults.

The system retains your changes until you reset them again. The Print Setup dialog box appears.

## 2. Select printing options.

<b>Definition</b>	Print a picture of what the definition looks like.
<b>PeopleCode</b>	Select if you want the report to include a listing of any PeopleCode programs that are attached to each field, identifying the program type and listing all of the PeopleCode statements. Selected by default.
<b>Graphics</b>	Not used for record definition printing.
<b>Margins(mm)</b>	Set the distance (in millimeters) from the edge of the page to the left, right, top, and bottom edges of the page image or report. The defaults are: 20 (top), 5 (bottom), 5 (left), and 0 (right).
<b>Header</b>	Print a header at the top of the report indicating the date and time at which you printed the report and the database name, record name, version number, and page number. Selected by default.
<b>Footer</b>	Print a footer at the bottom of the report indicating the date and time at which you printed the report and the database name, record name, version number, and page number. Cleared by default.
<b>Border</b>	Print a border or box around a record definition report. To print reports faster, clear this check box so that the printer can print the report in character mode, rather than in graphics mode. Cleared by default.
<b>Border Space(mm)</b>	Insert a set amount of space between a graphical boarder around the record definition report and the margins of the report. The default value is 3.

## 3. Click OK when you are done, to close the Print Setup dialog box and save your settings.

## 4. Select File, Print.

***Record Definition Report***

The following table lists the columns in the record definition report and their contents:

<b><i>Column</i></b>	<b><i>Contents</i></b>
Field Name	The name of the field.
Type	The field type.
Length	The length of the field (not specified for long character fields).
Format	The field format.

<b>Column</b>	<b>Contents</b>
Long Name	The 30-character name of the field.
Short Name	The 15-character name of the field.
Key	Identifies key attributes (characters defined by position).
Req (required)	<i>Yes</i> indicates that the field is required.
TblEdt (table edit)	<i>Prompt</i> : Field values are edited against a specified prompt table. <i>Y/N</i> : The field uses the Yes/No Table. <i>Xlat</i> : The field has values in the translate table. Values (if printed) are listed below the field.
AU (auto-update)	<i>Yes</i> indicates that the auto-update option is enabled.
Dt (date)	<i>Yes</i> indicates that the reasonable date option is enabled.
PC (PeopleCode)	<i>Yes</i> indicates that the field contains PeopleCode. PeopleCode text (if printed) appears below the field.
Aud (audit)	<i>Yes</i> : The field audit flag is enabled. <i>A</i> : Audit add. <i>C</i> : Audit change. <i>D</i> : Audit delete.
Prompt Table	The name of the prompt table.
Default Value	Any default value or constant.

You might also see these additional reference lines below each field or at the end of the report:

<b>Reference</b>	<b>Description</b>
SQL View	Shows the SQL view Select statement for view-type record definitions. This appears at the top of the report.
System Maintained	Indicates that the System Maintained check box is selected. Because this option is for documentation purposes only, it doesn't warrant its own column heading.

<b>Reference</b>	<b>Description</b>
DbField Help Context: <i>nnn</i>	Indicates that the field has been assigned a field help context number to link it to a help file that describes how the field is used wherever it appears in the database.
RecField Help Context: <i>nnn</i>	Indicates that the field has been assigned to a record field help context number to link it to a help file that describes how the field is used only as it appears in this record definition.
Audit Record	Identifies a user-defined audit record. Lists the audit record name and the type of audit.
Set Control Field	Identifies a set control field that is designated for the record definition.
Related Language Record	Identifies a related language record that is designated for the record definition.
Query Security Record	Identifies views that are used to restrict query access to data that is stored in the table.
Parent Record Name	Identifies the hierarchical relationship of the record for query reporting.

Printing records with subrecords takes into account whether the record view is expanded or collapsed. If the record view is expanded, the subrecord fields are indented to the appropriate level of nesting.

## Creating SQL View and Dynamic View Select Statements

If you are creating a SQL view or dynamic view record definition, you must enter a SQL view Select statement, to indicate which field values you want to join from which tables. The only difference between the standard view and dynamic view is that the dynamic view is not defined as a view to the database—it is stored on the client and run as a Select at runtime. Dynamic views avoid some constraints on views on some platforms.

**Non-Standard SQL Table Name** (nonstandard Structured Query Language table name) Override the standard convention of prefixing *PS\_* to the record name.

**Build Sequence No** (build sequence number) Set the order in which the dynamic view is to be created. The default value is *1* when the record or view is initially created. Views that must be created first can be set to *0*, while views that you want created last can be set to *1* or greater. The build sequence number is stored with the other details of the record or view in the database.

**Click to open SQL Editor** The view text is saved when the record is saved, by selecting File, Save. The record must be saved first, before opening the SQL Editor.

As a general guideline, the SQL used for dynamic views should be as simple as possible because the runtime system appends WHERE clauses to the SQL definition to add key field lookup criteria and doesn't parse the dynamic view SQL definition to find correlation names. For example, the following format is not recommended:

```
SELECT...FROM(SELECT * FROM...WHERE...)
```

Rather than having a WHERE clause in the sub query, one approach would be to create a static view that the dynamic view references. For example, that static view might be:

```
...
ABC_PRD_INS_VW as

SELECT B.SETID AS SETID
, A.BO_ID_CUST AS BO_ID_CUST
, A.INST_PROD_ID AS INST_PROD_ID
, B.PRODUCT_ID AS PRODUCT_ID
, B.DESCR AS DESCR
, A.SERIAL_ID AS SERIAL_ID
, A.INSTALLED_DATE AS INSTALLED_DATE
FROM PS_RF_INST_PROD A
, PS_PROD_ITEM B
, PS_RF_INST_PROD_ST C
, PS_PROD_PGRP_LNK D
WHERE A.SETID = B.SETID
AND A.PRODUCT_ID = B.PRODUCT_ID
AND A.SETID = C.SETID
AND A.INST_PROD_ID = C.INST_PROD_ID
AND C.INST_PROD_STATUS <> 'UNI'
AND A.SETID = D.SETID
AND A.PRODUCT_ID = D.PRODUCT_ID
AND D.PRODUCT_GROUP = '00'
```

Which is referenced by the dynamic view:

```
SELECT SETID
,BO_ID_CUST
,INST_PROD_ID
,PRODUCT_ID
,DESCR
,SERIAL_ID
,INSTALLED_DATE
FROM PS_ABC_PRD_INS_VW
WHERE SETID = :1
AND BO_ID_CUST = :2
AND INST_PROD_ID = :3
```

---

**Note.** The order of the columns in the Select statement must be identical to the field order in the corresponding record definition. Also, only certain types of meta-SQL statements can be used in view text.

---

**Note.** Related language records should not use dynamic views, but should be limited to physical views and records.

---

## See Also

[Chapter 7, "Administering Data," Using Cross-Platform Meta-SQL for Dynamic Views, page 185](#)

## Chapter 7

# Administering Data

This chapter provides an overview of the PeopleSoft Application Designer data administration and the Build process and discusses how to:

- Use the Build menu.
- Create tables.
- Create indexes.
- Alter tables.
- Create triggers.
- Create views.
- Administer DDL.

---

## Understanding Data Administration and the Build Process

Using PeopleSoft Application Designer, you create several kinds of definitions that represent database components. For instance, field definitions represent table columns, and record definitions represent tables and views. From these field and record definitions, you can create tables, views, and indexes. The important concept to remember is that the definitions are just the blueprints for the actual, physical database components with which they are associated. In the context of the entire application development cycle, use the Build process to create Structured Query Language (SQL) tables, views, triggers, and indexes after you define new fields and create the record definitions. After you build the SQL tables, you begin creating page definitions.

After you create and define your field and record definitions, you must create or build the underlying SQL tables that house the application data that your users enter online in your production environment. This build process is the centerpiece of the data administration features in PeopleSoft Application Designer. The Build process uses Data Definition Language (DDL) to construct a physical database component that is based on the associated record and field definitions that you created. With the build feature, you can create the following:

- Tables
- Indexes
- Views
- Triggers

You can also use the build feature to *alter* existing tables if you change the record definition after the table already exists. Altering a table is useful, because it enables you to make changes without losing the application data that is already housed in the table. In general, the results of the build operation are written to a script file that a database administrator can run later. On some database platforms, you can run the SQL online, if you prefer, so that your changes are immediately reflected in the physical database. However, if you run the SQL immediately, you cannot review it to make sure that the table that was built truly meets all of your requirements.

Before you begin using the data administration tools and running a build process, make sure to:

- Review what DDL means for PeopleSoft applications.
- Grant build authority.

### **Reviewing DDL**

DDL is the part of SQL that pertains to the creation of tables, indexes, views, triggers, and tablespaces. DDL is also the part of SQL that differs most between the various relational database platforms. Each database vendor provides different syntax and configuration options for creating and organizing tables and for optimizing performance. Because PeopleTools supports multiple database platforms, PeopleSoft has designed a flexible way of specifying DDL that enables you to take advantage of each vendor's features. The basic components of the PeopleSoft DDL support include the following:

- DDL model definition: A complete set of the supported DDL statements for each database platform.  
Statements include Create Table, Create Tablespace, and Create Index. Each DDL model statement has substitution parameters that can be specified at the database level or overridden for individual records.
- Record DDL: Specify the DDL model substitution parameters for an individual record.
- Index DDL: Specify the DDL model substitution parameters for an index.
- Sizing sets: A way to maintain multiple versions of your DDL model statements for a database platform.

For example, you can use one sizing set during a development phase, when tables have only test data, and another during production, when tables have much more data.

### **Granting Build Authority**

PeopleSoft Security enables you to specify which users can build scripts, run scripts (Execute SQL now), maintain DDL, and so on. Access Security from the PeopleTools link in the menu.

### **See Also**

*Enterprise PeopleTools 8.50 PeopleBook: Security Administration*, "Setting Up Permission Lists," Setting PeopleTools Permissions

---

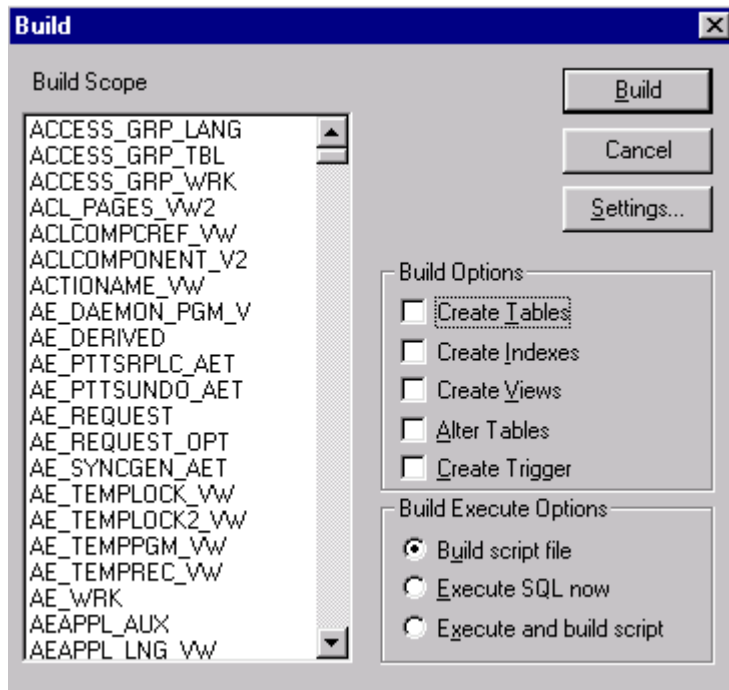
## **Running the Build Process**

This section provides an overview of the Build menu and discusses how to:

- Select the build scope.
- Select build options and run the Build process.

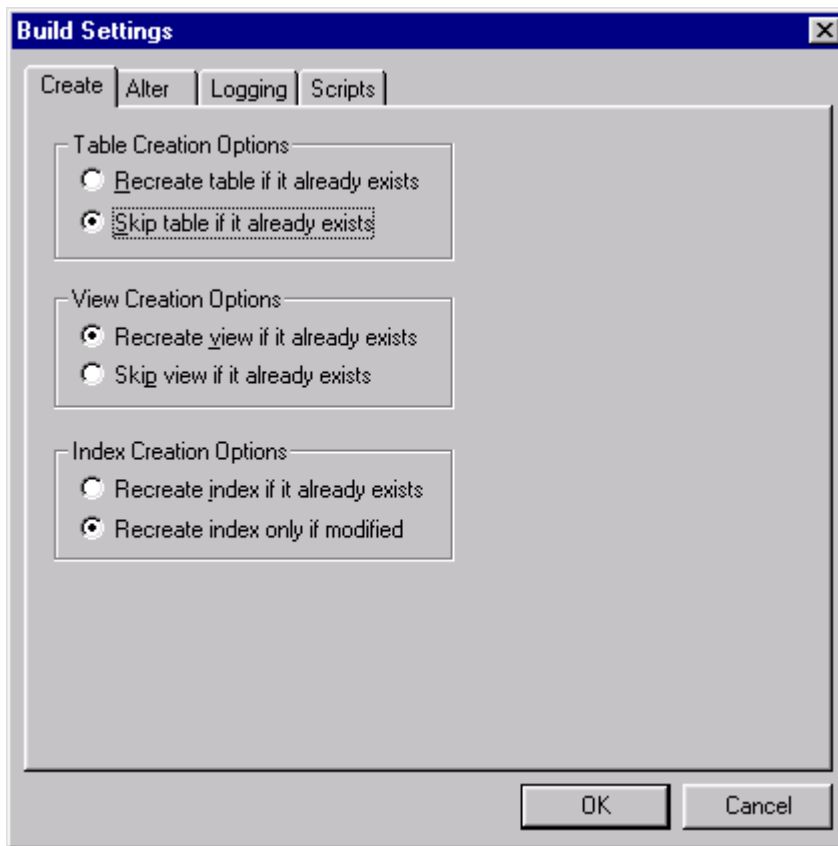
## Understanding the Build Menu

Access all of the commands and options that are associated with building tables, views, indexes, triggers, and altering tables from the Build menu. When you choose to build an item from the Build menu, the tasks for creating or running SQL scripts that define the underlying database components are included in the Build dialog box.



Build dialog box

The settings for the Build process on the Build Settings dialog box enable you to configure various aspects of the process. Depending on which build option you are running, your settings vary. The Build Settings that you select are stored on your workstation so that they can be reused during future PeopleSoft Application Designer sessions. Access the Build settings by selecting Settings from the Build menu in PeopleSoft Application Designer or click the Settings button in the Build dialog box.



Build Settings dialog box

When you select the Create Tables, Create Views, or Create Indexes options in the Build dialog box, specify the appropriate options on the Create tab of the Build Settings dialog box such as if you need to specify whether an existing table, view, or index is skipped or dropped and re-created.

If you modify your record definitions, the tables in your production database should reflect those changes to maintain your data integrity. When you alter a table, select the appropriate options, such as the options in the Drop Column Options area on the Alter tab of the Build Settings dialog box.

You can monitor the Build process by reviewing the log files that it automatically generates. Keep in mind that the log file is *entirely* separate from the script file. The amount of information that the log file contains is up to you. You can set up logging so that all statuses (both good and bad) appear in the log, or you can specify that only errors or warnings appear in the log. Control the settings for the Build log file (such as the logging level and the location for the output log file) on the Logging tab of the Build Settings dialog box.

If you build a script file, then the Build process generates a script file that contains all of the CREATE and ALTER SQL statements so that you can review them before running the SQL through another SQL command processor. If the generated script file meets your requirements, a database administrator (DBA) can run the script later. The Build process can produce multiple scripts during a single run—one for each build option—depending on the script settings that you specify at runtime. For example, you can specify that the Build process generate a separate script for your tables, views, indexes, triggers, and alters, or you can have all of the SQL for each action contained in one script. Use the Scripts tab of the Build Settings dialog box to specify where the SQL script is written, whether you want multiple scripts generated for each definition type, and whether you want previous scripts overwritten.

## Selecting the Build Scope

The Build feature includes three different scopes that determine the size of your build project. Depending on what you actually want to build—one record or an entire project—you can determine this by selecting one of the scopes described in the following sections.

### **Current Definition**

Select Current Definition from the Build menu to build or alter the active record definition. *Active definition* means the definition that is currently active in the definition workspace.

### **Project**

Select Project from the Build menu to build or alter all of the records, indexes, triggers, and views in the current project—that is, all of the records that appear in the project workspace. You typically find numerous record definitions in the Build Scope list box.

To build all definitions in a database, create a new project and insert all of the appropriate records. For example, insert records of a particular type, such as views. If you create another record after the *all records* project is built, you must manually add the new record to the project.

You may also build a project using the command line and specific parameters. The prerequisites for using the command line to build a project are:

- The project exists with all objects.
- The build settings need to be set in the Windows registry.

---

**Note.** The Build Project process accesses every single record that is in the project, including work records, even if they are marked to be deleted.

---

### **Select Definitions Into Project**

In some situations, you might want to build a subset of the records in the project that is currently open. If you do not want to build an entire project or even the current project, you can build only the definitions that you select in the project workspace by pressing the ctrl key and selecting multiple records in the project workspace. After you select the appropriate records, right-click and select *Build* from the pop-up menu. Only the definitions that are selected in the project workspace appear in the Build Scope list box.

---

**Note.** You must invoke the Build process from the pop-up menu; you cannot build selected definitions from an open project using the Build menu.

---

## Selecting Build Options and Running the Build Process

Access the Build dialog box.

<b>Build Scope</b>	This list box displays all of the definitions that are included in the current Build process. The records that appear in this list box cannot be selected or cleared. If you must narrow the scope of your build, do so before accessing the Build dialog box.
<b>Build</b>	Click this button to start the Build process. While the Build process is running, the Build Progress dialog box appears, showing the current record being processed, the record number being processed in the series, and the total number of records that comprise the build. To stop a running Build process, click the Cancel button. When the Build process completes, the Cancel button is unavailable and the Close button is the only valid button.
<b>Settings</b>	Click this button to open the Build Settings dialog box, where you can view or change your options.

---

**Note.** After the Build process begins, there are no runtime prompts until the process is complete. All events are written to the Build log.

---

### ***Build Options***

The Build Options group box enables you to specify what action you want to occur: Create Tables, Create Indexes, Create Views, Alter Tables, or Create Trigger. If you select Create Tables or Alter Tables, Create Indexes is automatically selected by default. Also, if you select Alter Tables, Create Indexes and Create Trigger are automatically selected by default.

---

**Note.** You can select both Alter Tables and Create Tables to run concurrently, as long as the Skip table if it already exists option is enabled on the Create tab in the Build Settings dialog box. Otherwise, there is no way to determine whether you wanted to alter or re-create a table that already exists.

---

### ***Build Execute Options***

<b>Build script file</b>	Select this option to review and update the SQL before running the script. This is the safest method.
<b>Execute SQL now</b> (execute Structured Query Language now)	Select this option if you don't want to invoke another program to run the SQL; the SQL runs as part of the Build process. The disadvantage with running the SQL immediately is that you have no opportunity to review the SQL before it is committed to the database.
<b>Execute and build script</b>	Select this option to review the SQL that the Build process just ran.

To run the SQL script file that the Build process generates, you must use a third-party SQL interpreter, such as Oracle SQL\*Plus or Microsoft ISQL/W. Most database vendors include a native command processor that you can run on the client or the server.

---

**Note.** The Execute SQL now option is unavailable if you use the Alter Tables Build option. However, you can use the Execute and build script option when using Alter Tables. Selecting this option builds a script with the entire alter SQL. This script may help if errors are encountered during the alter process. However, if there are multiple tables in an alter process and an error occurs, the errors are reported to a log file and the process continues to the next table.

---

## Using Command Line Syntax to Build a Project

Use the same command line syntax as identified previously to build a project.

See [Chapter 19, "Using Command Line Parameters," Understanding PeopleSoft Application Designer Command Line Parameters, page 497.](#)

Use the command line statement for the project build to:

- Connect the project to the source database.
- Use the build settings from the Windows registry to generate the SQL script for the create or alter processes.

Before running the build command line, verify that:

- The project's system components are created and loaded.
- The build options are set in the Windows registry.

### Available Parameters

This table lists all of the available parameters for the project build statement:

<i>Parameter</i>	<i>Description</i>	<i>Error Handling</i>
-HIDE	Use to hide the PeopleSoft Application Designer interface. Always use this parameter when performing a command line Copy process.	Required for all project command line processes.
-CT	Database type. Enter the type of the database to which you are connecting (MICROSFT, ORACLE, SYBASE, and so on).	Required. If you omit this parameter, the last database type is taken from the registry. If it fails, the process stops and error messages are written to the log file.
-CS	Server name. Enter the name of the source database server for the database to which you are connecting.	Required for some database types. If you omit this parameter, the process stops and error messages are written to the log file.

<b><i>Parameter</i></b>	<b><i>Description</i></b>	<b><i>Error Handling</i></b>
-CD	Database name. Enter the name of the source database to which you are connecting, as you would enter it into the PeopleSoft Signon dialog box.	Required. If you omit this parameter, the process stops and error messages are written to the log file.
-CO	User ID. Enter the PeopleSoft user ID that is needed to sign in to the source database.	Required. If you omit this parameter, the last database type is taken from the registry. If it fails, the process stops and error messages are written to the log file.
-CP	User password. Enter the password for the specified user ID for the source database.	Required. If you omit this parameter, a PeopleSoft Signon dialog box appears for the user to enter a valid user password. If the password fails, the process stops and error messages are written to the log file.
-PJB	Project name. Enter the name of the project to be built. This project should be available in the database before starting the command line project build.	Required. This is the main parameter and is used internally to decide whether the user can build a project. If you omit this parameter and if you supply all of the source database login parameters, the program only starts the application.

### ***Example***

For this example, assume that the:

- Project name is CJR1.
- Database type is Microsoft.
- Database name is CJR810G.
- User ID is PTDMO.
- Password is PTDMO.
- Path name of the SQL script file is c:\temp\psbuild.sql.
- Path name of the log file is c:\temp\psbuild.log in the Windows registry.

For this example, you would enter the following at the command line:

```
PSIDE.EXE -CT MICROSOFT -CD CJR810G -CO PTDMO -CP PTDMO -PJB CJR_PRJ
```

## Creating Tables

This section provides an overview of the Create Table process and discusses how to:

- Create a table.
- Specify Create options.
- Specify Logging options.
- Specify Script options.
- Confirm a table build.

## Understanding the Create Table Process

The Build process generates the appropriate SQL Create statements to build tables based on both your record definition and your platform. It prefaces each new application SQL table with *PS\_* to identify it as an application that was built using PeopleTools. For example:

```
PS_ABSENCE_HIST
```

---

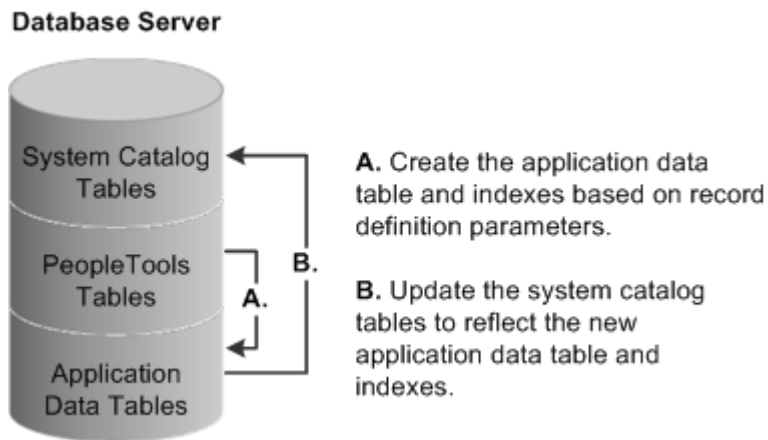
**Note.** PeopleTools does not preface table names with *PS\_* if you specified a custom name in the Non-Standard SQL Table Name edit box located on the Type tab of the Record Properties dialog box.

---

The Build process also creates the appropriate indexes. Because long character fields operate more efficiently in some database environments when they're located at the end of the table, the Build process automatically positions long character fields at the end of the table for each database environment.

The Create Table process creates a new application table based on parameters defined in the record definition. When a new table is created, the database management system updates the system catalog tables to reflect the attributes of the new table.

The Create Table process is illustrated by the following figure:



Create Table process

After you build an updated SQL build script and run it, the PeopleTools tables and the system catalog tables are synchronized; the record definition and the application data table are synchronized. The following table lists the steps of the Create Table process and the associated record definition parameters:

<i>Create Table Process</i>	<i>Record Definition Parameter</i>	<i>Usage</i>
Drop the table if it already exists.	NA	NA
Create the application data table.	Record definition name. Field names. Field type. Field length.	Table name (add <i>PS_</i> prefix). Column names. Column type. Column length.
Create indexes.	Key fields. Alternate search key fields. Custom indexes.	Unique index: ( ) Alternate index: 10 (0–9) per table. Custom index: (A–M) per table.

The following procedure covers all of the high-level steps that you need to successfully complete the Create Table build process. Steps that involve a variety of options include links to the area where those options appear in the PeopleBooks.

---

**Note.** You can have a maximum of 500 columns per table.

---

## Creating a Table

To create a table:

1. Open the project for which you want to build SQL tables.
2. Select Build, Project.
3. From the Build menu, select the appropriate scope of your build.

See [Chapter 7, "Administering Data," Selecting the Build Scope, page 151](#).

The Build dialog box appears.

4. Select Create Tables in the Build Options group box.

PeopleSoft automatically activates the Create Indexes option. If you're creating tables, the indexes that are used to extract information from those tables must also be updated. Definitions to be built appear in the Build Scope list box, which does not allow you to edit, remove, or add definitions.

5. Select one of the Build Execute options.

The default option builds an SQL script file containing the commands to run the selected build options.

6. Click the Settings button in the Build dialog box to set user-defined defaults.

The Create tab in the Build Settings dialog box appears.

7. Select the appropriate settings on the Create tab.

See [Chapter 7, "Administering Data," Specifying Create Options, page 157.](#)

Re-creating an existing table removes all data contained in the table as well as views or grants referencing that table.

8. Select the Logging tab.

9. Set the Logging Levels and associated options.

See [Chapter 7, "Administering Data," Specifying Logging Options, page 159.](#)

10. Select the Scripts tab.

11. Specify the script file options.

If you selected Execute SQL Now as your build option in the Build dialog box, you can skip the Scripts tab, because you will be running the SQL online. Settings on the Scripts tab are relevant only if you are building a SQL script to run later.

See [Chapter 7, "Administering Data," Specifying Script Options, page 162.](#)

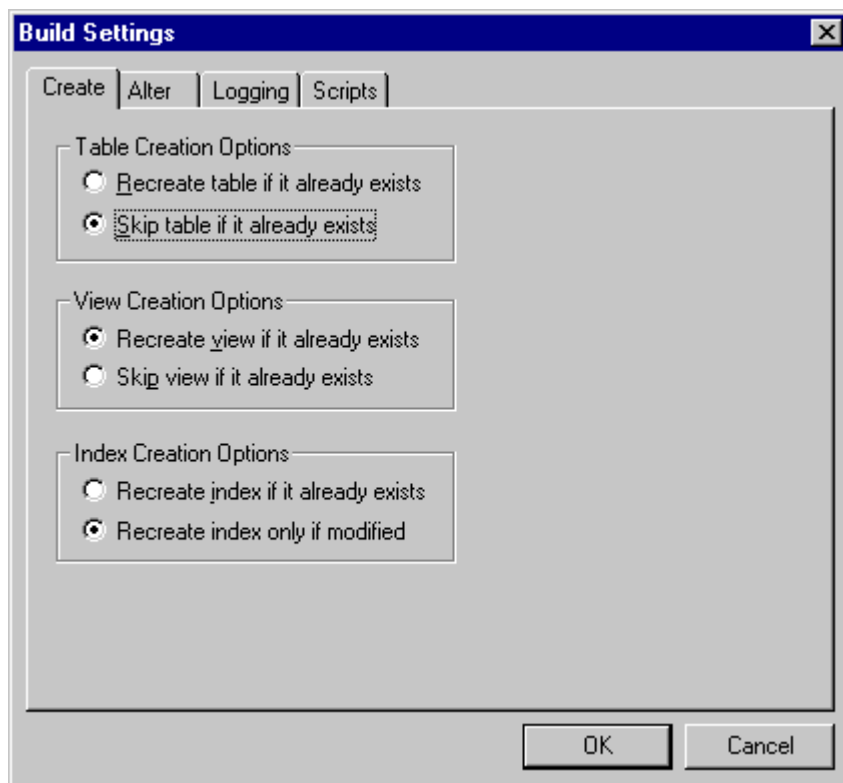
12. Click OK to close the Build Settings dialog box.

13. In the Build dialog box, click the Build button to run your build option.

The length of a build process depends on the number of definitions that require building. Watch the Build Progress dialog box to monitor the Build process. When the process completes, check any errors listed in the log file.

## Specifying Create Options

Access the Build Settings dialog box. Select the Create tab.



Build Settings dialog box: Create tab

### ***Table Creation Options***

These options determine when a table should and *should not* be created.

#### **Recreate table if it already exists**

Select to drop and re-create a table if it already exists. Select this option with extreme care because if data already exists in the table, it is also dropped. If you select this option, the Build process prompts you to confirm your intention before performing any destructive action. If you don't care about losing the data that resides in the table, then this option is faster than the Alter Tables option.

#### **Skip table if it already exists**

Select to create only those tables that are newly defined. If you want to preserve the data that already resides in existing tables or if you're just interested in creating the tables that do not already exist, select this option.

### ***View Creation Options***

Similar settings are available for creating views that you use for creating tables.

#### **Recreate view if it already exists**

Because views are just logical views of data, it is safe to select this option; you don't run the risk of losing physical data. Using this option ensures that all views in the project are rebuilt, whether they need to be or not.

**Skip view if it already exists**

If you're concerned only with creating views that do not already exist in the database, select this option. This option is useful if you want to run Build Project on a large project and only a subset of the views in the project must be created. This consumes less time than re-creating all of the views in the project.

**Index Creation Options**

These options determine when an index should be re-created.

**Recreate index if it already exists**

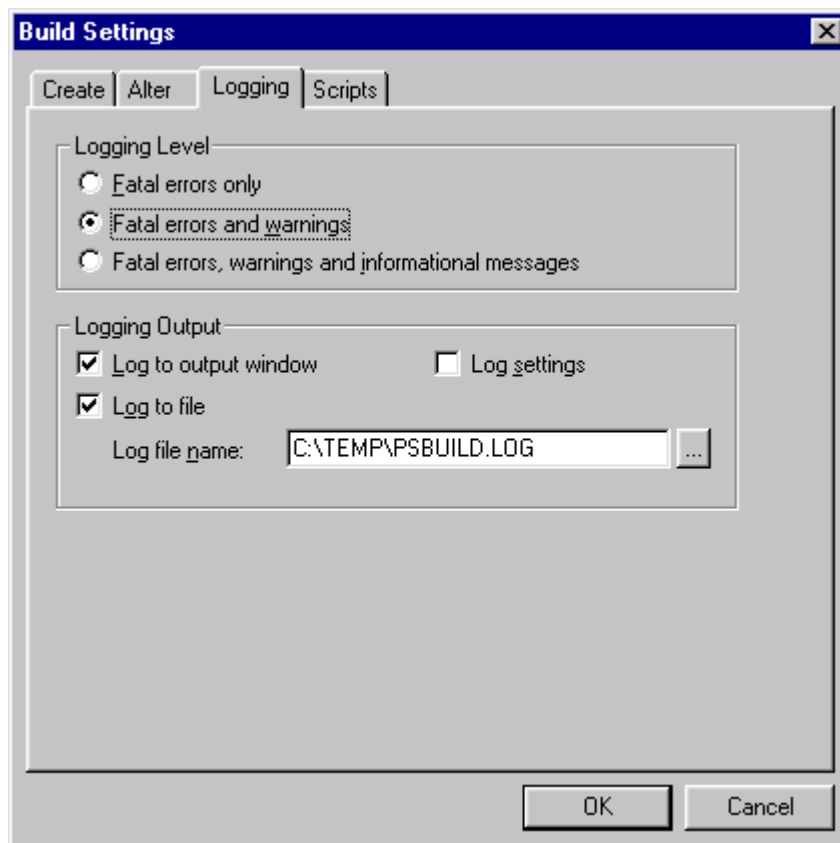
Select to re-create the index for your tables, even if you already have one in place. It re-creates the index no matter what the situation.

**Recreate index only if modified**

Select to re-create the index only if the data is being modified in the indexes.

## Specifying Logging Options

Access the Build Settings dialog box. Select the Logging tab.



Build Settings dialog box: Logging tab

## Logging Level

Specify the detail of information that you want in your Build log output.

<b>Fatal errors only</b>	Select this option if you want to see only the operations that failed.
<b>Fatal errors and warnings</b>	Select this option to add warnings to the fatal errors.
<b>Fatal errors, warnings and informational messages</b>	Select this option to see everything that processed successfully <i>and</i> all of the errors and warnings.

---

**Note.** These options are personal preferences and do not affect the actual build of your SQL tables, but they might help you track down potential failures. For example, if you're processing a large number of definitions, you'll want to see only the operations that failed. Whereas, if you're processing a small number of definitions, you'll want to see all of the successful operations too.

---

## Logging Output

Specify where and in what form you want to view your log output.

<b>Log to output window</b>	Select to write the same information that appears in the log file to the PeopleSoft Application Designer output window. Always be aware of the level that you select. If there is a lot of detail (high logging level plus many records to process), it might be easier to copy the logging information to a file and print it later.
<b>Log to file</b>	Select to write the log to a file. If you select this check box, you can specify a location and name for your log file in the Log file name edit box. To search for a name, click the Browse button. Select a .LOG type file from the Build File Selection dialog box. If you do not select this check box, the edit box is unavailable for entry.
<b>Log settings</b>	Select to write all of the runtime options to the log. If you select this check box, it is not necessary to rerun a test when you have a problem. Should it be necessary to report a problem with the Build process or troubleshoot, it is important to know the settings that were active.

---

**Note.** PeopleSoft recommends selecting the Log to file check box *and* the Log to output window with a minimum logging level of fatal errors and warnings. In case you run into a problem, you already have the information that is needed to research the problem. Otherwise, you must run the process again with a higher logging level.

---

## Example Log Results

The following are samples of the type of information that you can expect to see in your log output that corresponds to the selections that you make on the Logging tab:

- Fatal errors:

```
SQL Build process began on 7/30/2003 at 8:46:46 PM for database
OM51U20
Error: BI_BILL_BY_F_VW - Unable to retrieve current DDL space name.
Alter processed failed. (76,47)
Error: BI_CHARGE_VW - Unable to retrieve current DDL space name.
Alter processed failed. (76,47)
Error: PS_BI_CYCLE_VW - alter failed due to an unknown column type
(UNDEFINED) found for field BILL_CYCLE_ID. (76,22)
Error: PS_BI_IVCLAYOUT_VW - alter failed due to an unknown column
type (UNDEFINED) found for field INVOICE_LAYOUT_ID. (76,22)
Error: CARINA - alter failed because SQL table does not exist. (76,7)
SQL Build process ended on 7/30/2003 at 8:54:34 PM.
2487 records processed, 11 errors, 0 warnings
SQL Build script for all processes written to file C:\TEMP\step38.SQL.
SQL Build log file written to C:\TEMP\step38.LOG
```

- Warnings:

```
SQL Build process began on 8/27/2003 at 4:00:32 PM for database DXD7L
Warning: ABSENCE_HIST2 - data would be truncated by altering
REASON. Record processed. (76,24)
```

- Fatal errors, warnings, and informational messages:

```
SQL Build process began on 8/27/2003 at 3:58:58 PM for database DXD7L
Warning: ABSENCE_HIST2 - data would be truncated by altering
REASON. Record processed. (76,24)
Error: ABSENCE_HIST2 - field PAID_UNPAID to be deleted has data
present. Record skipped. (76,23)
Informational: ACCOMPLISHMT_VW - view created and scripted
successfully. (76,18)
Informational: ACCT_AD2_INV_VW - view created and scripted
successfully. (76,18)
Informational: ACCT_ADJ_INV_VW - view created and scripted
successfully. (76,18)
Informational: ACCT_ISS_INV_VW - view created and scripted
successfully. (76,18)
Informational: ACCT_STK_INV_VW - view created and scripted
successfully. (76,18)
Informational: ACCT_TYPE_VW - view created and scripted
successfully. (76,18)
Informational: ACTION_SP_VW - view created and scripted
successfully. (76,18)
Informational: ACTION_XLAT_VW - view created and scripted
successfully. (76,18)
Informational: ADJUST_INV2_VW - view created and scripted
successfully. (76,18)
Informational: ADJUST_INV_VW - view created and scripted
successfully. (76,18)
11 records processed, 1 errors, 1 warnings
SQL Build script for all processes written to file
C:\TEMP\PSBUILD.SQL.
SQL executed online.
SQL Build log file written to C:\TEMP\PSBLD.log
```

- Log settings:

```

**** Begin Build Settings for this Run
Create Tables = No
Create Indexes = Yes
Create Views = No
Create Triggers = Yes
Alter Tables = Yes
Log to File = Yes
Log to Window = Yes
Write comments to script = No
Always overwrite files = Yes
Execute SQL Now = No
Write SQL to script file = Yes
Logging level = Log fatal errors, warnings and informational messages
Table creation option = Skip table if it exists
View creation option = Drop and recreate view if it exists
IndexCreationOption = Drop and recreate index always
Alter Adds = Yes
Alter Changes = Yes
Alter Renames = Yes
Alter Deletes = Yes
Alter even if no changes = No
Alter Table Option = Alter by Table Rename
Write script output to: Single file
Log filename = c:\temp\PSBUILD.LOG
Single script filename = c:\temp\PSBUILD.SQL
Alter drop column option = skip record if data present
Alter truncate column option = skip record if data would be truncated
Target database platform = MICROSOFT
Target sizing set = 0

```

### ***Working With the Output Window***

You can navigate from error messages in the PeopleSoft Application Designer output window to the corresponding record definition by selecting the appropriate line in the Build log and double-clicking anywhere on that line. PeopleSoft Application Designer opens the corresponding record in the definition workspace.

Also, you can double-click any line in the output window that corresponds to a file—not just a record. PeopleSoft Application Designer opens that file with whatever application that you defined to open files with a particular extension. For example, if you associated SQL files with your native SQL command utility, your SQL utility opens and loads your PSBUILD.SQL script when you double-click the line that reads:

```
Single script file name = C:\TEMP\PSBUILD.SQL
```

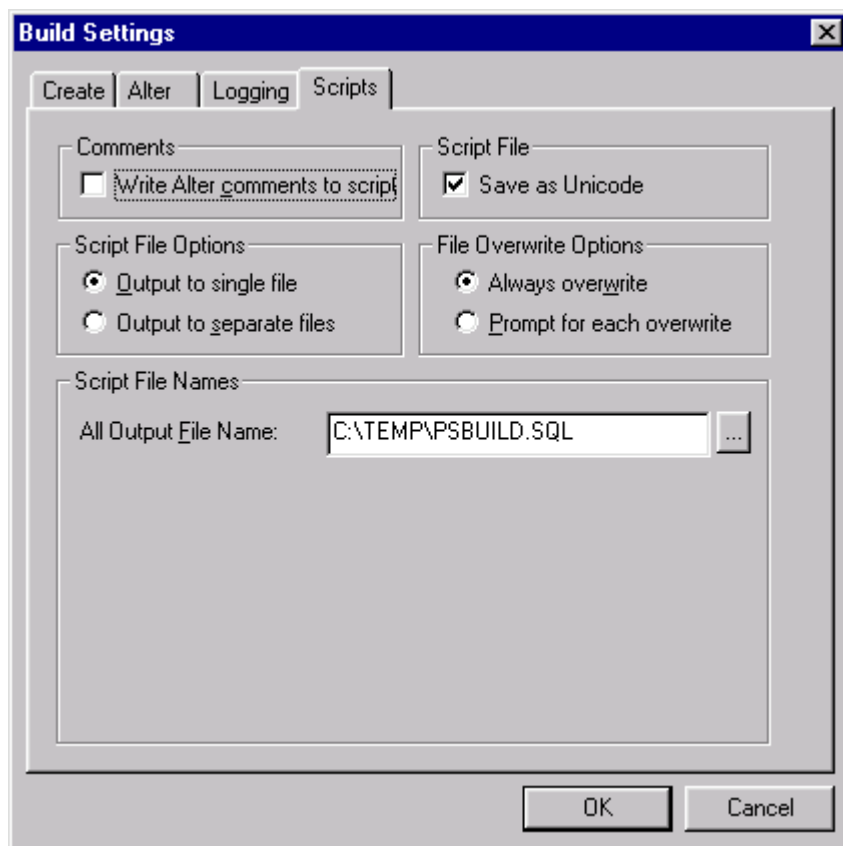
---

**Note.** PeopleSoft Application Designer uses the standard Microsoft Windows method for defining which programs are associated with particular file types. To modify the file associations, open My Computer, select Tools, Folder Options, and then select the File Types tab.

---

## **Specifying Script Options**

Access the Build Settings dialog box. Select the Scripts tab.



Build Settings dialog box: Scripts tab

### **Comments**

The Write Alter comments to script check box enables you to either include or suppress alter comments. The following example displays the types of comments that you see in your PSALTER.SQL script if you enable this option:

```

-- Alters for record PS_AE_RUN_CONTROL ;
--           AE_THREAD - change ;
--           OPRID - change ;

-- Start the Transaction ;

-- Data Conversion ;

-- Drop Old Indexes ;

DROP INDEX SYSADM.PS_AE_RUN_CONTROL
/

-- Add Columns ;

ALTER TABLE PS_AE_RUN_CONTROL ADD Z_TMP_ALTER_1 DECIMAL(12,4)
/
ALTER TABLE PS_AE_RUN_CONTROL ADD Z_TMP_ALTER_2 CHAR(12)
/
UPDATE PS_AE_RUN_CONTROL SET
    Z_TMP_ALTER_1 = AE_THREAD,
    Z_TMP_ALTER_2 = OPRID
/

```

If you do not view the alter comments, the script containing the same commands as the previous script looks like this:

```

DROP INDEX SYSADM.PS_AE_RUN_CONTROL
/
ALTER TABLE PS_AE_RUN_CONTROL ADD Z_TMP_ALTER_1 DECIMAL(12,4)
/
ALTER TABLE PS_AE_RUN_CONTROL ADD Z_TMP_ALTER_2 CHAR(12)
/
UPDATE PS_AE_RUN_CONTROL SET
    Z_TMP_ALTER_1 = AE_THREAD,
    Z_TMP_ALTER_2 = OPRID
/

```

### Script File

The default for the Save as Unicode check box is based on the UNICODE/ANSI database. This check box is available only when using MSSQLServer, Oracle, DB2/UDB for Linux, UNIX, and Windows, Sybase and DB2 UDB for OS/390 and z/OS databases. Only on Informix , the check box is permanently unavailable, because the script files are always ANSI.

The following table explains the default settings for the Save as Unicode check box:

<b>Database Type</b>	<b>Default for Save as Unicode Check Box</b>
Unicode	Selected
ANSI	Unavailable

The script file option, Save as Unicode, is stored in the registry with the other build settings. Therefore, the previous setting is always the default every time that the dialog box is opened.

The type of script file that is generated during the Build process is based on the Save as Unicode check box setting for the UNICODE/ANSI build and UNICODE/ANSI database.

The details are as follows:

<b>Database Platform</b>	<b>Unicode</b>	<b>ANSI</b>
MSSQL Server	UCS2	ANSI
Sybase	UTF8	ANSI
Oracle	UTF8	ANSI
DB2/UDB for Linux, Unix, and Windows	UTF8	ANSI
DB2 UDB for OS/390 and z/OS	UTF8	ANSI

### **Script File Options**

If you want all of your CREATE TABLE statements to be written to one SQL script file and your CREATE INDEX statements to another, then select the Output to separate files option. On the other hand, if you prefer to have all of your statements—CREATE, ALTER, and so on—contained in a single file, then select the Output to single file option.

---

**Note.** Script files can be quite large. To reduce the size of files so that they are easier to manage, select the Output to separate files option. Also, the system never appends script files; it overwrites them or you can terminate the process if you elected to be prompted before a build script file is overwritten.

---

Depending on which output option you select, the options in the Script File Names group box change to suit one file or multiple files.

### **File Overwrite Options**

These options indicate whether you want to overwrite automatically or be prompted before script files are overwritten. Your personal preference determines which technique you employ.

**Always overwrite**                      Select to overwrite your previous build scripts.

**Prompt for each overwrite**                      Select to be alerted when a script is about to be overwritten.

To avoid the possibility of overwriting files altogether, you can use a naming convention that, in itself, prevents any overwrites. For example, you could specify a unique name for each build script, as shown:

- C:\TEMP\PSINDEX1.SQL
- C:\TEMP\PSINDEX2.SQL

## ***Script File Names***

The options available to select in this group box depend on your selection in the Script File Options group box.

### **Output to single file**

When you select this option, an edit box appears in the Script File Names group box: All Output File Name. The default name for the generated script is PSBUILD.SQL.

---

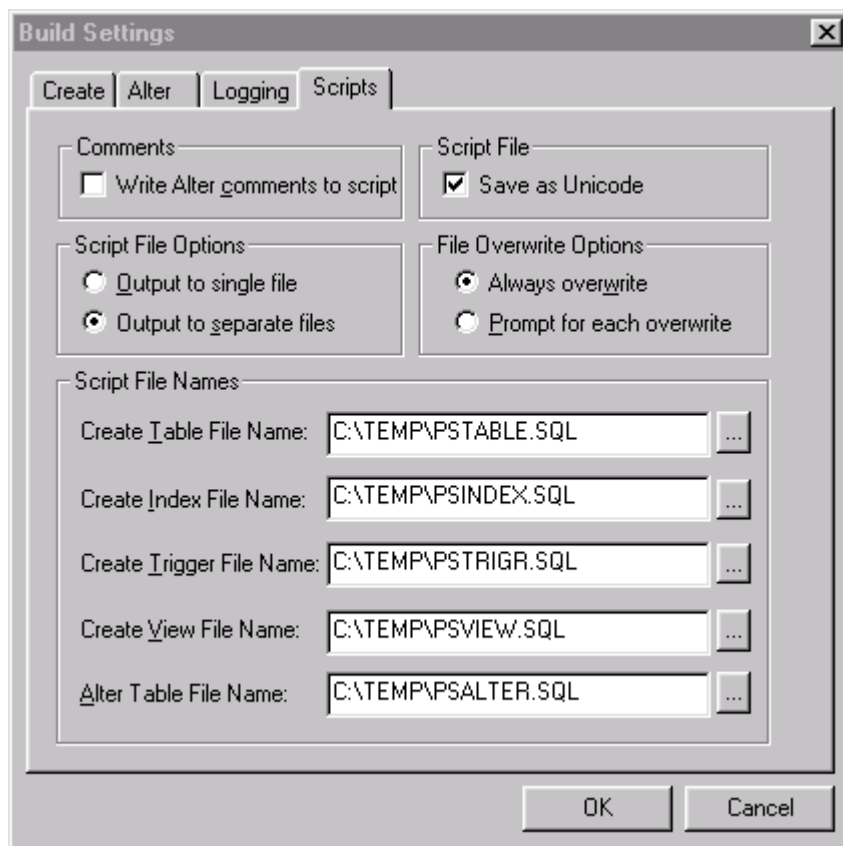
**Note.** Script files can be quite large. To reduce the size of files so that they are easier to manage, select the Output to separate files option. Also, the system never appends script files; it overwrites them or you can terminate the process if you elected to be prompted before a build script file is overwritten.

---

### **Output to separate files**

If you select this option, the following edit boxes appear—one for each build option:

- Create Table File Name: The default name for this script is PSTABLE.SQL.
- Create Index File Name: The default name for this script is PSINDEX.SQL.
- Create Trigger File Name: The default name for this script is PSTRIGGER.SQL.
- Create View File Name: The default name for this script is PSVIEW.SQL.
- Alter Table File Name: The default name for this script is PSALTER.SQL.



Output to separate files

## Confirming a Table Build

If you are running SQL online, complete steps 1 and 3. Step 2 is not required.

To confirm a table build:

1. Review the SQL script that was generated by the Build process.

Use your native SQL command processor to open the SQL script. The script is located where you specified on the Scripts tab of the Build Settings dialog box.

2. Run the script against your database.
3. Confirm that the tables now exist.

Use your query tool and SQL Select statements to confirm that the Create Table process created an application table that corresponds to your record definition and updated the system catalog tables. The SQL statement that you use to confirm depends on the table that you just created.

---

## Creating Indexes

This section provides an overview of indexes and discusses how to:

- Configure indexes.
- Add a custom index.
- Edit the DDL.

## Understanding Indexes

Indexes are an important part of your physical database, because they affect the efficiency and speed with which your application can store and retrieve data. PeopleSoft application indexes are defined in two ways.

Some indexes are defined for you automatically, based on the key, search key, list box items, and alternate search keys that you specified on your record definition. These indexes are used by the application processor for quick access when you use the online system.

However, it is sometimes necessary to define additional indexes to improve the performance of demanding queries, reports, or batch programs. These additional indexes are defined and modified using the data administration tools in PeopleSoft Application Designer.

---

**Note.** You can only create indexes for record definition types that are SQL or temporary tables. You would not create indexes on views, such as a SQL view, because views are never physically created. The Indexes option in the Tools, Data Administration menu is disabled when a view is open.

---

When you use SQL Create or SQL Alter on a table, the system automatically builds database indexes to the SQL table that are based on the keys and list items that you specify in the record definition. Your database uses indexes to find definitions in the database the same way that you use indexes to find specific information in a book. If an index entry exists for a specific field on a table, the database can search and quickly find what it needs to complete your request; otherwise, it must search through the contents of the entire table.

Indexes enhance system performance by generating key lists that the application processor uses to extract the data that it uses to populate pages and search records. The system automatically creates:

- Standard indexes (key or duplicate order key indexes), if at least one field in the table is a key or duplicate order key.

The index contains all key and duplicate order key fields. The system automatically names this index with a prefix of *PS\_*.

- Alternate search key indexes for each alternate search key.

The system automatically names this index with a prefix of *PSn*, in which *n* is a number between 0 and 9.

- Custom indexes, which can be defined with the Edit Index, Edit DLL, and Add Index dialog box in the PeopleSoft Application Designer administration tools.

The system automatically names this index with a prefix of *Psa*, in which *a* can be any letter between *A* and *M*. Custom indexes are not automatically created.

For most database tables (those with unique keys), the key index defines unique keys for a table, but the system also uses it to access high-level keys and orders by all keys. The alternate search key indexes support the search record and field prompts. The purpose of these different indexes is to enable the system to respond to all system prompts through index-only SQL access, minimizing direct database table access.

---

**Note.** If you are using Create Table or Alter, the Create Index option is automatically selected. With the Create Table or Alter options, you can't clear the Create Index option.

The alter process takes care of indexes regardless of the Alter Table Options you select on the Alter tab of the Build Settings dialog box. In the case of "Alter by Table Rename," the indexes are dropped automatically when the temp table is dropped. For the "Alter in Place" option, the index creation process goes through the "Recreate index only if modified" option on the Create tab.

When creating indexes with the Alter method, prior to altering the table, drop the statistics using "DROP STATISTICS" that were created explicitly on columns using "CREATE STATISTICS."

---

The following record definition modifications require the creation of new indexes:

- Add, delete, or modify keys, duplicate order keys, alternate search keys, or descending keys in the Record Field Properties dialog box.
- Change the order of keys (ascending, as opposed to descending) in the Change Record Indexes dialog box.
- Change the Unique settings in the Edit Index dialog box.

### ***Platform Support for Indexes***

Most database platforms support indexes with an unlimited number of keys. However, Microsoft SQL Server 2000, DB2/UDB for Linux, UNIX, and Windows, and Informix support only indexes that have a maximum of 16 keys. PeopleSoft supports indexes for these databases by creating functional indexes. If a record definition has more than 16 keys, the Build process automatically alters the table to support functional indexes. The system adds a warning message to the log file indicating this. Sample code for functional indexes for each of these platforms can be found in Appendix A of this PeopleBook.

See [Appendix A, "Understanding Functional Indexes," page 539](#).

The following list offers more detail on functional indexes by database platform:

- Functional indexes for Microsoft SQL Server 2000.

If an index has more than 16 keys, the key fields are concatenated to a functional key. This key field is then added to the table as the last column called MSSCONCATCOL. A unique index is created for MSSXCONCATCOL as well as an additional index called PSW<record name> that is used as a search index.

- Functional indexes for DB2/UDB for Linux, UNIX, and Windows.

Similar to the concatenation that occurs with SQL Server, DB2/UDB for Linux, UNIX, and Windows also concatenates the key fields to a functional key. The key is then added to the table as the last column called DBXCONCATCOL. A unique index is created for DBXCONCATCOL as well as an additional index called PSW<record name> that is used as a search index.

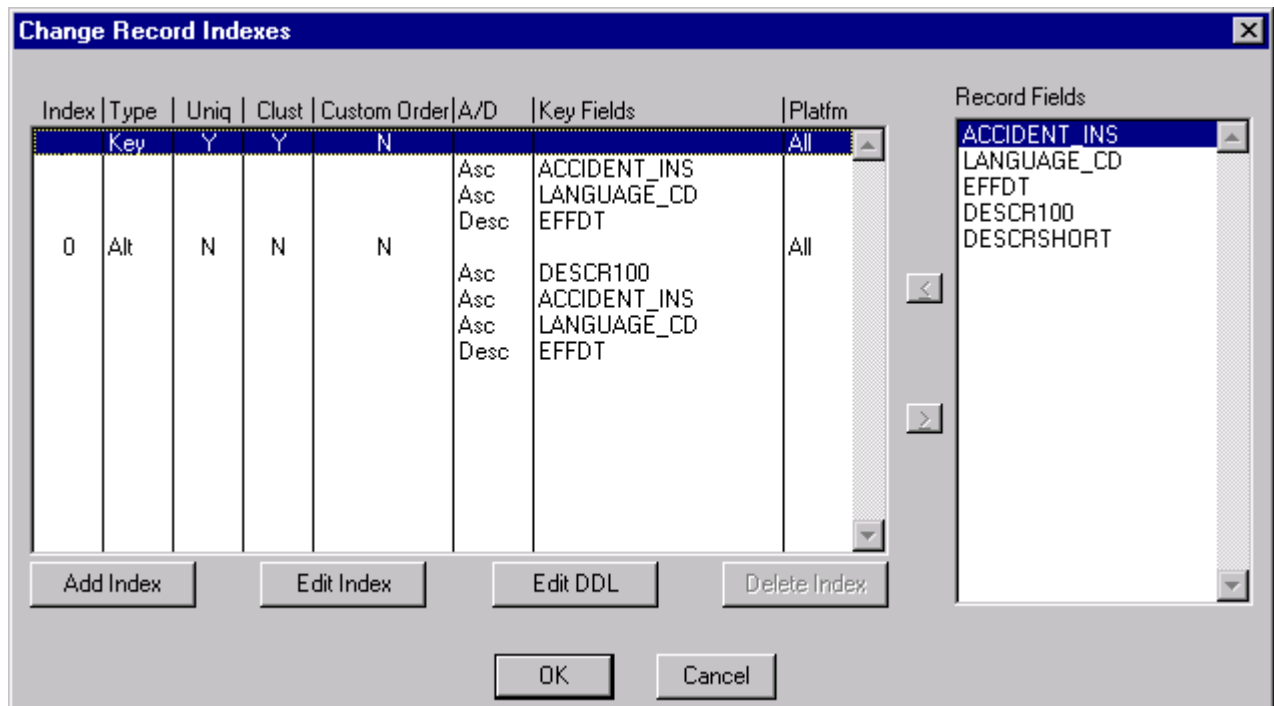
- Functional indexes for Informix.

For Informix, if there are more than 16 keys in an index, a function is created for every 16 keys. The name for this set of 16 keys is PS\_<record name>\_SP<the incremented number for each function>. The functions are used to create a unique index. An additional index called PSW<record name> is also created.

## Configuring Indexes

For performance tuning, you might want to change the order of your index keys. PeopleSoft Application Designer provides direct control over all of the indexes created by PeopleTools. You can inspect, add, or change all defined indexes or delete custom indexes for any record in the system through the Change Record Indexes dialog box.

Standard indexes are based on the key, duplicate order key, or alternate search key only. However, with the Change Record Indexes dialog box, you can modify those indexes and create additional indexes, which might be required if you do custom processing. Any of the indexes that you entered can be maintained for all database platforms or for a list of specific platforms. All indexes and keys created by PeopleTools are stored in PeopleSoft system tables.



Change Record Indexes dialog box

To edit the index key order:

1. Open the record definition for which you want to make the custom index.
2. Select Tools, Data Administration, Indexes to access the Change Record Indexes dialog box.

The Custom Order column indicates whether the index keys have been configured. The default selection is *N*, indicating that no configurations have been made.

3. Double-click the *N* or *Y* in the Custom Order column for the index that you want to alter, or click Edit Index.
4. Select the Custom Key Order check box.
5. Click OK.

The Custom Order entry changes from *N* to *Y*.

6. Drag and drop the field that you want to move into the appropriate order.

After configuring the key order, to reset the key order of the index to the original order as defined in the record definition, clear the Custom Order entry (Y to N), and click OK. The orders of the key fields reset to the original position in the Change Record Indexes dialog box.

7. Check the key order in the Use Display mode of your record definition.

The Num (number) column represents the original order of the keys in your record and the Ord (order) column reflects the key order in the index.

8. Create the indexes:

- a. Select Build, Current Project.
- b. Select Create Indexes in the Build Options group box.

Make sure that you select the appropriate build settings.

9. Run the Build process to either generate a script or run the SQL online.
10. Confirm the construction of the new index with your native SQL command processor.

---

**Note.** Users might also configure indexes for "parent" records that contain one or more subrecords without key fields, such as the subrecord ADDRESS\_SBR. However, if the subrecords contain key fields, index key order changes are restricted. The Custom Key Order option is deactivated and cannot be changed from *N* to *Y*. Change the index key order for the parent before inserting the subrecord.

---

## Adding a Custom Index

To add an index:

1. Open the record.
2. Select Tools, Data Administration, Indexes to access the Change Record Indexes dialog box.
3. Click Add Index.

The Add Index dialog box enables you to determine for which database platforms you want to define the index if you selected the Some option for Platform. The Custom Key Order check box has been selected by the system to ensure that the underlying record definition is not adversely impacted by the change to your index.

For all user-defined indexes, you must enter a description of the index in the Comments section. It is useful to know the purpose of the index, the transactions and processes that use this index, and other relevant information. This information assists you when you want to remove unused indexes. An error message appears if you do not do so.

---

**Note.** You cannot add more than 22 user indexes (A-V). The system disables the Add Index button if the record already contains 22 user indexes.

---

4. Indicate the database platforms for which you want to create the index.
5. Click the OK button to add a row to the Change Record Indexes dialog box and enable the arrow buttons.

6. Move the field under the new index row:
  - a. In the Record Fields list box, select the field that you want indexed.
  - b. Click the arrow button, or double-click the field.
  - c. Repeat this step for each field that you want to add.
7. Click OK.
8. Save the record definition.
9. Select Build, Current Project.
10. Select Create Indexes under Build Options.

Make sure that you select the appropriate build settings.
11. Run the Build process to either generate a script or run SQL online.
12. Run the generated SQL script if you opted to generate one.
13. Confirm the construction of the new index with your native SQL command processor.

## Editing the DDL

In the Change Record Indexes dialog box, you can view the DDL for the index and override DDL parameters that are defined in the DDL model for this index.

To edit the DDL:

1. In the Change Record Indexes dialog box, select the index that you want to edit.
2. Click the Edit DDL button to access the Maintain Index DDL dialog box.

The upper list box in this dialog box lists the platform and DDL parameter, and the lower list box shows the DDL templates for the various platforms. Your database might have fewer platforms enabled. The sizing set enables different collections of tables to have different model statements and parameters.

3. Select the platform and sizing set.
4. Click View DDL.

The dialog box that appears shows the index DDL for the platform and sizing set that you selected. Review this DDL statement.

5. Click the Close button when you're finished.
6. Select the parameter that you want to edit in the upper list box.

The default value is set in the DDL Model Defaults dialog box. You can change this setting by selecting Go, PeopleTools, Utilities, Use, DDL Model Defaults in PeopleSoft Application Designer, or by selecting PeopleTools, Utilities, Use, DDL Model Defaults in the browser.

7. Click the Edit Parm (edit parameter) button to access the Edit Override Parm Value (edit override parameter value) dialog box.

8. Enter the override value for the parameter that you selected.
9. Select OK for both open dialog boxes.

---

## Altering Tables

This chapter provides an overview of the Alter process and discusses how to:

- Determine tables to be altered after a field change.
- Alter a table.
- Specify alter options.
- Confirm your table alter.

## Understanding the Alter Process

The Alter process enables you to make changes to your record definitions at any time without losing the application data that is already housed in the table. To ensure that the Alter process produces the appropriate results, be aware of when to alter an existing table and what types of conversions PeopleSoft Application Designer supports. To preserve data that is stored in tables and also reflect a recent change in the table definition, use the alter tables function to make the appropriate changes.

The alter function simplifies the process of modifying database tables when new data fields are added or when existing data fields are modified. It effectively eliminates the need to write SQL statements to perform the same function.

PeopleSoft specifically designed the alter process to automate the task of writing alter scripts and protect the integrity of your database. In essence, it protects your interests by ensuring that you always control data loss. The alter function performs tests that verify whether data will be lost during the conversion on each column to be altered. Data loss normally occurs when reducing the character width of a column, dropping a column that contains data, or converting a number field to a character field that is too small.

The generated statements for this method are sent to a SQL script file—the alter script. After the script has completed, you are responsible for reapplying any SQL table DDL that you applied outside of PeopleTools. Typically, this consists of GRANT and REVOKE statements controlling security.

---

**Note.** On Oracle databases, procedural language/Structured Query Language (PL/SQL) is required to use the PeopleSoft Application Designer Build process.

---

### ***Alter Settings***

The Alter tab in the Build Settings dialog box handles the most common types of data conversions, including:

- Increasing the length of character fields.
- Changing a character (CHAR) field to a long character (LONG VARCHAR) field.
- Increasing the size of number or character fields.

- Reducing the decimal positions in number fields.

If any column of the table has data that will be lost when truncating, the system action is determined by the Change Column Length options.

- Changing integer-only number fields to character fields, where the character field is formatted with leading zeros.

If any column of the table has a number that does not fit in the new character field, system action is determined by the Change Column Length options. This prevents data from being lost.

- Adding columns to SQL tables.
- Dropping columns from SQL tables.
- Renaming columns in SQL tables.
- Renaming SQL tables.
- Moving tables to a new tablespace.

The following sections describe the types of alter conversions that you can expect to occur during the Build process.

### Data Conversions

When altering tables with existing field data, PeopleSoft Application Designer handles conversions as outlined in the following table. A *No* indicates that conversion cannot take place. A number in the cell following a *Yes* indicates that there are restrictions involved with the conversion. Refer to the corresponding numbered note following the table for additional information:

PSFT Old Field Type Format	PeopleSoft New Field Type Format										
	Char	Oracle— Long Char with Length < 2000	Long Char	Image	Long Char — Raw Binary	Number/Signed Number where n = length and d = decimal			Date	Time	DateTime
						SmallInt * (small integer)	Integer **	Decimal ***			
Char	Yes (1)	Yes (1)	Yes (9)	No	No	No	No	No	No	No	No

<b>PSFT Old Field Type Format</b>	<b>PeopleSoft New Field Type Format</b>										
<i>Oracle — Long Char with Length &lt; 2000</i>	Yes (1)	Yes (1)	Yes	No	No	No	No	No	No	No	No
<i>LongChar</i>	Yes (2)( 9)	Yes (2)	Yes (8)	No	No	No	No	No	No	No	No
<i>Image</i>	No	No	No	Yes (8)	Yes (8)	No	No	No	No	No	No
<i>Long Char— RawBinary</i>	No	No	No	Yes (8)	Yes (8)	No	No	No	No	No	No
<i>SmallInt *</i>	Yes (5)	Yes (5)	No	No	No	No Action	Yes (4)	Yes (4)	No	No	No
<i>Integer **</i>	Yes (5)	Yes (5)	No	No	No	Yes (4)	No Action	Yes (4)	No	No	No
<i>Decimal ***</i>	Yes (7)	Yes (7)	No	No	No	Yes (3)	Yes (3)	Yes (3)	No	No	No
<i>Date</i>	No	No	No	No	No	No	No	No	No Action	No	No
<i>Time</i>	No	No	No	No	No	No	No	No	No	No Action	No

























- Space Name.DB Name**

(tablespace name.database name)

Displays the available space name that is assigned to the current record. This field is display-only.
- Available Space Name-DB Name**

(available tablespace name-database name)

Select the space name and database name to assign to the record.
- Records Allocated to this Space**

Displays all of the records that have already been allocated to the space that you selected from the Available Space Name-DB Name drop-down list box. This field is display-only.

See *Enterprise PeopleTools 8.50 PeopleBook: System and Server Administration*, "Using PeopleTools Utilities," Tablespace Utilities.

## Using Physical Data Storage

You have control over the physical storage of your data. The DDL for creating tables, indexes, and tablespaces can be viewed and edited in the browser.

Access the DDL Model Defaults page.

### DDL Model Defaults

Platform ID:

1

DB2

Copy...

Sizing Set:

0

DDL

Find | View All

First

1 of 5

Last

Statement Type:

Table

+

-

\*Model SQL:

CREATE TABLE \*\*OWNER\*\*.[TBNAME] ([TBCOLLIST]) IN [DBNAME].  
[TBSPCNAME];

Parameter Count:

1

Parameters

Customize | Find | View All |

First

1 of 1

Last

DDL Parm

DDL Parameter Value

OWNER

+

-

DDL Model Defaults page

To view or edit the DDL parameters for creating tables, indexes, and tablespaces:

- From the homepage of your application, select PeopleTools, Utilities, Administration, DDL Model Defaults to access the DDL Model Defaults search page.
- Press enter to view platform names.

### 3. Select a platform name.

The DDL Model Defaults page appears.

The Model SQL edit box shows the DDL template for the specified platform and sizing set. The items in square brackets are special parameters that are populated when instances of SQL are generated. For example, the preceding example shows an index model statement for ALLBASE. In this example:

- [TBNAME] is the name of the table.
- [TBCOLLIST] is replaced by the columns that are specified in the index definition.

The model statements also contain parameter names enclosed in pairs of asterisks. A parameter name is replaced by a value when instances of the SQL are generated. In the preceding example, there is one parameter, **\*\*FILESET\*\***. When DDL is generated using this model, the FILESET parameter is replaced by FILE, unless it is overridden for the specific record or tablespace. The other text in the model statement is copied to the generated SQL.

### See Also

*Enterprise PeopleTools 8.50 PeopleBook: System and Server Administration*, "Using PeopleTools Utilities," Using Administration Utilities



## Chapter 8

# Designing Pages for Optimal Performance and Accessibility

Developing pages for internet applications requires certain design techniques to optimize performance and offer the greatest accessibility for users. This section discusses how to:

- Improve online performance.
- Design accessible pages.

---

## Improving Online Performance

To improve online performance of internet applications, you must optimize the transmissions to the server. Optimizing transmissions improves overall processing speed and reduces the user's data entry time.

Transactions that are designed using deferred mode reduce traffic to the application server but do not operate interactively.

This section provides an overview of processing modes and discusses how to:

- Refresh a page.
- Trigger server processing.
- Use PeopleCode in deferred mode.
- Use page controls in deferred mode.
- Use error and warning messages.

## Understanding Processing Modes

PeopleSoft applications can run in two different processing modes: *interactive* and *deferred*.

### ***Interactive Mode***

In interactive mode (formerly called standard), when the user exits a field that has a field-level event (for example, FieldChange, FieldEdit, RowInsert PeopleCode, prompt validation, related display, and so on):

1. A transmission to the application server is performed to run that field-level event.
2. The page is displayed again.



<i><b>Event</b></i>	<i><b>Deferred Processing</b></i>	<i><b>Interactive Processing</b></i>
Entering data in fields that have prompt table edits.		X
Entering data in fields that have related displays.		X
Inserting a row or series of rows in a grid or scroll area.	X	X
Deleting a row from a grid or scroll area.	X	X
Using the grid or scroll area controls to move forward or back.	X	X
Selecting another page tab.	X	X
Selecting another grid tab.	X	X
Expanding or collapsing a collapsible section.	X	X
Clicking any button or icon on the page.	X	X
Clicking any link on the page.	X	X
Clicking the Refresh toolbar button.	X	X
Using the Refresh hot key.	X	X

## Using PeopleCode in Deferred Mode

When designing transactions to run in deferred mode, note the following:

- Code field edits in FieldEdit PeopleCode; it is not necessary to move field edits to SaveEdit PeopleCode.
- Code field change logic in FieldChange PeopleCode.
- Set all fields, including those that have FieldEdit or FieldChange PeopleCode, to deferred processing.







For FieldEdit error messages running in deferred mode, the system changes the field to red and positions the cursor to the field in error when it displays the message. This allows the user to associate the error message with a specific field. However, for warning messages, the system does not display fields, nor reposition the cursor. Consequently, create warning messages that clearly describe to which fields they apply. For example, if a page contained a "Date out of range" warning, it would be confusing to users if there were multiple date fields on the page.

---

## Designing Accessible Pages

You want to design an application page that is useful to the user and supports assistive technologies.

This section provides an overview of accessibility issues and discusses how to:

- Use labels.
- Use color to convey meaning.
- Verify titles for grids and scroll areas.
- Add unique labels for buttons and links.
- Deactivate adornment images.
- Maintain logical tab order.
- Use deferred processing.
- Use foreground and background colors.
- Use instructional text.

## Understanding Accessibility Issues

PeopleSoft is committed to designing transactions that are accessible to all users. When you create or modify PeopleSoft Internet Architecture pages for your system, consider the needs of all users who might access these pages. Make sure that users with disabilities can navigate quickly and cleanly through your transactions when using screen reader software. Before you design your page, use the following checklist to create a properly designed application page that also addresses important usability issues:

- Do all functional images have labels?  
Do these labels effectively communicate the purpose of the image?
- Do all data entry fields have labels near the entry field?
- Do all grid columns have labels (excluding columns with buttons or links)?
- Do all grids and scroll areas have titles?
- Do all buttons and links have unique labels?
- Do adornment images have labels deactivated?
- Is the tab order correct?



- If the image is an active link, button, or area, then choose the text alternative to be the function of the image.
- If the image is not active but conveys information choose the text alternative to convey the same information.
- If the image is redundant or conveys no information choose the text alternative to be <blank>.

---

**Note.** Any alternate text that you select is visible to all users as mouse-over text on some browsers regardless of whether the system administrator makes the accessibility features available to users in PeopleTools Security.

---

To specify a label for an image push button or link associated with a record field:

1. Open the Page Field Properties dialog box for the control.
2. Select Image as the type.
3. Determine the RFT names:
  - a. If the RFT name is descriptive, set the label type to that RFT name.
  - b. If the RFT name is not adequate, edit the RFT name or write a message using the Message Catalog.

---

**Note.** Image buttons and links not associated with record fields should use messages exclusively.

---

The steps to specify a label for other types of images are the same as for image buttons and links, with the following differences:

- For dynamic images, you can choose to use RFT names or messages.
- For static images, you can only use messages or static text.

PeopleSoft recommends using the Message Catalog.

### ***Labels for Data-Entry Fields***

All edit boxes, check boxes, radio buttons, and long edit boxes should have labels near the entry field that are physically close to the associated control, either immediately above, or immediately to the left or immediately to the right (depending on the input or interaction element). The label should be the actual label for the field or the label from a hidden field, and so on.

Do not use a second independent field as the label for an edit field. This causes the field that is serving as a label and the edit field to be disassociated by a screen reader, and it is confusing to blind users. It also results in a page that contains a field without a label and a label without a field.








### ***Labels for Grid Columns***

All grid columns need a label. The only grid columns that do not have labels include those that contain buttons or links.

## Using Color to Convey Meaning

Do not use color as your only way to convey information. Include a redundant clue so that users who are blind or color-blind are aware that there is something special or different about a field or process. A common misuse of the color red is to change the background or text color to indicate an error. Another common mistake is to use an icon that changes color based on status but is otherwise the same. Color-blind users cannot distinguish between certain colors, nor can screen readers. In addition, if a user prints a page on a black and white printer, color-dependent items on the page become indistinguishable.

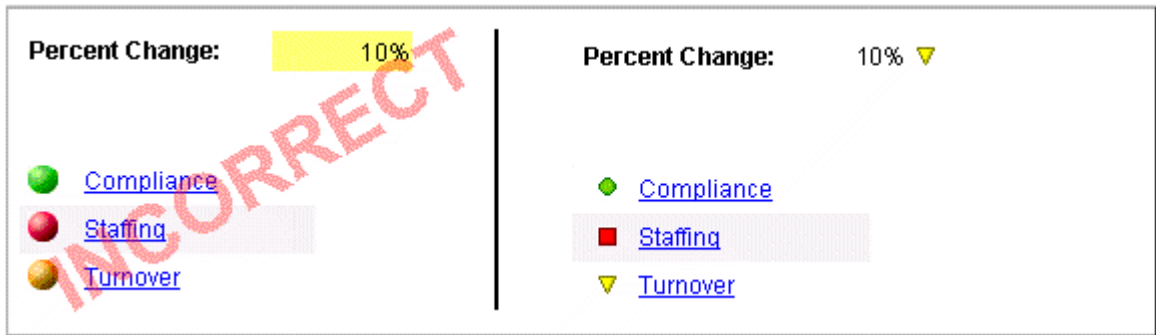
Use the approved set of icons in the table and see the following example on how to show status. All icons are GIFs or JPEGs.

	PS_STATUS_OK_ICN: OK, normal, good
	PS_STATUS_CAUTION_ICN: Warning, caution, at risk
	PS_STATUS_ERROR_ICN: Error, critical
	PS_STATUS_CLOSED_ICN: Completed
	PS_STATUS_CANCELLED_ICN: Cancelled.
	PS_STATUS_TREND_STABLE_ICN: Trend unchanged.
	PS_STATUS_REQUIRED_ICN: Required.

Use a text field that describes the status. For example:

Status: Error

Use icons of different shape *and* color to represent each status. The icons should also have descriptive mouse-over text so that screen readers for blind users can interpret the meaning of the icons.



Using icons of different shapes and colors to denote status

---

**Note.** Use more than color alone to indicate important information. Avoid flickering, moving, blinking, scrolling, or auto-updating objects or pages.

---

## Verifying Titles for Grids and Scroll Areas

All grids and scroll areas need to have meaningful titles. Sometimes titles are not used due to redundancy because the label repeats the name of the page. However, while the grid may be visually close in proximity to a page title, it may not be close when read aloud by a screen reader. This may cause a loss of association between the page and the grid, and the contents of the grid may become ambiguous.

Use the summary attribute to give a brief overview of the structure of the grid. Use the header and footer elements to add information for the grid or scroll area.

## Adding Unique Labels for Push Buttons and Links

Buttons and links need unique names on a page. While you might visually associate a link with an area, the association might not be apparent to a screen reader. Also, screen readers have the ability to list all links on the page. If there are two links with the same label, no distinction can be made to the user.

## Deactivating Adornment Images

Adornment images serve no functional purpose other than to provide visual interest. If the images have no information to convey to users and do not serve as navigation, then they should not have a label. Images used as spacers, such as invisible GIFs, should not have labels.

To deactivate the label for an image:

1. Access the Image Properties dialog box.
2. Select the Label tab.
3. Set Label Text Type to *Static Text*.
4. Delete any characters from the Text field, if it is present.

It is not necessary to enter a space. Leave the Text field blank.

## Maintaining Logical Tab Order

Maintaining a logical tab order on your page is critical because blind users cannot see the groupings or field proximity. Tab order is also important for sighted users because it can be confusing if the cursor moves to fields in an order different from how they appear on the screen.

## Using Deferred Processing

Every time there is a trip to the server, your current page is refreshed. This "refresh" can make the screen reader lose focus of the field that the user was on, and it will start reading the page from the beginning. This includes navigation links and all other items on the page. It is difficult for users using a screen reader to find their place as well. Using deferred processing enables you to determine when the transmission to the server needs to be made.

## Using Foreground and Background Colors

Text and background colors must have sufficient contrast to be viewed by someone with low vision or color-blindness. The PeopleSoft standard corporate style sheet is designed with this in mind. You might consider using the provided style sheets instead of custom ones.

If the information is conveyed in background image or color that information should be available to the screen reader through some other avenue.

## Using Instructional Text

Instructional text provides a brief overview of the page and a summary of the page contents for blind users. By providing instructional text, you save users from having to hear the contents of an entire page read aloud by the screen reader. All self-service pages should contain at least a minimum of one or two lines of instructional text.

When writing instructions, labels, or any other text for transaction pages, follow these general guidelines:

- Know your audience and write to its level of expertise.
- Do not make assumptions based on your expertise.
- Avoid highly specialized technical slang that is unique to an occupational group.
- Understand the task that you are describing.
- Write clearly and concisely.
- Use the fewest words required for clarity.
- Use short sentences; avoid paragraphs.
- Choose precise, imperative verbs.

For example, use "Complete all fields" instead of "Please fill out all the fields below."

- Do not construct sentences that merge variable information into a sentence, as these sentences are difficult to translate.

Use "OK to delete John Smith?" instead of "OK to delete dependent John Smith?"

- Correctly reference graphical user interface (GUI) elements.

- You click buttons, you do not click on a button.
- You select options, check boxes, and items in list boxes.

You do not choose them.

- You press keyboard keys.
- You clear options and check boxes (not uncheck, unmark, deselect, or other such variation).

You can cancel a selection as well.

- You delete items (not remove them).
- Use written English, not verbal English.  
(The English that you speak is not necessarily the English that you write.)
- Avoid contractions:

<b><i>Poor</i></b>	<b><i>Better</i></b>
Won't	Will not
Didn't	Did not
Shouldn't	Should not
Can't	Cannot

- Use short, simple, nonthreatening words:

<b><i>Poor</i></b>	<b><i>Better</i></b>
Abort	Cancel, Stop
Execute	Complete, Done
Implement	Do, Use, Put Into
Invalid	Not Correct, Not Good, Not Valid
Terminate	End, Exit

- Avoid hyphenation.

- Eliminate unnecessary words:

<b>Poor</b>	<b>Better</b>
in order to	to
so that	so
in regards to	regarding

### **Text for Labels**

When writing labels, use the following guidelines:

- Write short, concise, yet descriptive labels.
- Do not capitalize all letters in the words included in your label; use sentence case.
- Use abbreviations only when there is limited space.
- Follow field labels with a colon (:).
- Do not use repetitive words.

Use "Copy the overall definition for the following: Rules, Rates, Offerings" instead of "Copy the overall definition for the following: Rules definition, Rates definition, and Offerings definition."

---

**Note.** When using ampersands (&) in field labels (and hyperlinks) consider that PeopleSoft determines if a field has an access key based on whether or not the field label contains an ampersand (&). If the field label contains any ampersands, the last ampersand in the label along with the text following the last ampersand is used to create the text for the HTML and the system removes the last ampersand.

If the label contains a single ampersand (&), and the following character is *not* a blank space or another ampersand, the character following the ampersand will be used as an access key. Use a double ampersand (&&) so that a single '&' appears in the label as well as enabling the character immediately following the second ampersand as the access key.

For example, '&TEST' appears on the page as 'TEST' with the 'T' activated as an access key. '&&TEST' appears on the page as '&TEST' with the 'T' activated as an access key.

---

### **Text for Titles**

When writing text for titles, use the following guidelines:

- Write titles that are short and descriptive.
- Use verbs (for example, Create Expense Report).

- When writing titles that will appear in the grid or scroll area navigation bars, do not repeat the transaction title or the page title.

Instead, use a similar but more descriptive phrase or a slightly longer description. For example, the title in the blue bar repeats the page title but adds more description:

## Direct Deposit

Antonio Santos

### Direct Deposit Detail

Account Type	Transit Number	Account Number	Deposit Type	Amt/Pct		
<a href="#">Checking</a>	235842	224673232	Percent	50%	<a href="#">Edit</a>	<a href="#">Delete</a>

[Add Account](#)

Grid navigation bar title text

### ***General Guidelines for Instructional Text***

When writing instructions, use the following guidelines:

- Do not use quotations around reference words.
- To reference a word, capitalize the first letter of the word.

In the following example, see the instructional text with a reference to an object (Primary Phone checkbox):

## Phone Numbers

Antonio Santos

Enter your phone numbers below and indicate your primary contact phone number by checking the Primary Phone checkbox.

*Phone Type	Telephone	Primary Phone	
<input type="text" value="Business"/>	<input type="text" value="805/222-3333"/>	<input checked="" type="checkbox"/>	<a href="#">Delete</a>

[Add a Phone Number](#)

[Save](#)

\* Required Field

Instructional text example

- Use short sentences; avoid paragraphs.

Use "The listed competencies are based on your job. Complete the information where applicable. To enter additional competencies, click Add. When completed, click Save" instead of "The following competencies listed have been populated based on your job. Please complete the information below (where applicable) for your perceived proficiency, your interest in developing this competency further, year acquired, year last used, and years of experience. If you need to add additional competencies, please click the Add button below."

- Use active voice.

Use "Enter the part numbers" instead of "The part numbers should be entered."

### ***Page Instructional Text***

Page instructional text provides users with instructions on how to complete a task on the page. Use this text when pages have unique requirements or are particularly complex. When writing page instructional text, follow these guidelines:

- Place instructional text after level 0 identifying data.
- Set the style of the instructional text to PAPAGEINSTRUCTIONS.
- Derive instructional text from the Message Catalog.

Use a unique message set for each product.

- Use Static Text to provide instructional text for a page.
- When you need to create a dynamic phrase, embed bind variables in a message catalog entry.

### ***Additional Instructional Text***

Additional instructional text is used within a page and it helps users understand a specific section or field on the page. Additional text can also be used as a footnote at the bottom of a section or page. Use this text when a page or sections of the page have unique requirements or are particularly complex.

## Chapter 9

# Using Page Controls

This chapter provides an overview of page controls and discusses how to use:

- Charts.
- Frames.
- Group boxes.
- Horizontal rules.
- Images.
- Static text.
- Check boxes.
- Drop-down list boxes.
- Edit boxes and long edit boxes.
- Radio buttons.
- Popup pages.
- Subpages.
- Secondary pages.
- Push buttons or links.
- HTML areas.

---

## Understanding Page Controls

There are three categories of controls that you use in page design and development. You determine which types of controls that you want to add to your page definition by considering how you want to organize information on a page as well as how your users will enter data.

This section discusses:

- Aesthetic controls.
- Data entry controls.
- Function and data processing controls.























































































































































1. Click the Properties button located in the Summary group box.
2. Clear the Default to Title check box to activate the other Summary options.
3. Enter static text or enter a Message Set and Number to retrieve information from the message catalog.
4. Click the OK button.
5. Save the page.

## Setting Grid Use Properties

On the Use tab in the Grid Properties dialog box, you control data entry, row style, and display options, as well as whether your grid has a pop-up menu in the navigation bar at runtime. Both the grid and scroll area share many of the same attributes. Refer to the proceeding term tables for information about these shared settings.











<b>Odd/Even Row Style</b>	Selected by default, this attribute displays the grid with contrasting colors by row. The Odd Row Style and Even Row Style drop-down list box options appear instead of the single Row Style option in the Row Styles group box.
<b>Show Column When Cells Hidden</b>	Select if you want the columns to appear even if all of the cells in a column are hidden. Columns do not collapse.
<b>Fixed Height (number of rows)</b>	Select to set the grid to a height corresponding to the number of rows in the occurs count attribute. The grid remains fixed at that height even if the actual number of data rows varies from that number. If the fixed height property is not used, the size of the grid dynamically changes based on the occurs count and the number of data rows retrieved.
<b>Enable View All Columns</b>	Displays an expand all button to the right of a tabbed grid, enabling the user to view all columns of the grid by scrolling to the right using the browser's horizontal scroll bar instead of selecting the grid tabs. The show grid tab button appears when columns are expanded to enable the user to view the tabs again.
<b>Disable Display in Modal Window</b>	Select to disable the zoom capability of the grid.
<b>Collapsible Data Area</b>	Select to collapse the data area for your grid or scroll area into a header bar with a small image that the user must click to activate or expand it. This enables the Default Initial View to Expanded State check box. You can then select the label image to represent the collapsed and expanded states.

▼ National ID					View All	First ◀ 1 of 1 ▶ Last
Country	*National ID	Type	Description	National ID	Primary ID	
USA	PR		Social Security Number	434-55-6666	<input checked="" type="checkbox"/>	 

Expanded grid

▶ National ID
---------------

Collapsed grid

## Chapter 11

# Creating Page Definitions

Pages are the graphical interface between your users and your application database. As a system designer, you configure or build pages that meet the data requirements of the application and that are easy to use and understand. Using PeopleSoft Application Designer, you can create, modify, and delete page definitions in your PeopleSoft system.

This chapter provides an overview of page design and development and discusses how to:

- Use page development tools.
- Create new page definitions.
- Add page controls.
- Manipulate page controls.
- Set page field properties for controls.
- Create display control and related fields.
- Order and space page controls.
- Design inquiry pages.
- Align page controls.
- Maximize performance.
- Access PeopleCode in page definitions.
- View pages in the browser.
- Produce pages.
- Work with pages in foreign languages.

---

## Understanding Page Design

This section discusses:

- Level-based controls.
- Keys for accessing pages.
- Multiple occurrences of data.

- Prompt fields.
- Derived work fields.
- Sensitive data.
- Hidden pages.
- Upgrade considerations.
- End user page customization.

## Level-Based Controls

Page design hinges on the type of data that you plan to access and maintain. In some cases, a page references a single record definition; in others, you may want to reference multiple records. To accommodate a variety of page designs, PeopleSoft applications use *level-based controls*. The three level-based controls are:

- Grids
- Scroll areas
- Scroll bars

There are four levels in level-based controls: Level 0, Level 1, Level 2, and Level 3. These levels are referenced as *occurs levels* on the Record tab in the page field properties for the level-based control that you are setting.

- Level 0

This is the nonscrolling area that directly relates to the key information of the underlying record. Level 0 information is usually display-only with data that the user entered on the initial search page.

- Levels 1–3

These levels include the scrolling data that is related to the Level 0, nonscrolling data. Level 1 is subordinate to Level 0; Level 2 is subordinate to and nested in Level 1; and Level 3 is subordinate to and nested in Level 2. You can nest level-based controls up to 3 levels.

The first occurs level on a page is Level 0. In general, this level is reserved for the primary key fields that are used to search for pages. It is possible to have a page that contains no level-based controls, making all fields set to Level 0. This is particularly true for secondary or subpages that contain few data entry fields, as shown in this example. The second column, Lvl (level), of the Order tab indicates that all fields on the page are at Level 0.

	ID	Lvl	Label	Type	Field	Record	Display Control
1	1	0	Subpage			NOSEARCH_VW	<input checked="" type="checkbox"/>
2	2	0	Activity Name	Group Box	ACTIVITYNAME	PSACTIVITYGUI	<input type="checkbox"/>
3	3	0	Step 1 Link	Push Btn/Link	STEP_1_URL	PSACTIVITYGUI	<input type="checkbox"/>
4	4	0	Image 1	Image	STEP_2_IMAGE	PSACTIVITYGUI	<input type="checkbox"/>
5	5	0	Image 1	Image	STEP_1_IMAGE	PSACTIVITYGUI	<input type="checkbox"/>
6	6	0	Step 2 Link	Push Btn/Link	STEP_2_URL	PSACTIVITYGUI	<input type="checkbox"/>
7	7	0	Image 1	Image	STEP_3_IMAGE	PSACTIVITYGUI	<input type="checkbox"/>

### Order tab

After you add a level-based control to your page, such as a scroll area, the default occurs level for that control is set to one in the properties dialog box. If you place a field in or below that scroll area, it is also set to Level 1 in the Order tab, even if it is another level-based control. If necessary, you can use the Set to Level 0 feature of the horizontal rule control to restart the occurs level on your page.

Note that you can have any number of level-based controls at the same level.

### ***Nested Level-Based Controls***

Nesting controls involves two or more level-based controls on a page, such as two scroll areas, when the second scroll area has an occurs level of 2. You nest controls when the new data that you want to add is a repeating set of data for each entry in your first level-based control. In doing so, you create a hierarchical, or "parent and child" relationship between the controls and the processing of the record definitions. The Level 2 control is the child of, and is subordinate to, the Level 1 control.

For example, on the Compensation page, the Level 2 grid is nested in the Level 1 scroll area. On the Amounts tab of the grid, there are two rate codes for each compensation action in the Level 1 scroll area.

Work Location

Job Information

Job Labor

Payroll

Salary Plan

Compensation

Albion,Charles

Employee

ID: Z1000

Empl Rcd#: 0

Compensation

View All

First

1 of 1

Last

Effective Date:

12/01/1997

Effective Sequence:

0

Job Indicator:

Primary Job

Action / Reason:

Hire

Current

Compensation Rate:

6,176.78

USD

\*Compensation Frequency:

M

Monthly

Change Amount:

Change Percent:

Compa-Ratio:

Annual Benefits Base Rate:

Pay Rates

Hourly Rate:

35.64

USD

Daily Rate:

285.08

USD

Monthly Rate:

6,176.78

USD

Annual Rate:

74,121.39

USD

Pay Components

First

1-2 of 2

Last

Amounts

Changes

*Rate Code	Seq	Details	Comp Rate	Currency	Frequency	Points	Percent	Rate Code Group
1	NAANNL	0	Details	6000.000000	USD	M		
2	BLMAL1	0	Details	1316.000000	BEF	W		

Job Data

Employment Data

Earnings Distribution

Benefits Program Participation

Save

Return to Search

Next in List

Previous in List

Previous tab

Next tab

Update/Display

Include History

Correct History

Work Location

Job Information

Job Labor

Payroll

Salary Plan

Compensation

Compensation page

You can nest up to three levels of scrolls or grids on your page. For example, the Additional Pay 1 page shows three levels of data that can be entered, two of which are nesting scroll areas in the Earnings scroll area.

Additional Pay 1 | Additional Pay 2 | Additional Pay 3

Albion, Charles      Employee      ID: Z1000      Empl Rcd#: 0

**Earnings**      Find | View All      First 1 of 1 Last

\*Code:

**Eff Date and Default Job Data**      Find | View All      First 1 of 1 Last

Effective Date:

Compensation Rate/Frequency: \$6,000.000000 Monthly

Standard Hours: 40.00      Employee Type: Salaried

**Or Overrides to Job Data**      Find | View All      First 1 of 1 Last

\*Addl Seq #:

Earnings End Date:

Rate Code:

Reason:

Earnings:       Hours:       Hourly Rate:

Goal Amt:       Goal Bal:

Sep Chk #: ☐      ☐ Disable Direct Deposit      ☐ Prorate Additional Pay      ☐ OK to Pay

Applies to Pay Periods: ☒ First      ☐ Second      ☐ Third      ☐ Fourth      ☐ Fifth

[Additional Pay 1](#) | [Additional Pay 2](#) | [Additional Pay 3](#)

### Additional Pay 1 page

In this case, we want to maintain information about the compensation history for a single employee, and, for each change or override in compensation status, enter the details about that change. The first scroll area, Earnings, associates the earning code with the second, subordinate scroll area, Eff Date (effective date) and Default Job Data, which enables the user to enter the effective date for the status change. The third scroll area, Or Overrides to Job Data, which is subordinate to the second, enables the user to enter the overrides to the default job data that appears in the second scroll area.

For each of the three scroll areas, you can have several rows of data. While you don't see actual scroll bars on your page, the navigation buttons and links in the navigation bars for each scroll area enable you to navigate or scroll through the rows.

### Nested Grids in Scrolls

As shown previously, you can nest a grid in a scroll or scroll area. The following page shows the Other Earnings grid at Level 3, which is nested in the Earnings scroll area at Level 2, which is nested in the Paysheet Details scroll area at Level 1. A nested grid serves the same function as a nested scroll area. It can offer a more compact way of viewing many fields of data, particularly if you use a tabbed grid.

---

**Note.** While you can nest a grid in a scroll area, you cannot nest a scroll area in a grid or a grid in another grid.

---



The search record that you define for a component determines the key list; that is, search keys for which users are prompted when they select a page and an action. A search record can be either a view that concatenates information stored in several tables or the underlying table itself. Select the search record that contains all of the key items for the primary record underlying the page.

The keys for which the search dialog box prompts should populate the high-level (Level 0 in the page definition) key controls on a page. These key controls always appear above any level-based control on the page and are typically display-only. A search record might differ from a primary record definition, but it must contain all of the Level 0 keys that you have placed on the page.

### ***Using Keys for Views***

When you create a view that should use key values from a page, each of those keys must be:

- Listed as output columns from the view (not just referenced in the WHERE clause of the view).
- Marked as key fields of the view.

For example, suppose that a page has PERSON.PERSONID as a Level 0 field. PERSONID is the key of the PERSON record. A grid on the page uses the PERSON\_ADDRESS\_VW view to display addresses for the current person. The SQL statement for the view cannot just mention PERSON.PERSONID in the WHERE clause; PERSONID must be both an output column of the view and a key field of the view.

### ***See Also***

[Chapter 6, "Creating Record Definitions," page 113](#)

[Chapter 12, "Creating Component Definitions," page 361](#)

## **Multiple Occurrences of Data**

On some pages, you may want a few of the field controls to display multiple rows or occurrences of data. To do this, you can add a level-based control—a scroll area, a grid, or a scroll bar. Users can then add, edit, delete, find, and scroll through multiple occurrences of data in a page control or group of controls using action buttons, links, or the browser's scroll bar, depending on how you set the occurs count. Using a scroll area or a grid, rather than a scroll bar, is the preferred page design to show multiple occurrences of data.

For example, this Checklist Table page contains the Assignment Checklist Item grid with an occurs count of 5. The Checklist Item scroll area in which it is nested must have an occurs count of 1 since we can see only one row of data in the scroll area.

Checklist Table

Checklist Code: K00001

**Checklist Item** [View All](#) First 1 of 2 Last

\*Effective Date: 07/27/2000 \*Status: Active Checklist Type: Transfer + -

\*Description: Foreign Loan Departure Chcklst Short Description: FLA Depart

**Assignment Checklist Item** [View All](#) First 1-5 of 10 Last

*Sequence	*Item Code			
1	100	K00014	Briefing with Human Resources	+ -
2	200	K00029	Apply for Visas/Work Permits	+ -
3	300	K00008	Reconfirm Relocation Package	+ -
4	400	K00001	Select moving/storage company	+ -
5	500	K00010	Confirm move date with movers	+ -

[Save](#)
[Return to Search](#)
[Next in List](#)
[Previous in List](#)
[Add](#)
[Update/Display](#)
[Include History](#)
[Correct History](#)

### Checklist Table page

To see the next set of five rows in the Assignment Checklist Item grid, a user can click the right arrow button. To see all rows of data at once, in either the grid or scroll area, the user can select the View All link in the navigation bar.

In PeopleSoft Application Designer, you set the occurs count for a level-based control on the General tab in the properties dialog box. You can set the occurs count to any number. You also have the option of selecting Unlimited Occurs Count so that the user sees all rows of data. If there are more options than can be seen in the window at runtime, the browser displays a scroll bar enabling the user to scroll to the bottom of the page.

**Note.** When occurs count is set to a number other than 1, the grid displays the specified number of rows. However, when it is set to 1, the grid displays as many rows as possible within the given space allocated for the grid control at design time.

## Prompt Fields

At runtime, you may want to enable your users to look up the valid values that they can enter in a field. For this, PeopleSoft provides prompts or look up buttons. There are three types of prompts:

- Drop-down list box.

This is a small list that opens below a field in the current page, such as the Job Indicator field on the Work Location page. To use a drop-down list box, the user clicks the down arrow inside the field. One does not have to exit the page. The drop-down list box is a predefined control that you place on your page.

- Calendar drop-down prompt.

A calendar drop-down prompt opens a small calendar next to a date field to enable the user to easily scan for the correct date. The Work Location page contains calendar prompts for the Effective Date, Position Entry Date, and Department Entry Date fields. Place a calendar prompt on a page by associating any date field with an edit box control and selecting the Show Prompt Button in the Display Options of the Record properties of the edit box.

---

**Note.** The calendar prompt feature is not supported on all browsers.

---

See "Troubleshooting Browser Limitations" on Customer Connection.

- Prompt button.

A prompt or lookup button opens a lookup page in the user's browser populated with up to 300 available values for that field. The user can then either select the desired value or refine their search further. For extremely large tables, the system administrator has the option of excluding that table from auto prompting on the Lookup Exclusion Table via PeopleTools, Utilities, Administration.

In PeopleSoft Application Designer, you can only associate a prompt button with an edit box control. Do this by selecting the Show Prompt Button check box from the Display Options on the Record tab of the Edit Box Properties. The record field with which you associate the edit box must list values in the Translate Table or prompt table for the system to display the prompt button.

Work Location   Job Information   Job Labor   Payroll   Salary Plan   Compensation

Smith, John   Employee   ID: G050   Empl Rcd#: 0

**Work Location**   View All   First 1 of 1 Last

Employee Status: Active   Date Created: 03/23/1998

\*Effective Date: 06/11/1998   Effective Sequence: 0   \*Job Indicator: Primary Job

Action / Reason: Hire

Position Number:   Position Date:   Position Entry Date:   Position Record

\*Regulatory Region: USA   \*Company: CCB   \*Business Unit: USSVC   Department Entry Date: 06/11/1998

\*Department: 10200   Location: 001   Supervisor ID:   Department

Supervisor ID:   Department Entry Date: 06/11/1998

France

Job Data   Employment Data   Earnings Distribution   Benefits Program Participation

Save   Return to Search   Next in List   Previous in List   Previous tab   Next tab   Update/Display   Include History   Correct History

[Work Location](#) | [Job Information](#) | [Job Labor](#) | [Payroll](#) | [Salary Plan](#) | [Compensation](#)

Work Location page

## See Also

[Chapter 9, "Using Page Controls," Using Drop-Down List Boxes, page 221](#)

[Chapter 4, "Creating Field Definitions," Specifying Date Field Attributes, page 69](#)

[Chapter 4, "Creating Field Definitions," Using the Translate Table, page 89](#)

## Derived and Work Fields

You can use a field definition from a derived or work record to store a temporary value that PeopleCode uses to determine the values of other field controls on the page.

For example, for a budgeting transaction in PeopleSoft General Ledger, assume that you have an annual amount that you must spread to multiple accounting periods. You can create a page that includes both a field control from a derived record for the annual amount and an amount control for each accounting period. You then write PeopleCode to derive the amount per accounting period from the annual amount. When a user enters the annual amount in the derived field control, PeopleCode calculates the amount per accounting period and inserts it into each accounting period field control. The annual amount isn't stored in the database, but the period amounts are.

## Sensitive Data

You can allocate sensitive data to a single page and limit access to that page to the users who need to update it. Alternatively, you can enter PeopleCode to hide certain fields on pages based on whatever criteria are appropriate. The page approach is simpler, however, and can be used in most situations. For secondary pages, this sort of security is more complex to manage.

### See Also

*Enterprise PeopleTools 8.50 PeopleBook: PeopleCode Language Reference*, "PeopleCode Built-in Functions," Hide

## Hidden Pages

Hidden pages are work pages that are associated with derived or work records and are often used in work groups. You can store all of your work field controls there. Create these pages when you want calculations to be performed in the background by PeopleCode that the user does not need to see. As a convention, PeopleSoft assigns work pages that are delivered with your application names that end with the suffix, `_WRK`. PeopleSoft recommends that you follow the same naming convention. For example, `MC_TYPE_WRK`.

Hide a page by selecting the Hidden check box in the component grid as you set up the component definition.

### See Also

[Chapter 12, "Creating Component Definitions," Setting Page Attributes, page 367](#)

## Upgrade Considerations

If you adapt a PeopleSoft application, you might affect your use of future PeopleSoft releases. The closer your PeopleSoft system is to the standard product, the easier your upgrades will be. Avoid superfluous or cosmetic changes to data structures in the standard product. Document all of your changes in the comment area on the General tab in the Page Properties dialog box.

## End User Page Customization

End users of PeopleSoft applications can customize various settings on a page including tab order, the display state of scroll areas and grids, and the state of collapsible sections. Users can also choose the default page to display for a component each time they open that component. In addition, users can choose to share their customization settings with or copy settings from another user.

The page customization option is set in the Component Properties dialog box, Internet tab. The Customize Page Link check box is selected by default. This enables the Customize Page link in the pagebar at runtime for all pages.

## See Also

[Chapter 12, "Creating Component Definitions," Setting Internet Properties, page 374](#)

*Enterprise PeopleTools 8.50 PeopleBook: Using PeopleSoft Applications*, "Setting User Preferences," Customizing Pages

---

## Using Page Development Tools

This section discusses how to:

- Use page definition menus.
- Use page definition toolbars.
- Use the page layout grid.

## Using Page Definition Menus

As you build pages, use the tools and options in these menus.

- File Menu.

In addition to the standard options for opening, deleting, and renaming pages, use this menu to access the definition (page) properties and project properties.

- Edit Menu.

Access the page field properties from this menu. To find out what other definitions reference the active page, select Find Definition References. You can also search for a text string in types of PeopleCode or SQL objects.

- View Menu.

- Select View Definition to view the underlying definition, such as the subpage or record, for the selected control.

You can view the PeopleCode for the page or the underlying record definition.

- Edit Fill Display, Edit Fill All, and Edit Fill None refer to how controls appear only during design time.

They do not affect the page during runtime.

- Use the Show Page Control Inspector tool to view summary data of the selected page field.
- Use Show Grid to display or hide a layout grid on the page that you are designing to help you place controls more accurately.

- **Insert Menu.**

Use this menu to insert different types of controls on your page. You can also insert the page on which you are working into a project.

- **Tools Menu.**

This menu provides access to various utilities, such as data administration, change control, upgrade, and translate.

- **Layout Menu.**

Use this menu to modify the number of pixels between points on the layout grid. Options on this menu also enable you to align page controls, view the processing order of the controls on the page, and test the tab order between fields on your page.

- The View in Browser function enables you to see a very general image of the page in the selected browser without data. Note, however, that PeopleCode is not run on this page rendering, so the page may look significantly different for the user if PeopleCode is used to hide or unhide page fields.
- The Generate HTML function creates a text file containing the HTML for your page and places it in the TEMP directory of your local drive. Note that the HTML generated contains no data or context. It is therefore likely to differ from the HTML on a page at runtime.

## Using Toolbars

The toolbars in PeopleSoft Application Designer change, depending on the active window. When a page window is active, you see the page definition toolbar and the page control toolbar.

### ***Page Definition Toolbar***



Click the Properties button to open the Page Properties dialog box (Alt+Enter).



Click the Project Workspace button to display or hide the project workspace (View, Project Workspace or Alt+0).



Click the Select Group button to select several controls at once (Edit, Select Group menu item or Ctrl+P).



Click the Default Ordering button to reorder all of the fields on the page based on their relative locations on the page (Layout, Default Ordering).



Click the Test Page button to test the tab order of your page design (Layout, Test Mode or Ctrl+T).



Click the View Page in Browser button frequently throughout the design process (Layout, View in Browser).



Click the Toggle Inspector button to display or hide the floating Definition Inspector window (View, Show Page Control Inspector).



Click the Auto Size button to size a subpage or secondary page to fit around page controls.



Click the Toggle Grid button to display or hide the page grid (View, Show Grid or Ctrl+G).



Click the Left Label Position button to place page control label for the currently selected field immediately to the left (Layout, Left Label Position or Ctrl+D).



Click the various alignment buttons to align the edges of selected page fields relative to the last clicked field.

See [Chapter 11, "Creating Page Definitions," Aligning Page Controls, page 340.](#)



Click the Show Help button to view the page definition help.

### ***Page Control Toolbar***

The page control toolbar contains a button for each type of page control that you insert. The buttons display images that look like the controls that they insert. You can move this toolbar on your workspace independently of the system toolbar or the other half of the page definition toolbar.



Frame.



Group box.



Horizontal rule.



Static text.



Static image.



Grid tab separator.



Check box.



Drop-down list box.



Edit box.



HTML area.



Image.



Long edit box.



Push button/hyperlink.



Radio button.



Tree.



Grid.



Scroll area.



Scroll bar.



Secondary page.



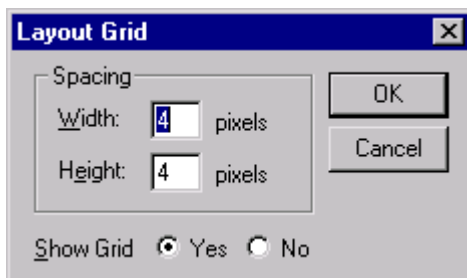
Subpage.



Chart.

## Using the Page Layout Grid

When you open a page definition, evenly spaced dots appear in the workspace. This is the page layout grid that you use to align controls. When you're running applications, the grid doesn't appear on pages. It appears only as a design aid in the page definition.



Layout grid

The default grid spacing is  $4 \times 4$ , which means that each row has one dot per 4 pixels and that the rows are 4 pixels apart on the vertical axis. PeopleSoft builds most of the pages in the standard applications using this grid. To change the spacing, open the Layout Grid dialog box by selecting Grid Settings from the Layout menu.

You can turn off the grid view of your page by clicking the Toggle Grid button in the page definition toolbar.

### **Changing Page Layout Grid Settings**

To change the grid spacing:

1. Select Layout , Grid Settings.

2. Enter values for the width and the height, which is the spacing of the dots in pixels.

The smaller the value, the finer the grid. The finer the grid, the more difficult it is to align fields precisely without using the align functionality. Note that if either the width or height are set to one, then the grid will be slightly off.

Select a Show Grid option to activate or deactivate the page grid.

If you copy an existing page and save it under another name, you also copy the grid settings that were used to create the original page. For example, if the original page was built with a  $5 \times 5$  grid, your new page also has a  $5 \times 5$  grid, by default. You can change the grid settings anytime while working on a page. Doing so has no effect on the position of a field.

The snap-to-grid functionality is only available when the layout grid is showing.

---

## Creating New Page Definitions

To create a new page definition, either clone an existing page or begin with a blank page. Cloning is more efficient, because you don't have to re-create controls that are common to all pages. You can also create a template that contains only the controls that you need.

This section discusses how to:

- Clone page definitions.
- Create a blank page.

---

**Note.** When saving new definitions, always make sure to adhere to accepted naming conventions.

---

See [Chapter 2, "Using PeopleSoft Application Designer," Saving Definitions, page 31.](#)

## Cloning Page Definitions

To clone a page definition:

1. Select File, Open.

The Open Definition dialog box appears.

2. Specify *Page* for the definition and locate the page that you want to clone.
3. When the system retrieves the page definition, select File, Save As.

If you save after you make modifications, you might accidentally overwrite the old page definition.

4. Enter the new page name.

To create a new page template, name the page so that it appears at the top of your page list.

5. Click OK.

---

**Note.** Record definition names are restricted to a maximum of 15 characters, while page names can have up to 18 characters.

---

### **See Also**

[Chapter 2, "Using PeopleSoft Application Designer," Opening Definitions, page 28](#)

[Chapter 2, "Using PeopleSoft Application Designer," Saving Definitions, page 31](#)

## **Creating a Blank Page**

To create a new page without copying an existing page, select File, New, Page.

When saving the page, make sure that you adhere to the accepted naming conventions.

See [Chapter 2, "Using PeopleSoft Application Designer," Saving Definitions, page 31](#).

---

## **Adding Page Controls**

There are multiple ways to add page fields to a page. You can copy page fields from one page and paste them onto another, or you can drag-and-drop page fields from one page to another. In addition to these basic methods, this section discusses how to:

- Drag field definitions from a record onto a page.
- Drag record fields from the project workspace onto a page.
- Drag record definitions onto a page.
- Use the page control toolbar.
- Use the Insert menu.

## **Dragging Field Definitions From a Record Onto a Page**

To drag field definitions from a record onto a page:

1. Open an existing record definition by selecting File, Open, Record.
2. Open a new page by selecting File, New, Page.
3. Drag field definitions from the record to the page.

The system selects the page definition control type based on the record field attributes. In the preceding page, the EMPLID (employee identification) field was defined as a prompt table edit in the record, so a Prompt button was added for EmplID when it was dragged onto the page.

4. Double-click the new page field to set the properties.

When you drag field definitions, the properties from the record definition are used to set page field default properties. You can alter the properties using the Properties dialog box.

## Dragging Record Fields From the Project Workspace Onto a Page

To drag record fields from the project workspace onto a page:

1. Open the project that you want to drag from an existing or new page.
2. Drag record fields from the project workspace to the page.
3. Double-click the new page field to set the properties.

---

**Note.** When you drag fields from the project workspace, use fields that are contained in records (as in record.fieldname), not field definitions.

---

## Dragging Record Definitions Onto a Page

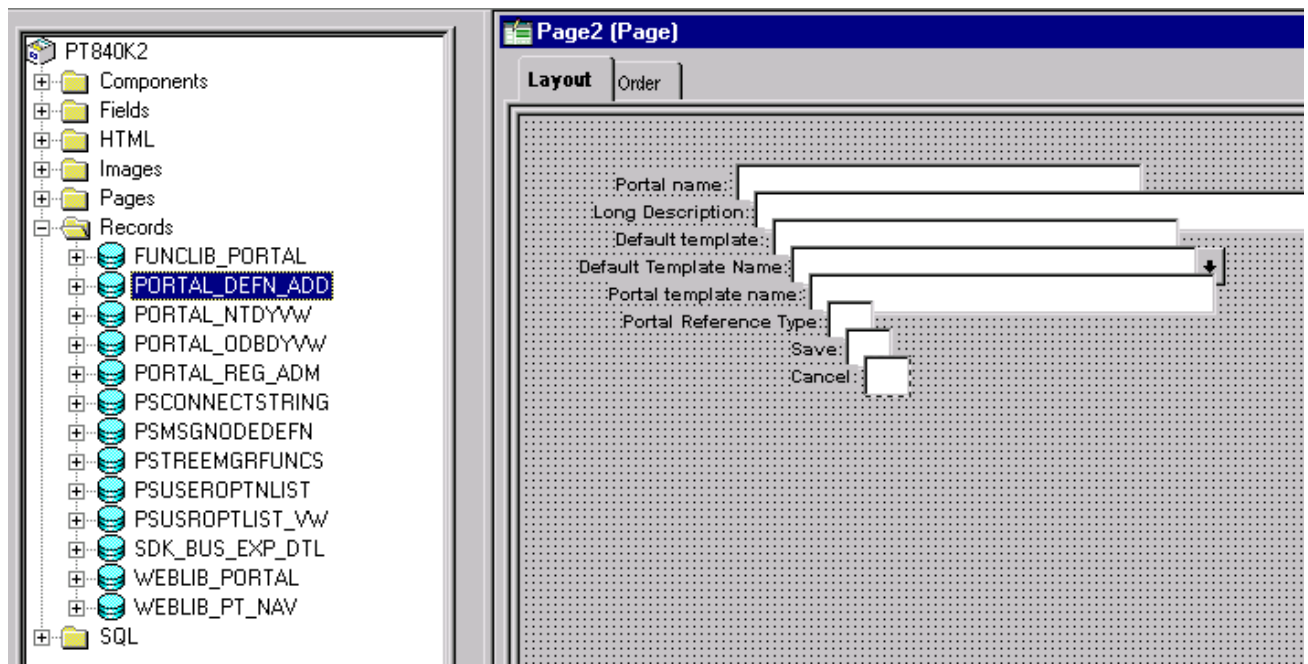
You can add an entire record definition to a page at once. This can save a great deal of time when generating pages. You can then arrange the fields as desired on the page.

---

**Note.** This can also be done for a grid. You can drag an entire record definition into a grid control that you've placed on a page.

---

See [Chapter 10, "Using Scroll Areas, Scroll Bars, and Grids," Inserting and Manipulating Grid Columns, page 269.](#)



Dragging a record definition to a page

To drag record definitions onto a page:

1. Open the project that you want to drag from an existing or new page.
2. Drag the record definition onto the page.

PeopleSoft Application Designer positions the record fields on the page as page fields in a cascading manner. Fields are in the order that they appear in the record with the appropriate control value assigned to each one.

3. Reposition the fields as necessary on your page.
4. Double-click each new page field to set the properties.

## Using the Page Control Toolbar

You can insert page controls by using the page control toolbar. When you click one of the page control buttons, the cursor changes to a cross-shaped icon or hand, depending on which control type you select. You can then move the cursor to where you want to position the control and click once. If you select a control from the toolbar erroneously and want to cancel the selection, press the Esc key, and then select the desired page control button.

If the control is a fixed size during insertion, the cursor changes to a cross-shaped icon. If the control type requires that you define the size of the control at the time of insertion, the cursor changes to the monogrammed hand icon. When a hand-shaped icon appears, press and hold the left mouse button as you drag the hand diagonally downward to where you want the lower, right-hand corner of the frame. Release the mouse button.

If the control isn't correct the first time that you draw it, you can easily adjust it. The dotted box and black or blue handles surrounding the control indicate that it is selected. Use any of the four directional arrow keys on the keyboard to move the frame one page grid unit in the indicated direction. You can also use Shift+ the arrow key to resize the page control.

## Using the Insert Menu

Select the control that you want to add to your page from the Insert menu. Your cursor changes to a cross-shaped icon or a hand, depending on which control type you select. Move the cursor to where you want to position the control and click once.

---

## Manipulating Page Controls

After you place a control on the page, you have several choices for manipulating it.

This section discusses how to:

- Select controls.
- Resize controls.
- Use the Page Control Inspector.

- Delete controls.
- Move controls on the same page.
- Copy and move controls to another page.
- Work with field labels.

## Selecting Controls

Before you do anything to a control, you must select it. You can do this by clicking the control or by using the Edit menu. A check mark beside the option in the menu indicates which of these modes you're in:

**Select Field** Select one control on the page by clicking on it. To deselect a control, click once outside the dotted box. This is the default mode.

**Select Group** Select a group of controls by drawing a box around the controls that you want to select. You can also activate this option by clicking and dragging over the target fields.

---

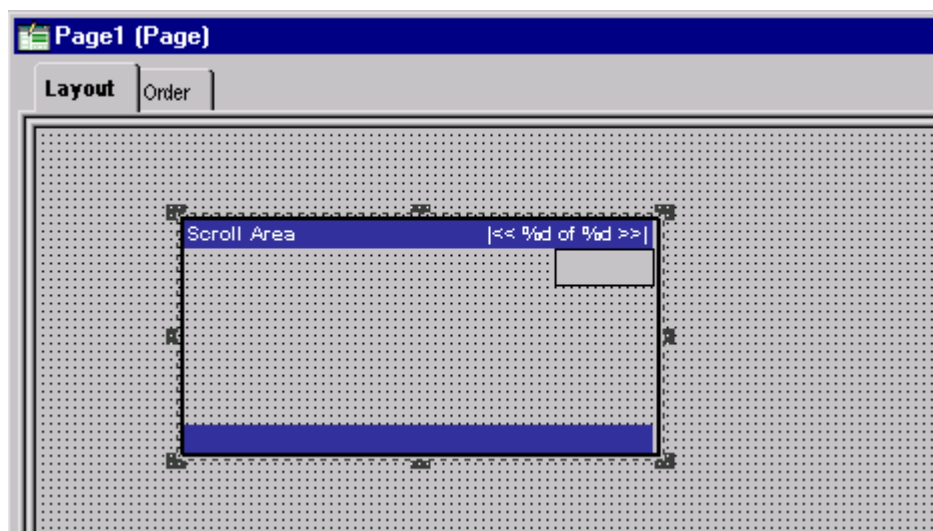
**Note.** You are in Select Group mode when the cursor changes to a pointing finger. If the cursor does not change when you select Edit, Select Group, try first selecting an individual control on the page and then selecting Edit, Select Group again.

---

**Select All** Select all of the controls on the active page. The result is the same as using the Select Group option to draw a box around the controls.

## Resizing Controls

You can adjust the size or shape of any page control that displays selection handles when it is selected.



Resizing a control with selection handles







## Setting Record Properties

For most controls, you must set record properties to assign the control to a specific field in a record definition. There's a distinct correlation between the field types in a record definition and controls in a page definition.

To select the appropriate page control type to associate with the corresponding field in your record definition use this table:

<i>Record Definition Field Type</i>	<i>Page Definition Control Type</i>
Character	Edit box
Character (Yes or No Table Edit)	Check box Edit box
Character (Translate Table Edit)	Radio button Edit box Drop-down list box
Long Character	Long edit box, HTML area  <b>Note.</b> Do not associate Long Character fields to edit box controls; use a <i>long edit box</i> control. Associating a Long Character field to an edit box control can produce unexpected behavior on the page.
Number	Edit box
Signed Number	Edit box
Date	Edit box
Time	Edit box
DateTime	Edit box
Subrecord	Not applicable; no direct association to subrecords
Image	Image

To set the record properties for a page control, access the page field properties for the control and select the Record tab.









**Example: Auto Decimal**

This table shows how Auto Decimal affects a numeric field control with a length of 4.2, depending on what you enter online:

<i>Entered</i>	<i>Displayed</i>
100	1.00
100.	100.00
1	0.01

**Setting Label Properties**

The Label tab enables you to set the label that appears on the control. This can be the long or short name that is specified on the record definition or other text. Use this tab to verify the label ID that is used as the internal reference.

Access the page field properties for the control and select the Label tab.







<b><i>Control Label</i></b>	<b><i>Related Field Label (Information-Only)</i></b>
Regular/Temporary	Regular/Temporary XLAT (XLAT is from the Translate Table)

When defining labels for columnar controls in a multi-occurrence scroll, you can apply several techniques. The first technique is to select First occurs only. This limits the display of the label to the control's first scroll occurrence, allowing the label to be used as a column heading. The second technique is to select No colon to omit the colon from the label, because column headings typically do not need a colon.

## Setting Use Properties

The Use tab defines how to use the page field on the page. You can make fields display-only or invisible. Use this tab to define the display and related control fields and the processing of the control. Use this tab to define a field as a mouse over popup field.

Access the page field properties for the page control and select the Use tab.



<b>Invisible</b>	<p>Select to make the control physically present on a page but invisible to users. Typically, you add an invisible control because it's required for a PeopleCode program that is associated with the page. An invisible control can also be used as a display control field.</p> <p>When you select the Invisible check box, the Show Label, Modifiable by JavaScript, and the Enable When Page is Display Only fields are enabled.</p>
<b>Show Label</b>	<p>Select to make the control label visible while the control itself is invisible. This option is useful if you add an invisible display control field to a page to show its related description. Show Label is enabled only if Invisible is selected.</p>
<b>Modifiable by JavaScript</b>	<p>This is a security-related feature and should <i>always</i> be cleared unless you are familiar with modifying an invisible field using JavaScript in an HTML area. If the Invisible check box is cleared, Modifiable by JavaScript is cleared and unavailable for entry. If the Invisible check box is selected, this check box is cleared, by default.</p>
<b>Enable When Page is Display Only</b>	<p>Select to enable to perform hyperlink actions on read only pages. This check box is active when the Invisible check box is selected. This check box is cleared by default.</p> <hr/> <p><b>Note.</b> This property appears on Edit box, Long Edit box and Image controls.</p> <hr/>
<b>Expand Field when Page is Display Only</b>	<p>Select to make a long edit box expand at runtime to display all the text it contains when it is marked as Display Only. The field automatically expands to accommodate text. This check box is cleared, by default. When selected, this option enables the Wrap Long Words option.</p>
<b>Multi-Currency Field</b>	<p>Select to identify the control as associated with multicurrency processing. This causes the field to appear at runtime only if Multi-Currency is selected on the PeopleTools Options page.</p>
<b>Display Control Field</b>	<p>Select to indicate that the field controls another field on the same page level. The controlled field is a related field. For example, on many pages, Department ID is the display control field and Department Name is the related field. The system uses Department ID to retrieve Department Name from the Department table to display on the page.</p>







---

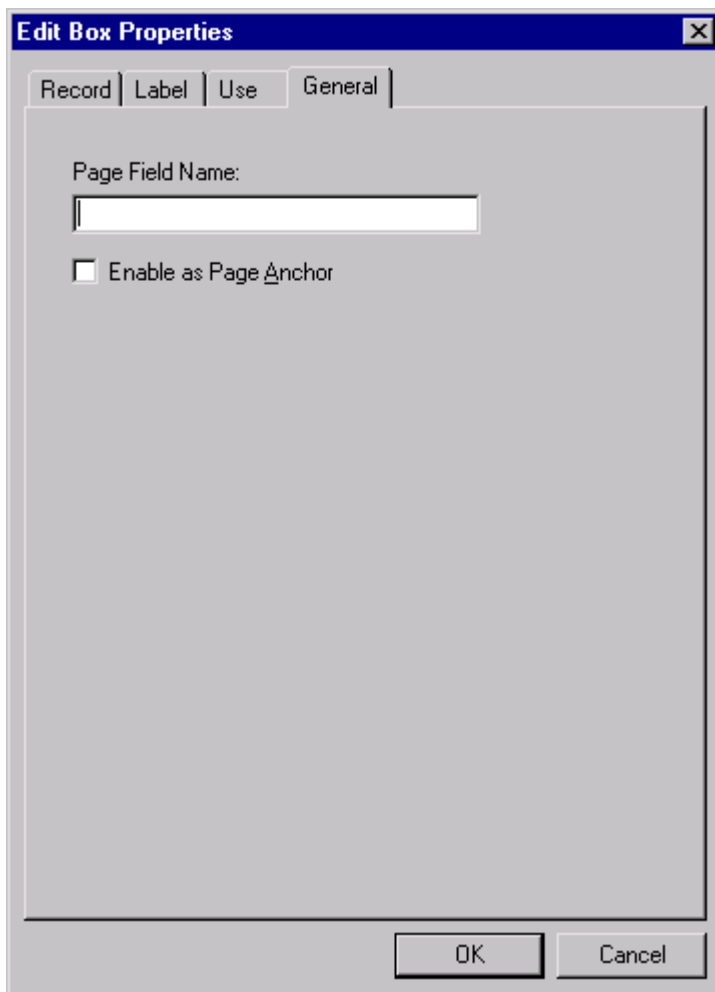
**Note.** Attempts to make an edit box editable at runtime when it has a mouse over popup page associated with it for example, by using PeopleCode, will fail.

---

## Setting General Properties

The General tab enables you to specify an optional internal page field name that is referenced by the page.

Access the page field properties for the control and select the General tab.



Edit Box Properties dialog box: General tab

---

**Note.** For most controls, the General tab contains only the Page Field Name and Enable as Page Anchor settings. Grids and scroll areas have additional settings on the General tab that are described later. The Enable as Page Anchor check box is available for all controls that have a page field name.

---

### Page Field Name

Enter a name for the page field that you are creating. This is a system-only setting. It does not appear as the label for the page field. The page field name enables you to refer to a particular field on a page using PeopleCode.

This setting is optional unless you select the Enable as Page Anchor check box.

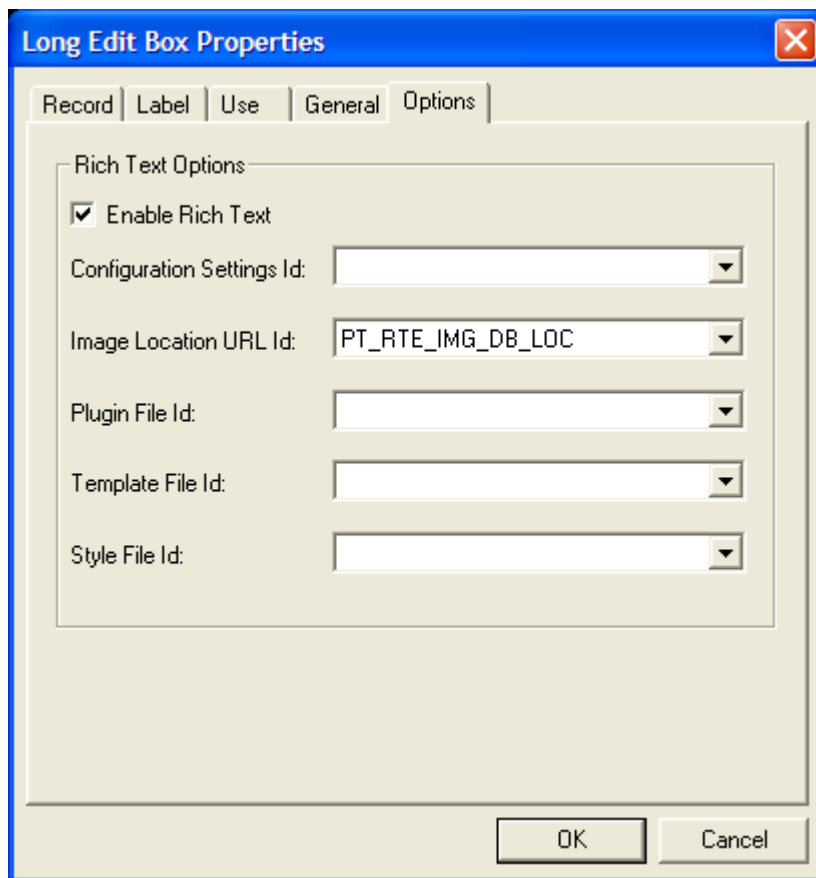
**Enable as Page Anchor** Select to apply an anchor tag to the current page field on the page. If you select this check box, you must add a page field name to identify this field when setting the related control for the link. This check box is cleared, by default.

See [Chapter 9, "Using Page Controls," Specifying Type Properties for Push Buttons or Links, page 245.](#)

## Setting Options Properties

The Options tab defines how to use the page field as a rich text editor on the page.

Access the page field properties for the long edit box page control and select the Options tab.



Long Edit Box Properties dialog box: Options tab

### Enable Rich Text

Select to enable rich text options for the long edit box. Selecting this check box enables the remaining fields on this tab.

---

**Note.** When you select this option the Multi-Currency and Wrap Long Words properties on the Use tab are disabled; and the Allow Deferred Processing property on the Use tab is selected and disabled.

---













## See Also

*Enterprise PeopleTools 8.50 PeopleBook: PeopleCode Language Reference*, "PeopleCode Built-in Functions," Hide

## Creating Invisible Control Fields

Your page design may require an invisible control field. For example, if users are interested only in the contents of the related display field, then make the control field invisible. Unlike visible control fields, you *can* overlap an invisible control field and its related display field.

---

**Note.** If you specify a field as invisible in PeopleSoft Application Designer, you cannot make it visible using the UnHide property in PeopleCode.

---

To add an invisible control field with a visible label:

1. Access the properties dialog box for the invisible control for which you want the label to appear.
2. Select the Use tab.
3. Set the appropriate field use options:
  - a. Select Display Only, Invisible, and Show Label to make the field invisible and its label visible.
  - b. Select Display Control Field.
4. Select OK.
5. Add an edit box to your page.

Place this edit box directly to the right of the display control field that you just set.
6. Access the properties dialog box for the new edit box.
7. Select the Use tab.
8. Select Related Field.
9. Select the appropriate control field from the Related Control Field drop-down list box.
10. Select OK.

---

## Ordering and Spacing Page Controls

This section provides an overview of page control order and spacing and discusses how to:

- Order controls visually.
- Order controls logically.
- Test page control order.

















- Generate HTML.
- View the HTML for a page.
- Identify page definitions online.

## Working With the View in Browser Feature

During the page design process, you should periodically view how your page looks in the browser. How you design your page in PeopleSoft Application Designer may look different when viewed online, particularly if you are using subpages that contain multiple fields.

To do this, select Layout, View in Browser. This feature is helpful if you are changing any of the style characteristics of the page or aligning and spacing controls, such as scroll areas.

**PeopleTools Options**

**Language Settings**

Base Language Code:  ☐ Translations Change Last Updated Information

Sort Order Option:

**General Options**

Disconnect Cursors After:  Seconds (0 = Never) Temp Table Instances (Total):

☐ Multi-Company Organization Temp Table Instances (Online):  22

☐ Multi-Currency Maximum App Message Size:

☐ Use Business Unit in nVision Base Time Zone:

☐ Multiple Jobs Allowed Last Help Context # Used:

☐ Allow DB Optimizer Trace Data Field Length Checking:

☐ Grant Access

☐ Platform Compatibility Mode

System Style Sheet:

**Help Options**

F1 Help URL:

Ctrl-F1 Help URL:

Sample page in PeopleSoft Application Designer



5. Enter the name, type, and version of the browser.

The name that you enter appears in the Layout, View in Browser menu.

## Generating HTML

The PeopleSoft Pure Internet Architecture system automatically writes HTML forms or tags for all of the page controls that you add to a page. You do not need to know how to write HTML code to create pages in PeopleSoft Application Designer and then view them in the browser. The table shows this conversion:

### *Page Controls as HTML*

<i>Page Controls</i>	<i>HTML Forms</i>	<i>Static HTML Tags</i>
Edit Box (editable)	<INPUT TYPE=TEXT>	N/A
Long Edit Box	<TEXTAREA>	N/A
Drop-down List Box	<SELECT><OPTION><OPTION>...<=> /SELECT>	N/A
Radio Button	<INPUT TYPE=RADIO>	N/A
Check Box	<INPUT TYPE=CHECKBOX>	N/A
Push Button or Link	<INPUT TYPE=BUTTON> <INPUT TYPE=HYPERLINK>	N/A
Static Text	N/A	HTML text
Static Image	N/A	<IMG>
Grid	N/A	HTML <TABLE>

You can view the HTML that is created for your page by using the Generate HTML feature. The Generate HTML feature writes the HTML code to a file on the local drive. This feature is mainly used for debugging purposes and is not needed for the creation of pages.

To generate HTML for a page:

1. Open the page for which you want to generate the HTML.
2. Select Layout, Generate HTML.

3. Select the browser type and version.

PeopleSoft Application Designer generates the HTML to the c:\TEMP\pshtml directory. The file is named according to the page name and the browser that you select. For example, the Absence History page is named ABSENCE\_HISTORY IE5.00.html.

## Viewing the HTML for a Page

To view the HTML for your page:

1. Open the generated HTML file.

The selected browser appears, displaying how your page appears at runtime.

2. Select View, Source in your browser window.

Notepad or another text-based application appears with the HTML that was used to create your page.

---

**Note.** Different browsers might have different methods for viewing the HTML for your page.

To adjust the look or function of how your page looks in the browser, do so in PeopleSoft Application Designer. If you make changes directly to the HTML in Notepad, your changes are *not* saved to the system. Each time that you open the page in the browser, it reverts to the settings of the original page definition.

---

## Identifying Page Definitions Online

In many cases, the page name online differs from the actual name of the page definition in PeopleSoft Application Designer. While viewing a page in the browser, locate the page definition name by pressing Ctrl + J. A browser page appears revealing the name of the page, along with other system elements:

Browser	IE/7.0
Operating System	WINXP
Browser Compression	ON (gzip)
Tools Release	8.50-808-R1
Application Release	PeopleTools 8.50.00.000
Service Pack	0
Page	USER_GENERAL
Component	USERMAINT
Menu	MAINTAIN_SECURITY
User ID	QEDMO
Database Name	QEDMO
Database Type	ORACLE
Application Server	//BUFFY:9211
Component Buffer Size (KB)	62

[continue](#)

System elements displayed when pressing Ctrl + J

You can also right-click the page and selecting *View Source* from the drop-down menu. The HTML code appears for your page in a text application, such as Notepad. In most cases, the page name is in the sixth or so line of code after *Page:*. You can also identify the component and menu definitions for your page next to the page definition name.



- Read your page definition report.

## Understanding Page Production

If you are changing pages or deleting controls in the page definition, use the tools in the Edit and File menus. To avoid making changes that might adversely affect your application database, review your plans with your database administrator. Together, you can evaluate the impact that your actions might have on your system database as a whole.

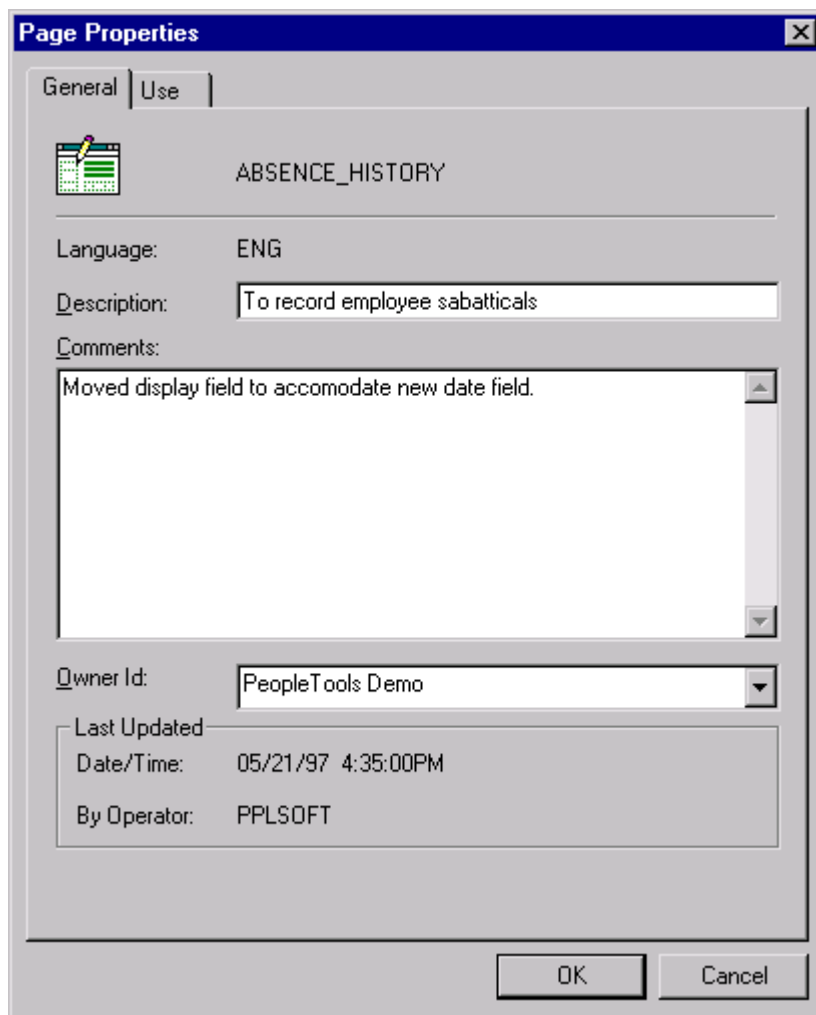
There are several additional steps in the page design process that help make future editing of page definitions easier.

After you design your page, you must set the attributes for how your page should function and make sure that you have the proper documentation to assist you when performing upgrades. Access the Page Properties dialog box by selecting File, Definition Properties in page definition mode.

Use the General tab in the Page Properties dialog box to document the page. Enter both short and long descriptions of the page here. You can also document what changes have been made, or must be made, to a page.

## Changing General Page Information

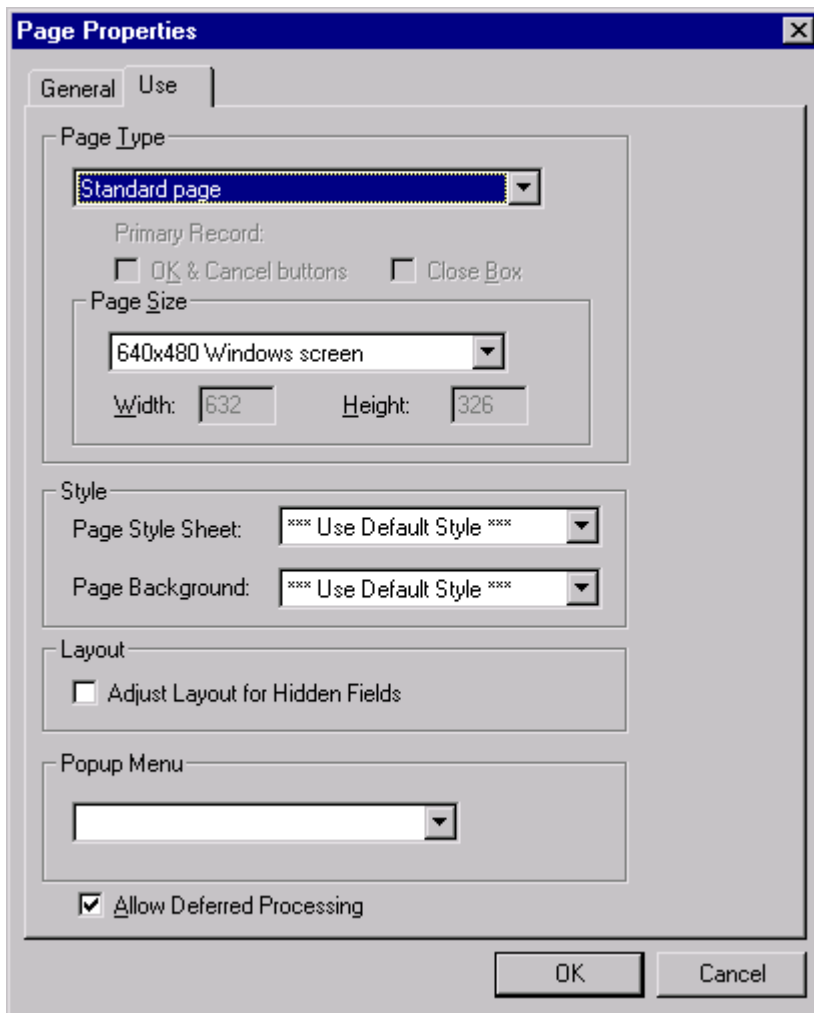
Access the Page Properties dialog box. Select the General tab. Enter a description and any comments about the page and assign an Owner ID.



Page Properties dialog box: General tab

## Changing Page Use Information

Access the Page Properties dialog box. Select the Use tab.



Page Properties dialog box: Use tab

### Page Type

Use the standard page, or select subpage, secondary page, or popup page from the drop-down list box.

See [Chapter 9, "Using Page Controls," Understanding Subpages, page 233.](#)

See [Chapter 9, "Using Page Controls," Understanding Secondary Pages, page 236.](#)

See

### Page Size

To accommodate different types of workstation monitors, you can adjust the page size.

#### 800x600 page inside portal

This is the standard option for viewing pages in the portal, making room for the universal navigation header and the menu. It provides a default page size of 570 × 330 in PeopleSoft Application Designer.

<b>800x600 page without portal</b>	Most pages should be developed using this option. It is the standard option for viewing pages, making room for the navigation header and the menu. This option provides a default page size of 760 × 330 in PeopleSoft Application Designer.
<b>1024x768 page inside portal</b>	Selecting this option provides an optimal display for users with a 1024 x 768 resolution, showing the full transaction page and the menu. Users with an 800x600 display, will see the full transaction page without the menu when the menu is enabled. This option provides a default page size of 760 x 498 in PeopleSoft Application Designer.
<b>1024x768 page without portal</b>	For the power user, this is a high-resolution option. It provides a default page size of 984 × 498 in PeopleSoft Application Designer.
<b>240xVar portal home page comp.</b> (240 x Variable portal home page component)	Use only for creating pagelets for the portal. This size is most usable because it can be viewed in both the narrow area and the wide area of the portal. While the width is set to 240, you can set the height.
<b>490xVar portal home page comp.</b> (490 x Variable portal home page component)	Use only for creating pagelets for the portal. This size can be displayed only in the wide area of the portal. While the width is set to 490, you can set the height.
<b>640x480 Windows screen</b>	Deprecated feature used only for Windows client users. This is designed for VGA resolution. The pages are actually less than 640 × 480 pixels in size in PeopleSoft Application Designer, because they provide space for various windows items, such as the window title, menu bar, and toolbar.
<b>800x600 Windows screen</b>	Deprecated feature used only for Windows client users. This is designed for Super VGA resolution. These pages provide space for the window title, menu bar, toolbar, folder tabs, and status line, as well as space at the bottom for the taskbar.
<b>Custom size</b>	Select to set a specific page size other than those listed previously. Set the width and height manually.

## Style

<b>Page Style Sheet</b>	Select a different style sheet for a specific page to override the style sheet that is selected for the application. If you do not select a different page style sheet (keeping <i>***Use Default Style***</i> ), the system uses the style sheet that is specified on the PeopleTools Options page.
<b>Page Background</b>	Select a different page background style class to override the background style of the page style sheet specified above. If you keep the <i>***Use Default Style***</i> , the background of this page is determined by the default background of the page style sheet.



## Printing Page Definitions

To keep track of your application definitions and refinements, print your page definitions and keep a log or binder with your new and revised pages as you reconfigure your applications. You may want a reference for each page that includes both the onscreen image and the page control information (the record definitions and special uses or edits that you've applied), as well as the information for any subpages that are included in the page definition.

Printing page definitions also helps you debug during testing. Each report includes a unique identifier that's automatically incremented by the system each time that you change the page.

To print a page definition:

1. Open the appropriate page.
2. Select File, Print.

## Altering Page Setup

Access the Print Setup dialog box by selecting File, Print Setup.

### ***Print***

All print options are selected, by default.

<b>Definition</b>	Select to print the list of fields on the page and all of the parameters that you've set for each one.
<b>PeopleCode</b>	Select if you used PeopleCode to design your page
<b>Graphics</b>	Select to provide a snapshot of your page design. Graphics generally take longer to print than page descriptions or PeopleCode reports.

### ***Borders***

<b>Header</b>	Select to print a header at the top of the report indicating the date and time that you printed the report and the database name, page name, version number, and page number. The default is selected.
<b>Footer</b>	Select to print a footer at the bottom of the report indicating the date and time that you printed the report and the database name, page name, version number, and page number. The default is cleared.
<b>Border</b>	Select to print a border or box around a page report. The default is cleared.

**Border Space (mm)**

Select to insert a set amount of space between a graphical border around the page report and the margins of the report. Clear this option to print in character mode (faster), rather than in graphics mode (slower). The default space is 3 mm, but can be changed.

**Margins (mm)**

Set the distance from the edge of the page to the left, right, top, and bottom edges of the page image or report. The defaults are:

- Top: 20.
- Bottom: 5.
- Left: 10.
- Right: 0.

---

**Note.** PeopleSoft has selected the HP LaserJet as the default printer.

---

## Reading Your Page Definition Report

This table lists the columns on the page definition report and describes their contents:

<b>Column</b>	<b>Contents</b>
Num	Sequential number that shows the relative position of the field in the field order list for the page.
Field Type	Field type, such as <i>edit box</i> .
Label Text	Label that's associated with the field, regardless of whether the label is actually displayed on the page.
Label Type	<ul style="list-style-type: none"> <li>• <i>None</i>: Label is not displayed on the page.</li> <li>• <i>Text</i>: Text label.</li> <li>• <i>RFT Long</i>: Long name for the field (from record definition).</li> <li>• <i>RFT Short</i>: Short name for the field (from record definition).</li> <li>• <i>XLAT Long</i>: Long name for the value (from the Translate Table).</li> <li>• <i>XLAT Short</i>: Short name for the value (from the Translate Table).</li> <li>• <i>Bitmap</i>: Bitmap displays on button face.</li> </ul>

<b>Column</b>	<b>Contents</b>
Record Name	Record name where the field is located.
Field Name	Field name on the record definition with which the page field is associated.
Siz (size)	Field size ( <i>custom</i> , <i>minimum</i> , or <i>maximum</i> ).
Alg (alignment)	Alignment ( <i>left</i> or <i>right</i> ).
On	Value of the radio button or the <i>on</i> value of the check box (typically <i>Y</i> ).
Off	The value of the radio button or the <i>off</i> value of the check box (typically <i>N</i> ).
DSP (display)	<i>Yes</i> indicates a display-only field.
INV (invisible)	<i>Yes</i> indicates an invisible field.
CTL (control)	<i>Yes</i> indicates a display control field.
REL (related)	<i>Yes</i> indicates a related display field.
RelNum (related number)	Number (Num column) of the display control field for this related display field.
CUR (currency)	<i>Yes</i> indicates a multicurrency field that does not display unless the user is operating in multicurrency mode.
DER (derived)	<i>Yes</i> indicates a field from a derived or work record.
OccLvl (occurs level)	Occurs level of scroll bar that governs this field.

You may see additional reference lines below each field or at the end of the report:

<b>Reference</b>	<b>Description</b>
Page Help Context: nnn	Indicates that the page has been assigned a page help context number, linking it to a help file that describes how the field is used wherever it appears in the database.

<i>Reference</i>	<i>Description</i>
Field Help Context: nnn	Indicates that this field, as used on this page, has been assigned a record field help context number, linking it to a help file that describes how the field is used only as it appears in this record definition.

---

## Working with Pages in Foreign Languages

This section provides an overview and discusses:

- Performing a Save As with foreign language pages.
- Deleting foreign language pages.

### Understanding Working with Pages in a Foreign Language

Many organizations are multinational, operating their business in multiple countries and continents. For efficiency, these organizations typically deploy a single application in multiple languages. However, because internal application developers and administrators may not be familiar with each language in which applications are deployed, this can cause difficulty when updating or maintaining the applications. To assist developers at multinational organizations, PeopleSoft provides a Save As option and a Delete option when working with foreign language pages.

### Performing a Save As with Foreign Language Pages

PeopleSoft enables developers to modify pages in their base language and save the page as another language. This decreases the time required to develop pages in other languages, and it relieves developers from having to modify pages in languages they do not understand. You can open a page definition in your base language and save the changes to another language, using the Save As option.

The Language drop-down list box on the Save As dialog is only available when you are signed on to the database in the base language. When performing the Save As, select the appropriate language from the Language drop-down list.

If the page definition exists in the foreign language (non-base language) the system prompts you to confirm overwriting the definition.

---

**Note.** You can save modifications to a non-base language only if the base language definition already exists. If the base language page does not exist, the system saves the page to the base language regardless of the language selected.

---

---

**Note.** After doing a Save As to another language, the page will still display in the Application Designer in the base language.

---

## Deleting Foreign Language Pages

If you need to delete non-base language pages, when you are signed onto the database in the base language, the Delete dialog box contains the Language drop-down list. The Language drop-down list enables the developer to delete multiple pages from multiple non-base languages.

---

**Note.** Deleting the base language page automatically deletes all translated pages for the deleted base language page.

---



## Chapter 12

# Creating Component Definitions

This chapter provides an overview of component definitions and discusses how to:

- Define components.
- Specify market-specific components.
- Determine where a component is used.
- Set component properties.
- Override the search record.
- Access the message catalog.
- Enable Expert Entry feature in Security component.

---

## Understanding Component Definitions

A component represents a complete business transaction. It comprises either a single page or a set of pages that are meant to be processed as one. After you create pages, you add them to one or more components to access them through menus or in business processes.

This section discusses:

- What component definitions manage.
- The component buffer.

## What Component Definitions Manage

Component definitions manage:

- Grouping of pages and their associated tab labels.
- Search records that are used to retrieve data for the page.
- Access keys for folder tab navigation.
- Links at the bottom of a page.
- The toolbar at the bottom of a page.

General	Pages	PeopleTools	Process	Sign-on Times
Permission List: ALLPNLS Description: ALLPNLS				
<b>PeopleTools Permissions</b>				
<input checked="" type="checkbox"/> <b>Application Designer Access</b> <a href="#">Object Permissions</a> <a href="#">Tools Permissions</a> <a href="#">Misc. Permissions</a> <input checked="" type="checkbox"/> <b>Data Mover Access</b> <input checked="" type="checkbox"/> <b>Definition Security Access</b> <input checked="" type="checkbox"/> <b>Query Access</b>				
<b>Data Archival</b>				
<input type="checkbox"/> <b>Generate SQL</b> <input type="checkbox"/> <b>Run SQL</b> <input type="checkbox"/> <b>Edit SQL</b> <input type="checkbox"/> <b>Purge Audit</b>				
<div>  Save            Return to Search            Next in List            Previous in List            Add            Update/Display         </div>				

Sample component with multiple pages (tabs)

If a component contains more than one page, you can control which page appears first by setting the display order in PeopleSoft Application Designer. At runtime the user can navigate between the pages of a component by selecting:

- Folder tabs at the top of the page.
- Links at the bottom of the page.
- Specified access or hot keys.

## Component Buffer

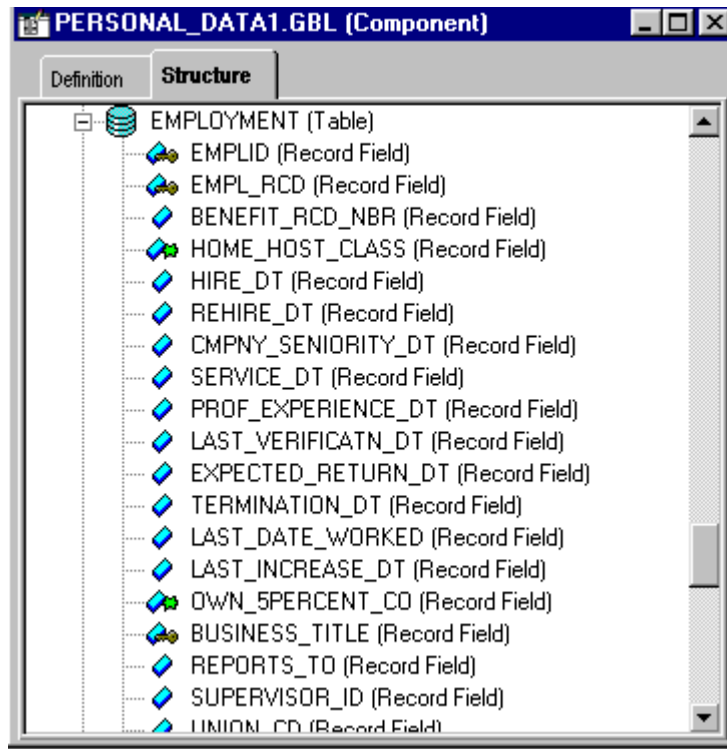
When you open any page in a component, the system retrieves all of the data records for the entire component and stores them in one set of record buffers, organized by scroll level and then by page level. This *component buffer* consists of rows of buffer fields that hold data for the various records that are associated with page controls, including the primary records, related display records, derived work records, and translate table records.

You can change the records in memory either through the user actions on the page or through PeopleCode that is associated with different page controls. At runtime, you open and save the entire component, not just individual pages. When you save *any* page in the component, you automatically save the whole component.



## Structure Tab

The structure view shows records and scrolls in a tree representation. It provides information about page levels and component records and their roles. You can double-click the objects in this view to open their definitions.



Component structure view

You can view the PeopleCode that is attached to any of the components by right-clicking and selecting the View PeopleCode option from the pop-up menu. The PeopleCode Editor opens directly in the workspace, displaying the related PeopleCode.

---

**Note.** The Structure tab only displays the runtime state of the PeopleCode. That is, it only displays record field PeopleCode. For example, PeopleCode programs that are orphaned as a result of a page definition change do not appear in the Structure tab. On the other hand, these orphaned PeopleCode programs do appear in the PeopleCode Editor, which displays the design-time view of PeopleCode.

---



A key icon appears next to all key and alternative search key fields in the component structure view.



An asterisk icon appears next to all fields that are required.

## Creating a New Component Definition

To create a new component definition:

1. With a new or existing project open in PeopleSoft Application Designer, select File, New.





2. Drag the row to the correct position.

The page is inserted immediately below the highlighted line. In the preceding example, the third page (UK) becomes the second.

## Copying or Moving Pages to Another Component

After you create a component, you can copy or move pages from one component to another.

---

**Note.** The system does not allow you to move or delete all pages from a component definition. The component must contain at least one page definition.

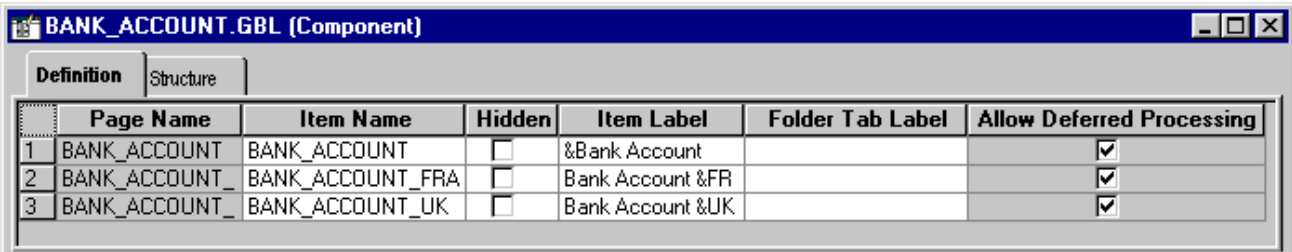
---

To copy or move a page from one component to another:

1. Open both components.
2. In the left-hand column, select the number of the page that you want to copy.
3. Copy or cut the page:
  - To copy the page to another component, select Edit, Copy.
  - To move the page to another component, select Edit, Cut.
4. Select the other component.
5. Select Edit, Paste.

## Setting Page Attributes

Each page in a component has attributes. Each attribute is represented by a column in the component definition.



	Page Name	Item Name	Hidden	Item Label	Folder Tab Label	Allow Deferred Processing
1	BANK_ACCOUNT	BANK_ACCOUNT	<input type="checkbox"/>	&Bank Account		<input checked="" type="checkbox"/>
2	BANK_ACCOUNT_	BANK_ACCOUNT_FRA	<input type="checkbox"/>	Bank Account &FR		<input checked="" type="checkbox"/>
3	BANK_ACCOUNT_	BANK_ACCOUNT_UK	<input type="checkbox"/>	Bank Account &UK		<input checked="" type="checkbox"/>

Component item label and tab label

<b>Page Name</b>	<p>Displays the read-only name of the page definition. If you rename the page, this column is updated automatically.</p> <p>When creating page definitions, you might want to use similar names to make them easily recognizable as a group of pages. For example, if you create three pages to hold the information for the Personal Data Table, the page names might be:</p> <ul style="list-style-type: none"> <li>• PERSONAL_DATA1</li> <li>• PERSONAL_DATA2</li> <li>• PERSONAL_DATA3</li> </ul>
<b>Item Name</b>	<p>Displays a name for each page in the component. This name must be unique in the component, and the default is the page name. This name is for informational purposes only. If you use the same page on more than one component, you might want to change the item name to reflect the purpose of the page in the component.</p>
<b>Hidden</b>	<p>Specifies whether the page can be viewed by the user at runtime. Pages are usually hidden when they are used in work groups or associated with derived work records. For example, information from a page might need to be loaded into the buffer for PeopleCode to perform calculations, but the user does not need to see it.</p> <hr/> <p><b>Note.</b> Marking the page as hidden also hides the toolbar, including the Save button, even if you choose to display the page by means of another method, such as the Transfer PeopleCode function.</p> <hr/>
<b>Item Label</b>	<p>Serves as the default folder tab label, unless a different label is specified. The item label should be unique for each page in a single component menu. The folder tab label is usually used when shorter names are needed for folder tabs.</p> <p>In the preceding example (BANK_ACCOUNT.GBL), for the first page in the component, the same text appears on both the folder tab and the component menu. The second and third pages have different menu names and folder tabs.</p>
<b>Folder Tab Label</b>	<p>Displays the text that appears on the folder tab label. If no text is specified, the text of the item label is used on the folder tabs.</p>
<b>Allow Deferred Processing</b>	<p>Indicates whether deferred processing is active for that page in its property settings.</p>

### Setting Access Keys

You can set *access keys* by placing an ampersand (&) in the text of each item label. Put the ampersand in front of the letter that you want to be underlined on the folder tab at runtime.

Definition				
Page Name		Item Name	Hidden	Item Label
1	JOB_DATA1	JOB_DATA_1	<input type="checkbox"/>	Work Location
2	JOB_DATA2_US	JOB_DATA_2	<input type="checkbox"/>	&Job Information
3	JOB_DATA3	JOB_DATA_3	<input type="checkbox"/>	Job &Labor
4	JOB_DATA_ERNDI	JOB_EARNINGS_DIST	<input type="checkbox"/>	&Payroll
5	JOB_DATA_BENP	BENEFIT_PROGRAM_	<input type="checkbox"/>	&Salary Plan
6	EMPLOYMENT_DT	EMPLOYMENT_DATA_	<input type="checkbox"/>	&Compensation

Item labels with ampersands (&)

At runtime, the user can navigate between pages by pressing Alt, plus the letter that is underlined, followed by Enter.



Underlined access keys on tabs

### See Also

[Chapter 8, "Designing Pages for Optimal Performance and Accessibility," Using Instructional Text, page 202](#)

## Validating Components

PeopleSoft Application Designer includes a validate utility to verify the integrity of a component. The validate utility runs a series of tests on components to check that the component will operate properly at runtime and sends the results to the Validate tab in the output window.

To validate a component:

1. Open the desired component definition.
2. Select Tools, Options, Validate, and make sure that Validate PeopleSoft Internet Architecture Execution is selected.

---

**Note.** This option must be selected for the validate tests to be invoked.

---

3. Select Tools, Validate Component.
4. View the results displayed in the Validate tab, and correct any errors.

### See Also

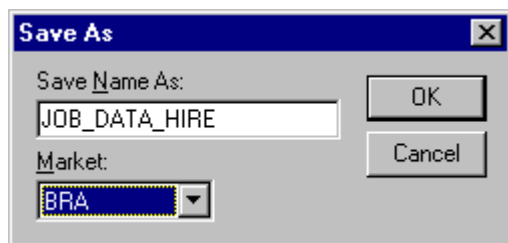
[Chapter 2, "Using PeopleSoft Application Designer," Setting PeopleTools Options, page 21](#)

[Chapter 3, "Working With Projects," Validating Projects, page 58](#)

---

## Specifying Market-Specific Components

You might want to add custom features to a component that is specific to an international market or region. For example, suppose that a page is used in a procedure for hiring new employees and that the procedure includes special transactions that are required only when an employee is hired in Brazil. You can create a custom component using the same component name as the component from which it is derived and saving it using a market setting of *BRA*.



Component saved with BRA market setting

Market-specific components are independent of system-side language settings and are accessible to any user who has security access to the component. This makes it possible for an English-speaking user in New York, for example, to perform a procedure for hiring an employee in Brazil.

Access the market setting of a component using the PeopleCode %Market system variable. Consequently, you can maintain a single set of PeopleCode programs for a set of market-specific components and build conditional logic into the PeopleCode programs that run only in components that have specific market settings.

### See Also

*Enterprise PeopleTools 8.50 PeopleBook: PeopleCode Language Reference*, "System Variables," %Market

---

## Determining Where a Component Is Used

The Find Definition References feature enables you to generate a list of the menus in the database that reference a specific component; that is, a list of the menus in which the component is used. You can invoke this feature from either the project workspace or the workspace. Select Find Definition References from the Edit menu.

### See Also

Chapter 2, "Using PeopleSoft Application Designer," Finding Definitions, page 34

---

## Setting Component Properties

In the Component Properties dialog box, specify the update and data entry actions; on the Use tab, specify the search record information. You can also save notes about the component on the General tab. Use the Internet tab to set attributes that affect how the component appears in the browser at runtime.

This section discusses how to:

- Open the Component Properties dialog box.
- Set general properties.
- Set use properties.
- Set internet properties.

### Opening the Component Properties Dialog Box

To open the Component Properties dialog box:

1. Open the component definition.
2. Select File, Definition Properties.

The Component Properties dialog box appears with the General tab active.

### Setting General Properties

Access the Component Properties dialog box.

<b>Description</b>	Enter a descriptive name for the component.
<b>Owner ID</b>	Select the application to which this component belongs. This list is helpful to identify the applications that are associated with the component during the application development phase.
<b>Last Updated</b>	View the date and time of the last modification made to the component and the name of the user who made the modification.

### Setting Use Properties

Access the Component Properties dialog box. Select the Use tab.

**Component Properties**

General Use Internet

**Access**

Search record:  
PSCLASSDEFN SRC

Add search record:

☐ Force Search Processing

Detail page:

**Actions**

☒ Add  
☒ Update/Display  
☐ Update/Display All  
☐ Correction

☐ Disable Saving Page  
☒ Include in Navigation  
☐ Mandatory Spell Check

**3-Tier Execution Location**

**Component Build**

☐ Client  
☐ Application server  
☒ Default (application server)

**Component Save**

☐ Client  
☐ Application server  
☐ Application server (with edits)  
☒ Default (application server)

OK Cancel

Component Properties dialog box: Use tab

## Access

**Search record** Specify the search record for this component.

**Add search record** Specify if you want a different search record specifically for add actions. For example, if you selected the auto-numbering option for employee IDs (EMPLID), don't include EMPLID in the search record. Likewise, you might want to create special security views for add actions that limit the rows that users can add, based on specific search criteria. The system default is the standard search record if you don't specify an add search record.

**Force Search Processing** Select to always run search logic (SearchInit PeopleCode) for this component. The default is cleared.

**Detail page** No longer used in PeopleSoft applications. This option appears for upgrade purposes only.

---

**Note.** To see a subset of search records, type a few characters first. The drop-down list box is populated with a list of all record definitions with names that begin with those characters. This reduces access time and enables you to find specific records faster.

---

## Actions

<b>Add</b>	Select to add a new high-level key, such as a new employee ID or customer. Except in the case of effective dating, Add is used to insert a new current row or to update future rows.
<b>Update/Display</b>	Select to update existing rows only.
<b>Update/Display All</b>	Select to update current and future rows in an effective-dated record. Use only with effective-dated records. Do not use these actions unless the main record that is associated with the page definitions is effective-dated. This is translated to include history at runtime.
<b>Correction</b>	Select to update any rows (history, current, and future) in an effective-dated record. Use only with effective-dated records. This is translated to correct history at runtime.
<b>Disable Saving Page</b>	Select when you want to hide the Save button in the toolbar and disable the Alt+1 (Save) hot key. This prevents the user from being prompted to save when exiting a page. However, it doesn't prevent using PeopleCode to save a page with the DoSave() or DoSaveNow() functions. This can be helpful for applications in which the user isn't making database changes and doesn't need to be prompted to save.
<b>Include in Navigation</b>	Select to include the component in the menu navigation at runtime. The default is selected. If you do not want the component included, clear the check box.
<b>Mandatory Spell Check</b>	Select to ensure that spell check is run on all eligible spell check enabled fields in the component when the user saves the page. A field is eligible if it has been changed (or its ReEdit set) and it has not already been manually spell checked since it was last changed.

---

**Note.** If you must add a new high-level key, such as a new employee number or customer, select both Add and Update/Display. If you are adding information to an existing high-level key, such as adding education data for an employee, select only Update/Display. You're not adding a new employee ID—it already exists on the database.

---

## Three-Tier Execution Location

This section applies to Windows client applications only.

To partition application processing between the client and the application server, you must define units that, as a whole, run in one location or the other. We call these units *processing groups*. Processing groups can encompass one or more PeopleCode events.

**Component Build**

Specify where you want all processing to occur after the key list of a page is selected and before the user can interact with the page. This includes building component buffers and running many types of PeopleCode.

**Component Save**

Specify where you want all processing to occur after the user saves the component and SaveEdit PeopleCode validations have succeeded. It includes SavePreChange, WorkFlow, and SavePostChange PeopleCode and updates to the database.

## Setting Internet Properties

Access the Component Properties dialog box. Select the Internet tab.

The screenshot shows the 'Component Properties' dialog box with the 'Internet' tab selected. The dialog is divided into several sections:

- Search Page:**
  - Primary Action: ☐ New, ☒ Search
  - Default Search Action:
  - Default Search/Lookup Type: ☒ Basic, ☐ Advanced
  - ☒ Allow Action Mode Selection
  - Link To Access Add Page -- Message Set/Nbr:
  - Link To Access Search Page -- Message Set/Nbr:
  - Instructional Text -- Message Set/Nbr:
- Multi-Page Navigation:**
  - ☒ Display Folder Tabs (top)
  - ☒ Display Hyperlinks (bottom)
- Processing Mode:**
  - ☒ Interactive, ☐ Deferred
  - ☐ Allow Expert Entry
  - ☐ WSRP Compliant
- Toolbar:**
  - Selected Toolbar Actions:
 

<input checked="" type="checkbox"/> Save	<input type="checkbox"/> Refresh
<input checked="" type="checkbox"/> Cancel	<input type="checkbox"/> Notify
<input type="checkbox"/> Spell Check	<input checked="" type="checkbox"/> View WorkList
<input checked="" type="checkbox"/> Return to List	<input checked="" type="checkbox"/> Next in WorkList
<input checked="" type="checkbox"/> Next in List	<input checked="" type="checkbox"/> Previous in WorkList
<input checked="" type="checkbox"/> Previous in List	<input checked="" type="checkbox"/> Add
<input type="checkbox"/> Next Page in Component	<input checked="" type="checkbox"/> Update/Display
<input type="checkbox"/> Previous Page in Component	<input type="checkbox"/> Update/Display All
	<input type="checkbox"/> Correction
  - ☐ Disable Toolbar
- Pagebar:**
  - ☒ Help Link
  - ☒ Copy URL Link
  - ☒ New Window Link
  - ☒ Customize Page Link
  - ☐ Disable Pagebar

At the bottom right are 'OK' and 'Cancel' buttons.

Component Properties dialog box

## Search Page

<b>Primary Action</b>	<p>Select the default mode by which the user accesses the associated page. Select New for Add mode. Select Search to use the action that is specified in the Default Search Action drop-down list box.</p> <p>Specify where you want all processing to occur after the key of a page is selected and before the user can interact with the page. This includes building component buffers and running many types of PeopleCode.</p>
<b>Default Search Action</b>	Select from a list of the actions that you set in the Actions group box on the Use tab of the Components Properties dialog box. If you set only one action, this drop-down list box provides that choice as the default. The default search action determines the default that users see, but they can select another option from the toolbar at runtime.
<b>Default Search/Lookup Type</b>	Select Basic or Advanced. The basic search page provides a choice of which field to search (either the primary and alternate search keys). PeopleSoft recommends using the basic search page for new users because it is easier to understand. If you select Advanced, the advance search page is the default. The search and lookup type applies to the search page and all of the lookup pages in the component.
<b>Allow Action Mode Selection</b>	Select to display the Include History and Correct History check boxes on a search page. If the user selects Include History on a search page, the page appears in Update/Display All mode. If the user selects Correct History, the page appears in Correction mode. However, both of these check boxes appear only on the search page if Update/Display All and Correction were selected as actions in the Use properties for the component and the user has security access to the modes. This check box is selected by default.
<b>Link to Access Add Page and Link to Access Search Page</b>	Specify the message set and number to display on an Add or Search page.
<b>Instructional Text</b>	This is set to display the Find an Existing Value heading on the search page. You can change the default message or redirect the component to use another message that is more appropriate for your use.

## Toolbar

Select the toolbar actions that you want to display on the component pages. When you select a check box, that item appears as a toolbar button at the base of the component at runtime. All toolbar actions are set to display by default with the exception of Next Page in Component ,Previous Page in Component and Notify. Note that the default settings for the Action and Save buttons are impacted by settings on the Use tab of the Component Properties dialog box.

Disabling any of the toolbar options also disables the hot key for that option in the PeopleSoft Pure Internet Architecture.

<b>Save</b>	Adds a Save, OK, or Apply button to the toolbar, depending on the page type.
-------------	--

<b>Cancel</b>	For modal components only. Clearing this box removes the Cancel button from the toolbar.
<b>Spell Check</b>	Runs spell check for all spell check enabled fields in the component, regardless of whether any changes have been made to the field.
<b>Return to List</b>	Returns user to the search page.
<b>Next in List and Previous in List</b>	Populates the page with the next or previous row of data from the database as shown on the Search page.
<b>Next Page in Component and Previous Page in Component</b>	Opens the next or previous page in the component if there is more than one page.
<b>Refresh</b>	Redisplays the page without saving any entered data.
<b>Notify</b>	Enables the user to send a notification to other users.
<b>View Worklist</b>	Enables the user to view the worklist from the current component page.
<b>Next in Worklist and Previous in Worklist</b>	Enables the user to access the next or previous item in the worklist.
<b>Add, Update/Display, Update/Display All, and Correction</b>	Select each to enable these toolbar buttons for this component. See <a href="#">Chapter 12, "Creating Component Definitions," Setting Use Properties, page 371.</a>
<b>Disable Toolbar</b>	Select if you do not want to display the toolbar on the component pages.

### ***Multi-Page Navigation***

<b>Display Folder Tabs</b>	Select to display folder tabs at the top of the component page.
<b>Display Hyperlinks</b>	Select to display links at the bottom of the component page.

### ***Processing Mode***

<b>Interactive</b>	Select to send messages to the server whenever logic must be run. When the user presses Tab to exit a field-level event, a transmission to the application server occurs to run that field-level event and the page is refreshed.
<b>Deferred</b>	Selected as the default. This mode reduces transmissions to the application server.  See <a href="#">Chapter 8, "Designing Pages for Optimal Performance and Accessibility," Understanding Processing Modes, page 191.</a>

**Allow Expert Entry** This is available for both processing modes.  
 See [Chapter 12, "Creating Component Definitions," Enabling the Expert Entry Feature, page 382.](#)

### ***WSRP Compliant***

Select WSRP Compliant to enable the component for WSRP publishing. If the component is enabled for WSRP publishing the component can be rendered, in-line, within the WSRP consuming portal.

If the component is *not* enabled for WSRP, when a user navigates to the component from within a WSRP portlet, the system directs the user to the PeopleSoft Portal, which loads the component. That is, the user has effectively left the WSRP consuming portal page and navigated to the PeopleSoft system.

---

**Note.** This option enables you to distinguish between components that are known to function only in the PeopleSoft Portal and those that are WSRP compliant.

---

See *Enterprise PeopleTools 8.50 PeopleBook: PeopleTools Portal Technologies*, "Understanding Web Services For Remote Portlets (WSRP)."

### ***Pagebar***

The pagebar is a small menu bar that displays at the top of each component page just below the universal navigation header. You can select which options display to the user for each component. All options are selected by default. Select the Disable Pagebar check box to disable the pagebar altogether.

**Help Link** Opens the online help PeopleBook entry for the current page.

**Copy URL Link** Copies the current page URL to the clipboard. The copied URL includes page, action, and search key information to display in the context of the portal, if the user passed through the portal. If the user did not pass through the portal, then only the page displays without the portal frames.

This option adds a small http button to the pagebar.

**New Window Link** Opens a new browser window with the search page for the current component. Users can view or enter data in the new window.

- Customize Page Link** Opens the Page Customization page which allows the user to control the initial display of the component. This option enables the user to:
- Select the initial page in the component to be displayed.
  - Save the state of the expanded and collapsed sections.
  - Save the state of the View All settings on the page for grids and scroll areas.
  - Customize the page field tabbing order.

Users can also elect to copy and share their settings for the component with other users.

---

**Note.** If you set this option for a particular component, all users accessing the component are able to customize the pages in the component unless their personalization options prevent them from customizing pages. Security permission list settings or personalization options do not allow the option to be enabled or disabled per end user per component.

---

See *Enterprise PeopleTools 8.50 PeopleBook: Security Administration*, "Managing PeopleSoft Personalizations," Understanding General Options.

See *Enterprise PeopleTools 8.50 PeopleBook: Using PeopleSoft Applications*, "Setting User Preferences," Customizing Pages.

You can also enable or disable any of these options using PeopleCode.

See *Enterprise PeopleTools 8.50 PeopleBook: PeopleCode API Reference*, "Page Class."

### **More on the Customize Page Option**

In order for the page customization option to display to the user, in addition to the selection of the Page Customization Link in the Pagebar, certain criteria must be met:

- A PeopleSoft Pure Internet Architecture page, including search and secondary pages, must be displayed to the user.
- The page cannot be a PeopleTools generated page, such as the Page Customization page or the Customize Column and Sort Order page for grids.
- The current page is not an iScript.
- The Customize Pagelink cannot be hidden using PeopleCode for the current component.
- The Customize Page Settings, CUSTOMPGSET, personalization option on the Define Personalizations page must be enabled.

This option is enabled by default. If this option is disabled, the end user cannot define, share, or copy page customizations and all existing page customizations for the user are deleted.

- The current page must allow some form of personalization, based on the end users security profile and meeting *at least one* of these conditions:
  - Display more than one page element with a TABINDEX  $\geq 1$  supporting customization (assume that all collapsible sections are expanded when determining visibility).
  - Contain at least one visible scroll area or grid with a visible View All link, enabled or disabled.
  - Contains at least one visible collapsible section.

---

## Selecting the Search Record

This section provides an overview of search records and discusses how to:

- Create a search record.
- Override a search record.

## Understanding Search Records

Correctly configuring and selecting the search record for a component is critical because the system uses the search record to generate the query that retrieves the data, which appears in the component.

The search record:

- Builds the search page for the component.
- Builds the search query and populates level zero for every page in the component.

When you click a content reference to access a component, the system examines the search record and dynamically builds the search page based on the search key, alternate search key, and list box item properties of the search record. If you designate fields as search keys or alternate search keys, then those fields appear on the search page. When the user enters values in the search fields and clicks the Search button, the system incorporates those values into the WHERE clause of the search query. If the values provided by the user uniquely identify *one* row of data, then the system proceeds directly to the main page with the information available. Otherwise the query retrieves the rows that match the criteria and displays the rows in a list box at the bottom of the search page. If you define fields as list box items on the search record, the system displays those fields as columns in the list of rows in the result set.

**Menu**

- Search: [ ]
- My Favorites
- ICE
  - Search - Configurable
  - Search - Ventry (Text)
  - Search - Old Key Fields
  - Work Report / Incident
  - Work Resolution
    - Resolution
    - Resolution Approval/Pin/Review
  - Bundles
  - Metrics & Reporting
  - User Setup
  - ICE Build Tracking
  - Test Cases
  - Test Script Database
  - Translation Reviewer
- ICE Administration
- Self Service: ICE Security
- Services Repository
- Reporting Tools

**Resolution**  
Enter any information you have and click Search. Leave fields blank for a list of all values.

**Find an Existing Value** **Add a New Value**

Resolution ID: [ ] [ ]  
 State: [ ] 0120-Posted  
 Resolution Type: [ ] [ ]  
 Bundle ID: [ ] [ ]  
 Create Date: [ ] [ ]  
 Created By Name: begins with [ ] [ ]  
 Resolved By Name: begins with [ ] [ ]  
 Subject: begins with [ ] [ ]

These are Search Key and Alternate Search Key fields on the Search record

Search Clear Basic Search Save Search Criteria

**Search Results**  
Only the first 300 results can be displayed. Enter more information above and search again to reduce the number of search results.

View All

Resolution ID	State	Resolution Type	Bundle ID	Lead Product	Create Date	Created By ID	Created By Name
7233114	Posted	Applic	0	ISCM Transportation Management 8.0	09/26/2007 19056	Peter Oram	
7233113	Posted	Applic	0	ISCM Transportation Management 8.0	09/26/2007 19056	Peter Oram	

These are List Box fields on the Search Record. Note that "Lead Product" and "Created By ID" aren't key fields, just Listbox items so they aren't above, just here.

Search page with search keys, alternate search keys, and list box items

When you select a row from the list box, the runtime system generates a second query to fill the level 0 row in the component. The runtime system examines the level 0 record for the component and checks its key fields. It tries to retrieve level 0 key field values from the fields in the list box and uses those values in the WHERE clause used to populate the level 0 record.

The search record for a component might differ from the level 0 record definition for a component and might contain any number of search keys and list box items, but it must contain all of the key fields that are in the level 0 record for a component.

See [Chapter 6, "Creating Record Definitions," Setting Record Field Use Properties, page 130.](#)

**Note.** When you add page items that are based on derived work records, do not select that record definition as the search record. Instead, select a record definition that stores the primary key or a SQL view that searches for the criteria that best enables users to identify the rows that they want to retrieve.

The search record might also contain logic to refine the search and limit rows of data. PeopleSoft applications implement data permission (or row-level) security in this way.

If you are configuring menus in an existing PeopleSoft application, the easiest way to decide which search record to use is to look at the other search records that are used on the same menu. Look for pages with similar key structures and then evaluate the search records to see if any are suitable. You might want to print the PeopleTools Menu Listing cross-reference report (XRFMENU) to review which search records are used for pages and gain a better understanding of how search records are assigned in the application.

## Creating a Search Record

To create a search record:

1. Create the record definition and add the fields.

See [Chapter 6, "Creating Record Definitions," Creating a New Record, page 118.](#)

2. Select the key fields, search key fields, and list box items.

See [Chapter 6, "Creating Record Definitions," Setting Record Field Use Properties, page 130.](#)

3. If the record is a view, write the SQL statement in the SQL editor.

See [Chapter 6, "Creating Record Definitions," Creating SQL View and Dynamic View Select Statements, page 145.](#)

4. Save and build the record definition.

See [Chapter 7, "Administering Data," Running the Build Process, page 148.](#)

## Overriding the Search Record

You might want to reuse a component, changing only its search record.

You can accomplish this by *overriding* the component search record at the time that the component is invoked from a menu. To override a component search record, specify a different search record in the properties of the menu item that invokes the component. The component uses the override search record when it is invoked from that specific menu item; the search record set in the component properties remains unchanged.

By reusing components in this way, you can limit redundancy, keeping your application smaller and easier to maintain.

### See Also

*Enterprise PeopleTools 8.50 PeopleBook: PeopleCode Language Reference*, "PeopleCode Built-in Functions," SetSearchDialogBehavior

---

## Accessing the Message Catalog

Message sets and number settings come from the PeopleTools message set catalog.

To access the message catalog:

1. Select PeopleTools, Utilities, Administration, Message Catalog.
2. On the search page, enter the message set number that you want to access.

Component definitions support messages between 0 and 99,999.

3. Click Search to access a message catalog page.
4. Enter the message set and number on the Internet tab of the Component Properties dialog box.

For example, enter message set and number *124* and *50* to display the Find an Existing Value message on the search page in the browser.

---

## Enabling the Expert Entry Feature

This section provides an overview of the Expert Entry feature and discusses how to enable it in Security.

### Understanding Expert Entry

Expert entry enables a user to change from interactive to deferred mode at runtime for appropriate transactions.

---

**Note.** Expert entry should be used only by *expert* users of the transaction who understand how all of the fields in the transaction behave. It requires the user to know the exact keystroke sequence for the transaction. This sequence typically varies for a transaction by customer, depending on the customer's setup and entry requirements. Therefore, expert entry should not be used by occasional users, because they might not be aware of the nuances of the transaction and when it is necessary to press the Refresh hot key.

---

To use expert entry, select:

- The Allow Expert Entry check box on the Internet tab of the Component Properties dialog box.
- The Enable Expert Entry check box in the user's profile.

If these check boxes are selected, an Expert Entry check box appears on the bottom of the page at runtime. If the user selects this check box, the transaction runs in deferred mode, regardless of the setting on the component, page, or field property.

When you set the component property to deferred mode, that component operates in deferred mode processing for all users, not just those who have expert entry enabled in their profiles.

If you want the component to process in deferred mode for some users and interactive mode for other users:

- Set the component property to Interactive mode.
- Select the Allow Expert Entry check box.

This enables security user profiles to control the users for whom deferred mode processing is possible.

---

**Warning!** Users who modify component properties from interactive mode to deferred mode or select the Allow Expert Entry check box should understand that this is an adaptation of the software, and, as such, they are responsible for thoroughly testing and supporting this change.

---

### Refresh Button

The Expert Entry feature enables you to specify whether a Refresh button should be included on the component toolbar when the page is displayed. The Refresh button forces a transmission to the server to determine which fields have changed since the last transmission to the server and to run any processing logic that is associated with those changes.

Users can also refresh by pressing the Alt+0 hot key, which keeps the cursor in the same field when the page is refreshed. One can refresh any time during data entry to enable an *expert* user to:

- Update related display field values for the data that is already entered.
- Recalculate totals and balances.
- Provide defaults based on prior data that was entered on the page.
- Validate the data that has been entered on the page so far.
- Invoke the code to hide or gray items associated with specific fields.

### ***Limitations of Expert Entry***

These limitations exist when expert entry is enabled:

<b>Hide, Unhide, Gray, and Ungray</b>	The user must click the Refresh button after entering a value in a field that is associated with PeopleCode associated that hides, unhides, grays, or ungrays other fields on the same page.
<b>Drop-down list box</b>	The user must click the Refresh button after entering a value in a field that is a high-order key that is used to control the values in a drop-down list box.
<b>Default values</b>	Whenever expert entry is properly enabled and selected by the user at runtime, the transaction runs in deferred mode. As such, defaults, totals, balances, and related displays are not updated until the next transmission to the application sever (for example, when the user clicks a button, icon, link, or another tab). If the user wants the page updated before then, the user must click the Refresh button.

## **Enabling Expert Entry Through Security**

To have the Expert Entry check box appear in the component at runtime, the security administrator must enable it in Security. This enables the Expert Entry check box to appear in any components for which you select the Allow Expert Entry property on the Internet tab of the Component Properties dialog box.

To enable expert entry through Security:

1. Sign in to the PeopleSoft system in your browser.
2. Select PeopleTools, Security, Use, User Profiles.
3. Select the General tab.
4. Select Enable Expert Entry in the General Attributes group box.
5. Click Save.



## Chapter 13

# Creating Menu Definitions

This chapter provides an overview of menus and discusses how to:

- Define custom component menus.
- Work with menu definitions.
- Define pop-up menus.

---

## Understanding Menus

A menu is a logical grouping for assigning security to your system. You create a menu as a placeholder for components. After you save your menu, use the Registration Wizard to assign components and security.

You can create two types of menus using PeopleSoft Application Designer:

- *Custom* (component) menus provide an internal reference for components and pages. They are grouped logically into menu groups to assign security. They are not used for navigational purposes.
- *Pop-up* menus appear when a user clicks a pop-up button.

### See Also

[Chapter 14, "Using the Registration Wizard," page 397](#)

*Enterprise PeopleTools 8.50 PeopleBook: Security Administration*, "Understanding PeopleSoft Security"

---

## Defining Custom Component Menus

This section discusses how to:

- Create custom component menus.
- Set menu item properties.
- Set general properties.
- Set use properties.

## Creating Custom Component Menus

After you create records, pages, and components, register your new component using the Registration Wizard. To do so you must first create a menu definition, if one does not already exist. This enables you to add the component to the new menu (for navigational purposes) and register it in the portal.

To create a new menu definition:

1. From the PeopleSoft Application Designer toolbar, select File, New..
2. Select Menu from the list.
3. Click OK.
4. Select the Standard option for the menu type.
5. Click OK.

The thick rectangle surrounding the bar item indicates that this element in the menu definition is currently selected. The empty, dashed rectangle is a placeholder for a new bar item label.

6. Assign a bar name and label.

You can define multiple menu bars for the menu. Double-click the rectangle to display the Bar Item Properties dialog box.

7. Enter the bar item name and label.

The most commonly used bar item name is *Use..*

8. Click OK to accept the settings in the Bar Item Properties dialog box.

The new label appears in the menu and the empty rectangle appears in a new location.

9. Set the menu item properties.

When creating a menu simply as a placeholder for components for the Registration Wizard, you need only add a separator bar to save the menu.

10. Set the menu properties.
11. Save the menu definition.

### See Also

[Chapter 13, "Creating Menu Definitions," Setting Menu Item Properties, page 387](#)

[Chapter 13, "Creating Menu Definitions," Setting General Properties, page 388](#)

[Chapter 14, "Using the Registration Wizard," page 397](#)

[Chapter 2, "Using PeopleSoft Application Designer," Saving Definitions, page 31](#)

## Setting Menu Item Properties

Set the menu item properties to define your menu item. When creating a new menu definition to be used in the Registration Wizard, you need only select *Separator* from the Type area of the Menu Item Properties dialog box. This enables you to save the menu definition for use in the Registration Wizard. If you decide not to use the Registration Wizard you need to perform the entire following procedure and set the general and use properties for the menu definition.

To define a menu item:

1. In a standard menu definition, double-click a menu item to access its properties.

To create a new menu item, double-click the empty rectangle at the bottom of the menu.

2. Specify the menu item.

**Name** Specify the system name for the menu or menu item. Menu and menu item names must conform to a specific set of naming conventions for the system to recognize them: all uppercase letters, no embedded spaces, and no special characters (for example, %, ^, &, \*, \$, #).

**Label** Specify the text label to appear on the menu. The label appears in the Home bar item or in a cascade menu of a menu group. You can place an ampersand in the text to assign a keyboard shortcut for the menu.

3. Select the menu item type.

**Component** Select for a standard component menu item.

**PeopleCode** PeopleCode menu items trigger PeopleCode programs. The program runs in a component buffer context, which allows PeopleCode to access values in the component buffer.

**Separator** Select this option when creating a menu for the Registration Wizard. Selecting this option disables all other fields in the Menu Item Properties dialog box.

4. Click the Select button.
5. Select an enabling component in the Open dialog box.

At runtime standard PeopleCode menu items are always visible in a menu, but they are active (enabled) only when their enabling components are open.

6. Save the menu item if it has not yet been saved.
7. In the menu definition, right-click the menu item, then select *View PeopleCode*.

The PeopleCode editor appears.

8. Add a PeopleCode program in the menu item's ItemSelected event.

9. When you have finished typing the program, save the PeopleCode program and close the PeopleCode editor.
10. (Optional) Override the component search record.

You might want to reuse the same component multiple times with different search records. You can accomplish this by overriding the component search record at runtime when the component is opened from a menu item without creating separate copies of the component. The component override is temporary, and occurs only when the component is opened from the menu item in which the override is set. It does not change the component definition.

Select the Override check box, then select an override search record from the Override prompt field. The override search record must be a valid search record for the component; that is, it must have all of the search key fields that are at level 0 of the component pages.

11. Click OK to accept the settings.
12. Save the menu definition.

## Setting General Properties

When you create or make changes to a menu, it's a good idea to document it for future reference. You can do this on the General tab of the Menu Properties dialog box.

<b>Description</b>	Enter a descriptive name for the menu.
<b>Comments</b>	Add a description of the menu or any other useful information.
<b>Owner ID</b>	View a list of applications with which this menu is used. This list is helpful to identify the applications with which the menu is associated during application development.
<b>Last Updated</b>	View the date and time of the last modification and the name of the user who made the modification.

## Setting Use Properties

Setting properties on the Use tab of Menu Properties is optional. Any settings you assign are stored internally only.

---

## Working With Existing Menu Definitions

PeopleSoft recommends that you do not modify PeopleTools menus, but that you create your own custom menu.

This section discusses how to:

- Add components to a menu.

- Rename menu definitions.
- Copy a menu definition.
- Delete menu items.
- Print menu definitions.
- Set up menu security.
- Import menu groups into portals.

## Adding Components to a Menu

Use the Registration Wizard to add your component to a menu.

See [Chapter 14, "Using the Registration Wizard," page 397](#).

## Renaming Menu Definitions

When you change the name of the menu definition, you do not change the name of the application window that appears when you select the new menu from the PeopleSoft application.

To rename a menu definition:

1. Close all open definitions in the definition workspace.

2. Select File, Rename.

The Rename Definition dialog box appears.

3. Select *Menu* from the Definition Type drop-down list box.

4. Click Rename.

5. From the list of available menus, double-click the menu that you want to rename.

6. Type the new name over the name that is selected on the menu definition.

7. Click Rename.

---

**Note.** If you rename a menu definition, be sure to rename the corresponding registry entries manually.

---

### See Also

[Chapter 2, "Using PeopleSoft Application Designer," Saving Definitions, page 31](#)

## Copying a Menu Definition

Creating a copy of a menu definition is different from renaming the definition. The Save As command creates a copy of the definition with a new name while keeping the old menu definition unchanged.

To copy a menu definition:

1. Open the menu definition that you want to copy.
2. Select File, Save As.

The Save As dialog box appears.

3. Type a new name for the copy of the menu definition.
4. Click OK.

You are prompted to save a copy of any PeopleCode that you have associated with the menu definition.

## Deleting Menu Items

Delete an item to remove it permanently from a menu definition. To help prevent accidental deletions, you can delete only one entry at a time.

To delete a menu item:

1. Select the menu item that you want to delete.
2. Select Edit, Clear.

If you attempt to delete a menu item that is linked to PeopleCode, the system issues a warning.

If you proceed to delete it, the linked PeopleCode menu items are also deleted.

If you delete an item by mistake, before you perform any additional edits or saves, select Edit, Undo to restore the menu item.

---

**Note.** When deleting menu items, remember to delete any corresponding registry entries.

---

## Printing Menu Definitions

In addition to printing a menu definition from the definition workspace, you can obtain a formatted report of all menu definitions by printing the PeopleTools Menu Listing cross-reference report (XRFMENU). This report lists application windows in alphabetical order and details all menus in each window and all page definitions in each menu. It also includes the associated search record definition name and detail page definition name.

To print a menu definition:

1. Open the menu definition that you want to print so that it appears in the definition workspace as the current definition.
2. Select File, Print.

The standard Windows Print dialog box appears.

## Setting Up Menu Security

Whenever you make a change or add items to an existing custom menu definition, you might need to adjust your security settings. When you add a new menu to a security profile, that menu is available the next time that you sign in to the system. Use the Registration Wizard to assign security.

### See Also

[Chapter 14, "Using the Registration Wizard," page 397](#)

*Enterprise PeopleTools 8.50 PeopleBook: Security Administration*, "Understanding PeopleSoft Security"

## Importing Custom Menu Groups Into Portals

The menu import process creates content references for all components and pages that belong to the menu group. This should be used to migrate custom menus only.

### See Also

*Enterprise PeopleTools 8.50 PeopleBook: PeopleTools Portal Technologies*, "Configuring the Portal Environment," Importing Menu Groups into the Portal Registry

---

## Defining Pop-up Menus

This section provides an overview of pop-up menus and discusses how to:

- Create pop-up menus.
- Define transfer menu items.

## Understanding Pop-up Menus

Pop-up menus define the transition to another page—either a system-defined page containing a set of standard commands or a pop-up menu page that was created and associated with the pop-up button by an application developer.

You can also use pop-up menus to run a PeopleCode program. PeopleCode in pop-up menus does not share the same limitations as PeopleCode in standard menus, so pop-up transition menus provide an effective alternative to using command push buttons on pages. Pop-up menu PeopleCode programs can be used for any number of purposes, for example:

- To perform a modal transfer.
- To recalculate a field value.

- To trigger a PeopleSoft Workflow business event.

### Behavior of Pop-up Menus in Pages

These standard icons appear on the page at runtime when a pop-up menu is defined for a page field:



When a user clicks the pop-up icon (PT\_POPUP) on a page, a screen appears with a list of menu items from which to select.



This pop-up icon (PT\_POPUP\_SKIP) appears if a pop-up menu has only one menu item. When a user clicks this icon, the target transaction page appears immediately, skipping an intermediate page containing a list with one menu item.

For example, the Annual Benefits Base Rate field on the following page contains a pop-up icon, which opens the Display in Other Currency page.

Benefit Program Participation			
S,John N	Employee	EmplID: GP0017	Empl Rcd#: 0
Benefit Record Number: <input type="text"/>	Deductions Taken: NoOverride	Deduction Subset ID:	
Benefit Status <span style="float: right;">Find First 1 of 1 Last</span>			
Effective Date: 04/01/2002	Effective Sequence: 0		
Action / Reason: Hire	Hired from Affiliate	Current	
*Benefits System: <input type="text" value="Base Benefits"/>	Benefits Employee Status: Active		
Annual Benefits Base Rate: <input type="text"/>	INR		
Benefit Program Participation <span style="float: right;">Find   View All First 1 of 1 Last</span>			
*Effective Date <input type="text"/>	*Benefit Program <input type="text"/>	Currency Code	<input type="button" value="+"/> <input type="button" value="-"/>
<a href="#">Job Data</a> <a href="#">Employment Data</a> Benefits Program Participation			

Pop-up button on a page

This type of transfer to another component is called a *definitional transfer* to distinguish it from transfers that are made using PeopleCode programs. Definitional transfers are always nonmodal. To run a modal transfer, you must use PeopleCode.

## Creating Pop-up Menus

This section discusses how to:

- Create a new pop-up menu.
- Associate a pop-up menu with a page field.

### ***Creating a New Pop-up Menu***

To create a new pop-up menu:

1. Select File, New.

The New dialog box appears.

2. Select Menu.

3. Click OK.

The New Menu dialog box appears.

4. Select the Popup option for the menu type.

5. Click OK.

A new pop-up menu definition appears.

6. Define the menu items for the pop-up menu.

7. Set the menu properties.

- a. Add a description and comments about your pop-up menu.

- b. On the Use tab, type the menu label that you want to appear for that pop-up.

8. Save the pop-up menu definition.

9. To provide users with access to the pop-up menu, associate it with a page field.

### ***Associating a Pop-up Menu With a Page Field***

To associate a pop-up menu with a page field:

1. Open the page definition that contains the page field to which you want to associate the pop-up menu.
2. Right-click the page field, then select Page Field Properties.
3. When the properties dialog box for that field appears, select the Use tab.
4. In the Popup Menu group box, select from the available pop-up menus in the drop-down list box.
5. Click OK.
6. Save the page definition.

### ***See Also***

Chapter 13, "Creating Menu Definitions," Setting Menu Item Properties, page 387

## Defining Transfer Menu Items

Transfer menu items, which can be created only in pop-up menus, enable you to specify a menu, component, and page to which the user is taken when the transfer menu item is selected. The following procedure describes how to create a new transfer menu item or redefine an existing menu item as a transfer menu item.

To define a transfer menu item:

1. In a pop-up menu definition, double-click a menu item to access its properties.

To create a menu item, double-click the empty rectangle at the bottom of the menu. This displays the Menu Item Properties dialog box for pop-up menus.

2. If this is a new menu item, enter a menu item name and label text in the Menu Item fields.

You can create a shortcut key for the menu item by placing an ampersand in the label.

3. Set the transfer properties.

- Click the Transfer button to make this a transfer menu item.
- Click the Define Transfer button to display the Transfer Properties dialog box.

This dialog box enables you to define the properties of a definitional transfer that is made from a pop-up menu item. The component from which you are transferring is called the *originating component*. The component to which you are transferring is called the *destination component*.

## 4. Specify the following:

<b>Portal</b>	Select the navigation registry, such as employee portal.
<b>Node</b>	Select the PeopleSoft database to which you want the transfer menu to point. <i>Local</i> is the default. Valid values include all of the values in the MSGNODENAME column of the PSMSGNODEDEFN table. If you select a node other than <i>Local</i> , the system clears the Use data from current page in search dialog check box.
<b>Menu</b>	Select the name of the menu containing the destination component.
<b>Component</b>	Select the name of the destination component.
<b>Market</b>	Select the market of the destination component.
<b>Component Item</b>	Select the name of the page to display at the front of the destination component.
<b>Action</b>	<p>Select an action mode (<i>Update/Display</i>, <i>Update/Display All</i>, <i>Correction</i>, or <i>Data Entry</i>) to constrain the user profile to a specific action mode at the time of the transfer. The action modes that are available in the prompt list depend on which modes are specified as valid in the destination component definition.</p> <p>Select <i>Prompt</i> to enable the user profile to select an available action mode at the time of the transfer. The action modes that the user profile can select depend on which action modes are defined as valid in the destination component definition and to which action modes the user profile is granted access in Maintain Security.</p>

## 5. Set the transfer type.

<b>Re-use current window</b>	Select to transfer to the component in the same window in which the transfer is initiated.
<b>Use new window (minimize current window)</b>	Select to minimize the current window and open the destination component in a separate window.

## 6. Specify whether to supply search key values from the originating component.

<b>Use data from current page in search dialog</b>	<p>Select to use the values from the originating component search keys in the search dialog box of the destination component. If these values allow the system to select a unique row from the search record, the search dialog box is bypassed and the destination component opens immediately.</p> <p>Clear this check box if you want the user to type the search key values into the search dialog box.</p>
--	---

## 7. Click OK to accept the Transfer Properties dialog box settings.

## 8. Click OK to accept the Menu Item Properties dialog box settings.

9. Save the menu definition.

## Chapter 14

# Using the Registration Wizard

This chapter provides an overview of the Registration Wizard and describes how to:

- Register a component or iScript.
- Register the component in multiple portals.
- Register a mobile page.

---

## Understanding the Registration Wizard

This section lists common elements and discusses the Registration Wizard.

### Common Elements Used in This Chapter

<b>content reference</b>	A reference to a URL (Uniform Resource Locator) for a transaction page. Appears as a link in the navigation menu at runtime.
<b>iScript</b>	A specialized PeopleCode function that generates dynamic web content. <i>See Enterprise PeopleTools 8.50 PeopleBook: PeopleCode API Reference, "Internet Script Classes (iScript)."</i>
<b>permission list</b>	A set of access rights to application objects, including processes and reports, weblib functions, and other elements. <i>See Enterprise PeopleTools 8.50 PeopleBook: Security Administration, "Setting Up Permission Lists."</i>
<b>portal registry</b>	A tree structure in which content for a portal is organized, classified, and registered. A portal registry consists of folders, content references, and nodes. <i>See Enterprise PeopleTools 8.50 PeopleBook: PeopleTools Portal Technologies, "Understanding Portal Technology," Portal Registry.</i>

## Working With the Registration Wizard

After you create your pages and assign them to a component, you can register that component to display the transaction page in the browser. You can do this using the Registration Wizard. The wizard gathers information, then attaches the component to a menu, assigns a permission list to allow security access rights, and places a content reference entry in the portal registry, which displays the component to the user on the navigation menu.

---

**Note.** Before running the Registration Wizard, the menu definition, permission list, and folder must exist. The wizard does not create these definitions. Instead it matches existing definitions to one another and creates the content reference.

If change control is enabled while you are using the Registration Wizard you will only be allowed to associate components with locked menus.

---



---

## Registering Components and iScripts

In Application Designer, open the component definition that you want to register. If you are creating a new component, make sure that you save it before opening the Registration Wizard. For iScripts, open the record definition that holds the iScript function. By convention, such records begin with the WEBLIB\_ prefix.

You can access the Registration Wizard using any of these methods:



Click the Register Component or Register iScript icon in the toolbar

**Pop-up menu**

Select Register Component from the pop-up menu after right clicking on the component.

**Tools menu**

Select Register Component or Register iScript from the Tools menu.

---

**Note.** All settings in the Registration Wizard default to those referenced in the previous wizard session.

---

To register a component or iScript:

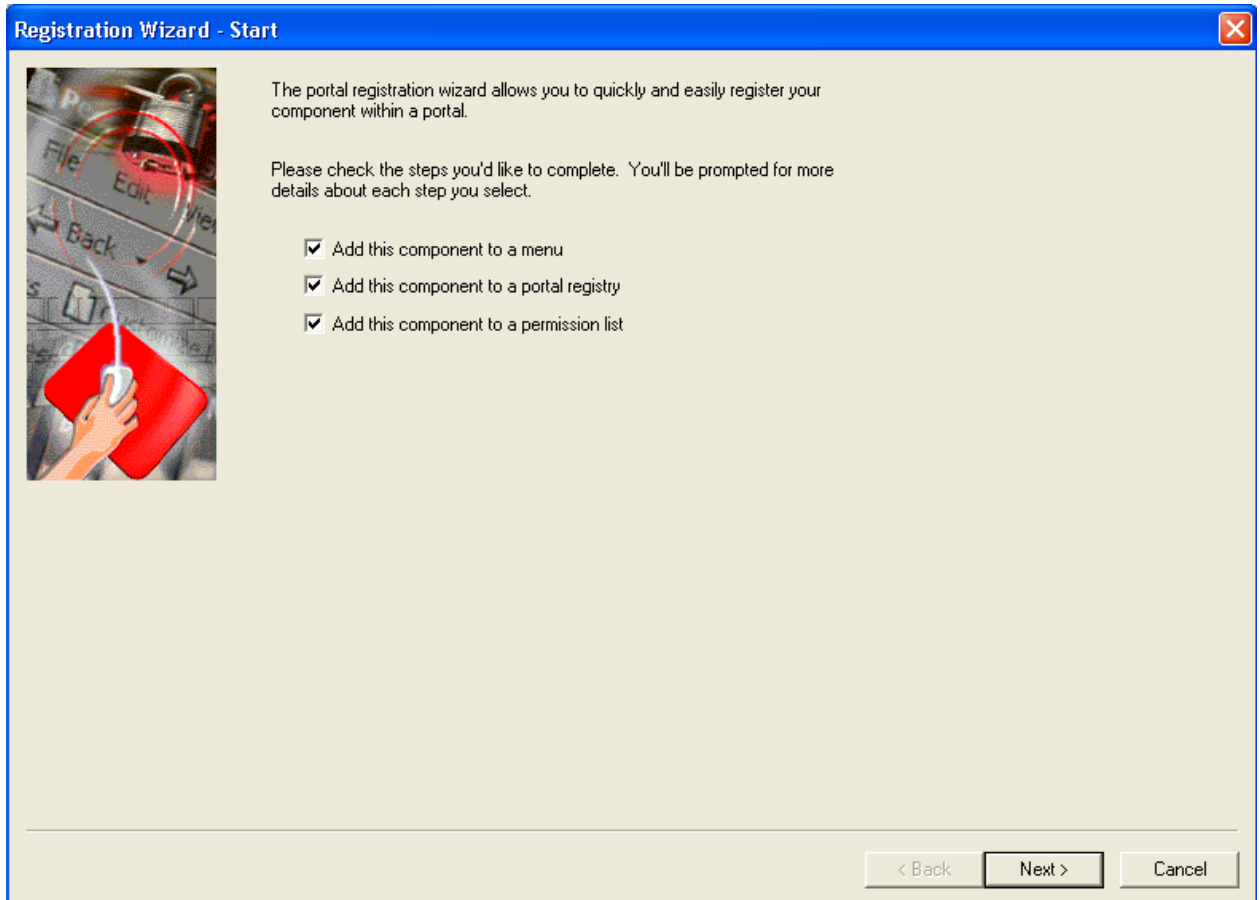
1. Open the component or iScript you want to register.

You can also launch the Registration Wizard from the menu definition by right clicking on a menu item that points to a component.

## 2. Open the Registration Wizard.

Select Tools, Register in Portal or click the icon on the Toolbar.

The Start screen of the Registration Wizard gives the option of adding the component to a menu, a portal registry, and/or a permission list. If this is a new component and the first time that you are running the wizard, you should select all three options. If you have already run the wizard on this particular component and just need to add it to an additional permission list or portal, deselect the other options.



Registration Wizard – Start page

**Add this component to a menu** Select to add the component to a menu. You must create the menu definition first if it does not already exist. You specify the menu on the following screen of the Registration Wizard. If you do not select this option, you will need to add the component to the menu manually using the Menu designer.

See [Chapter 13, "Creating Menu Definitions," Creating Custom Component Menus, page 386.](#)

**Add this component to a portal registry** Select if you want to create a content reference for the component and adds it to the portal registry you specify on the Create Content Reference screen of the Registration Wizard.

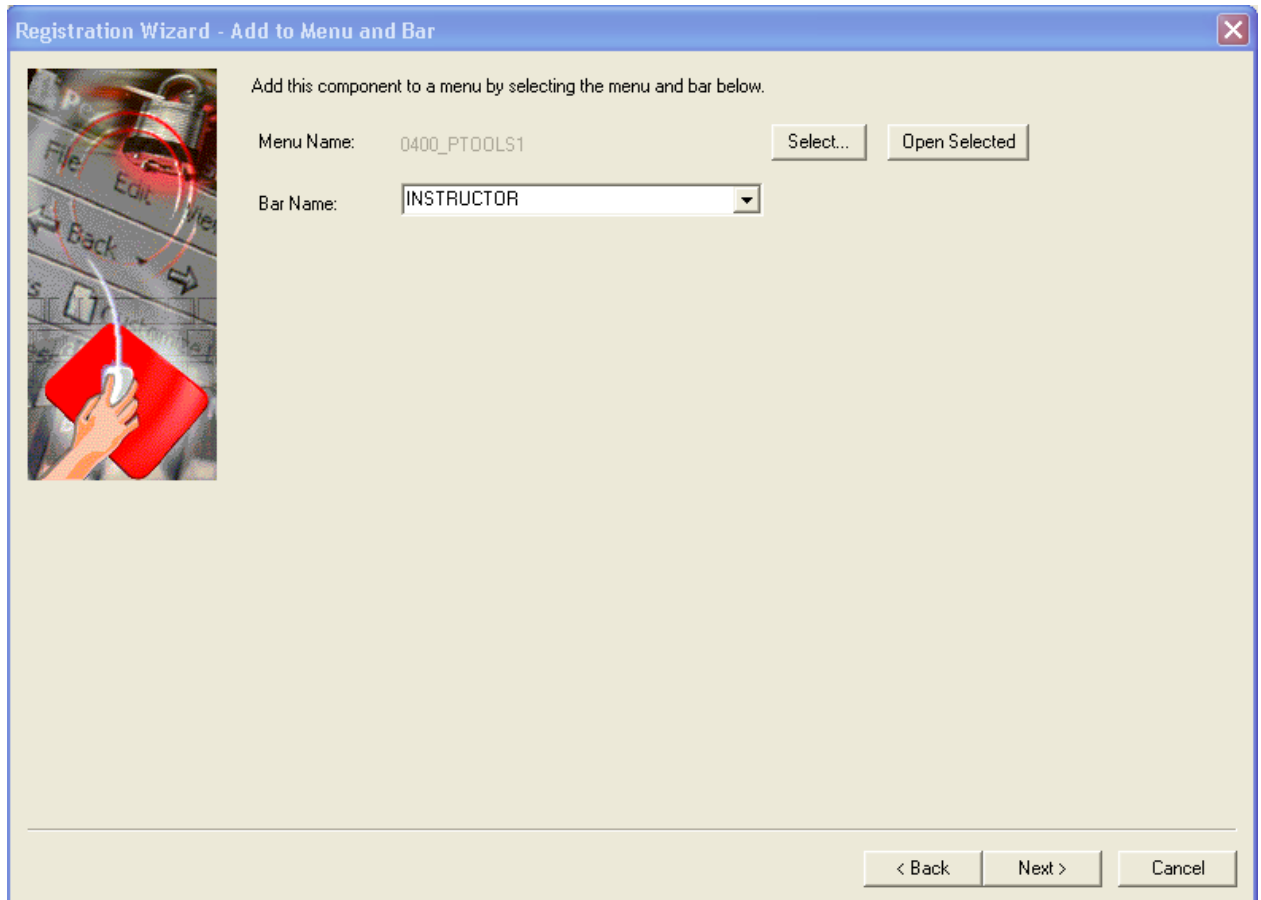
**Add this component to a permission list** Select if you want to specify the permission list on the final screen of the Registration Wizard.

If this is a new component or iScript and the first time that you've run the Registration Wizard, you will most likely want all three options selected. If you are rerunning the wizard to simply add the component to an additional permission list or another portal, you would clear the other options.

The wizard remembers the selection that you made the last time that you ran the wizard and selects those values by default.

3. Click the Next button.

If you selected to add the component to a menu, the next screen, Add to Menu and Bar, opens.

The image shows a Windows-style dialog box titled "Registration Wizard - Add to Menu and Bar". On the left is a small graphic of a hand holding a red mouse button over a menu bar. The main area contains the text "Add this component to a menu by selecting the menu and bar below." Below this, there is a "Menu Name:" label followed by a text box containing "0400\_PT00LS1" and a "Select..." button. To the right of the "Menu Name" section is an "Open Selected" button. Below the "Menu Name" section is a "Bar Name:" label followed by a drop-down menu currently showing "INSTRUCTOR". At the bottom right of the dialog are three buttons: "< Back", "Next >", and "Cancel".

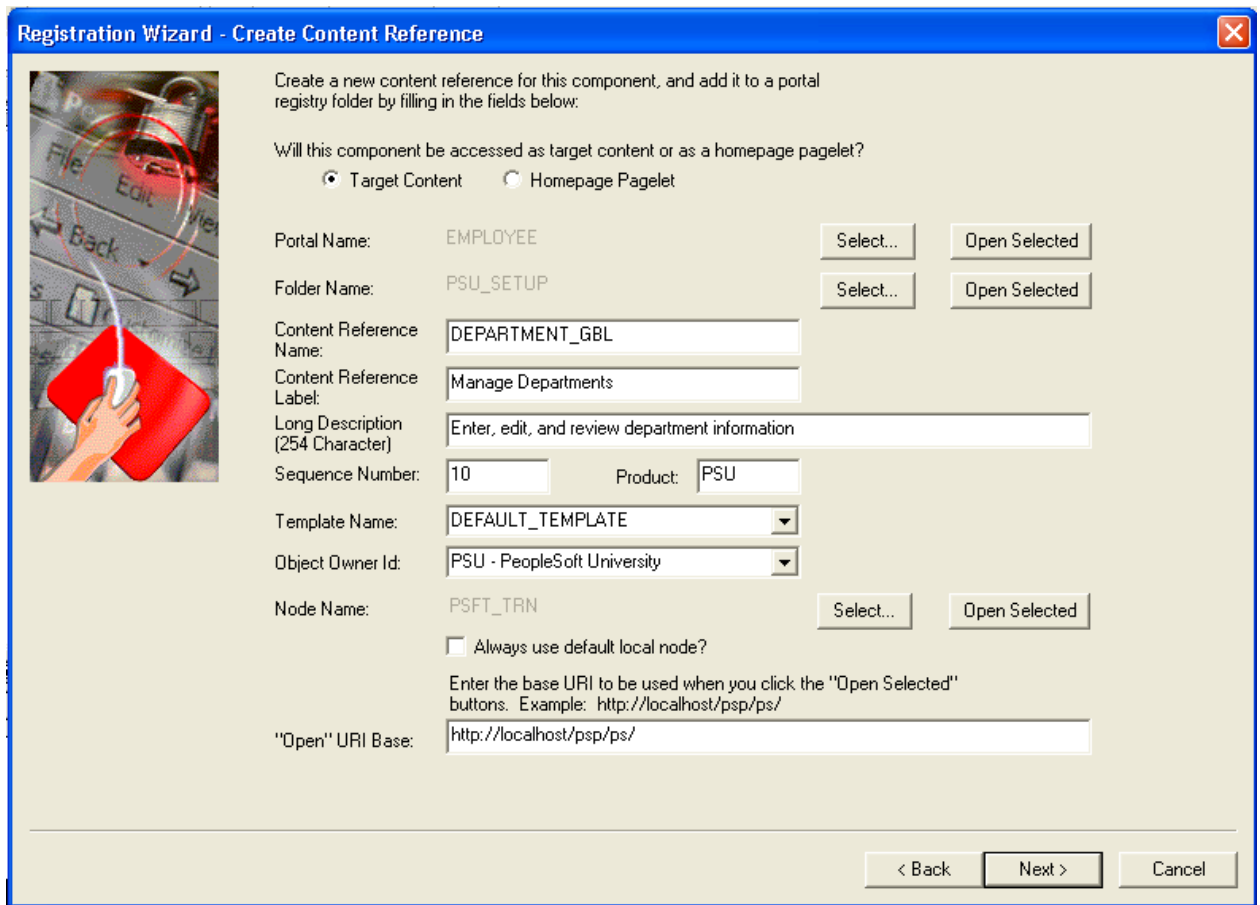
Registration Wizard – Add to Menu and Bar page

4. Click the Select button to find the menu name and then click the Bar Name drop-down list box to assign the bar name, then click the Next button.

These names are not exposed to users, but are required for internal location purposes. Click the Open Selected button to open the menu that you selected in Application Designer. To view the menu definition without closing the wizard, drag the Registration Wizard to the side.

- On the Create Content Reference page, select Target Content or Homepage Pagelet.

If the component is a standard transaction, register it as Target Content.



**Registration Wizard - Create Content Reference**

Create a new content reference for this component, and add it to a portal registry folder by filling in the fields below:

Will this component be accessed as target content or as a homepage pagelet?

☒ Target Content    ☐ Homepage Pagelet

Portal Name: EMPLOYEE    Select...    Open Selected

Folder Name: PSU\_SETUP    Select...    Open Selected

Content Reference Name: DEPARTMENT\_GBL

Content Reference Label: Manage Departments

Long Description (254 Character): Enter, edit, and review department information

Sequence Number: 10    Product: PSU

Template Name: DEFAULT\_TEMPLATE

Object Owner Id: PSU - PeopleSoft University

Node Name: PSFT\_TRN    Select...    Open Selected

☐ Always use default local node?

Enter the base URI to be used when you click the "Open Selected" buttons. Example: http://localhost/ps/ps/

"Open" URI Base: http://localhost/ps/ps/

< Back    Next >    Cancel

Registration Wizard – Create Content Reference page

## 6. Complete all fields in the Create Content Reference page.

<b>Target Content</b>	Select this check box to register all forms of target content (excluding homepage).
<b>Homepage Pagelet</b>	Select to register a new homepage pagelet. The homepage template will be applied automatically.
<b>Portal Name and Folder Name</b>	Click the Select button to change the portal or folder name. Click the Open Selected button to launch a browser that will take you to the administration page to create a new folder if necessary. The default values are the last ones referenced by the wizard.
<b>Content Reference Name</b>	The name is automatically generated, including the market code as the suffix.
<b>Content Reference Label</b>	This is the CREF hyperlink that the user sees as the registry entry.
<b>Long Description</b>	Enter the hover text for the CREF link.
<b>Sequence Number</b>	Enter a numeral to control the order in which the component appears within the chosen folder.
<b>Product</b>	Enter the 2-character product code.
<b>Template Name</b>	Specify the template to be used from the drop-down menu.  If you specified that the target content should be accessed as a homepage pagelet, then this field will not be displayed, and the homepage template will be applied automatically.
<b>Object Owner Id</b>	Select the application development group that owns the object.
<b>Node Name</b>	Click Select to change the Node name, or click the Open Selected button to launch the administration page to view the properties of the node.
<b>Always use default local node</b>	Select to always override the selected node name with the specified default local node.
<b>"Open" URI Base</b>	Enter the Universal Resource Identifier (URI) that specifies for PeopleSoft Application Designer the location of your PeopleSoft Pure Internet Architecture site. This is for informational purposes only. It is used only for the Open Selected button and not stored to the database.  If you entered a base URI on the Browser tab of the Options dialog box, that value displays in this field.  The URI you enter in this field is reflected on the remaining pages of the Registration Wizard. If you make a change to this field on any page, the change is reflected throughout.

## 7. Click the Next button.

8. You see the Add to Permission List page.

**Registration Wizard - Add to Permission List**

This component is an entry in the following permission lists:

Name: Description:

-- No Permission Lists contain this object

Open List Selection

Add this component to a permission list by selecting the permission list below:

Permission List Name: ALLPAGES

Select...

Open Selected

Add all pages of this component with the following actions:

☒ Add

☒ Update / Display

☒ Update / Display All

☒ Correction

Enter the base URI to be used when you click the "Open Selected" buttons. Example: http://localhost/ps/ps/

"Open" URI Base: http://localhost/ps/ps/

< Back Next > Cancel

#### Registration Wizard – Add to Permission List page

Click the Select button to find the menu name, then use the drop-down list box to select the name that displays on the bar.

Menus are used as logical groupings to which you can apply security. PeopleSoft components are addressed by menu name (the URL includes the menu name). To add this component to a menu, specify the menu name and bar name.

If you did not select the Add this component to a menu check box on the Prompt page, this page displays the menu name and the Select button only. You then must select the correct menu that already contains the component.

If you are working with an iScript, you must select the iScript function to register.

9. Click the Next button.
10. Select Target Content if your component is a standard transaction and complete the appropriate fields.

The previous term table defines each of these fields.

If you did not select Add this Component to a Portal Registry on the Prompt page, this page does not appear.

11. Click the Next button.

12. Select the appropriate permission list, select all the actions that you want to grant this component or iScript, and enter the base URI.

Use the Select button to view and select a permission list.

If you did not select Add this Component to a Permission List, this page does not appear.

13. Click the Next button.

14. Use the Finish page to review the selections made so far. If you need to change anything, use the Back button to edit your entries.

Select the options in the Add to project area to place your item into an active PeopleTools project. This is cumulative behavior—the more times you run the wizard while the project is active, the more menus, registry structures, and permission lists are added to that project.

15. Click Finish to complete the registry process into the specified portal.

If you click Cancel all of the entries that you have made in the wizard will be lost and no changes will be made.

### **See Also**

[Chapter 13, "Creating Menu Definitions," page 385](#)

*Enterprise PeopleTools 8.50 PeopleBook: Security Administration, "Setting Up Permission Lists"*

---

## **Registering a Component in Multiple Portals**

After you have used the Registration Wizard to register your new component into the portal you may need to register the component in other portals also.

There are two ways to do this.

- You can use the Portal Registration Wizard again. Select only the second option: Add this component to a portal registry. Then reenter the content reference label, long description, and sequence number (if not 1) to match the other portal's entry.
- You can just copy the component to other portals from portal Administration pages in the web client.

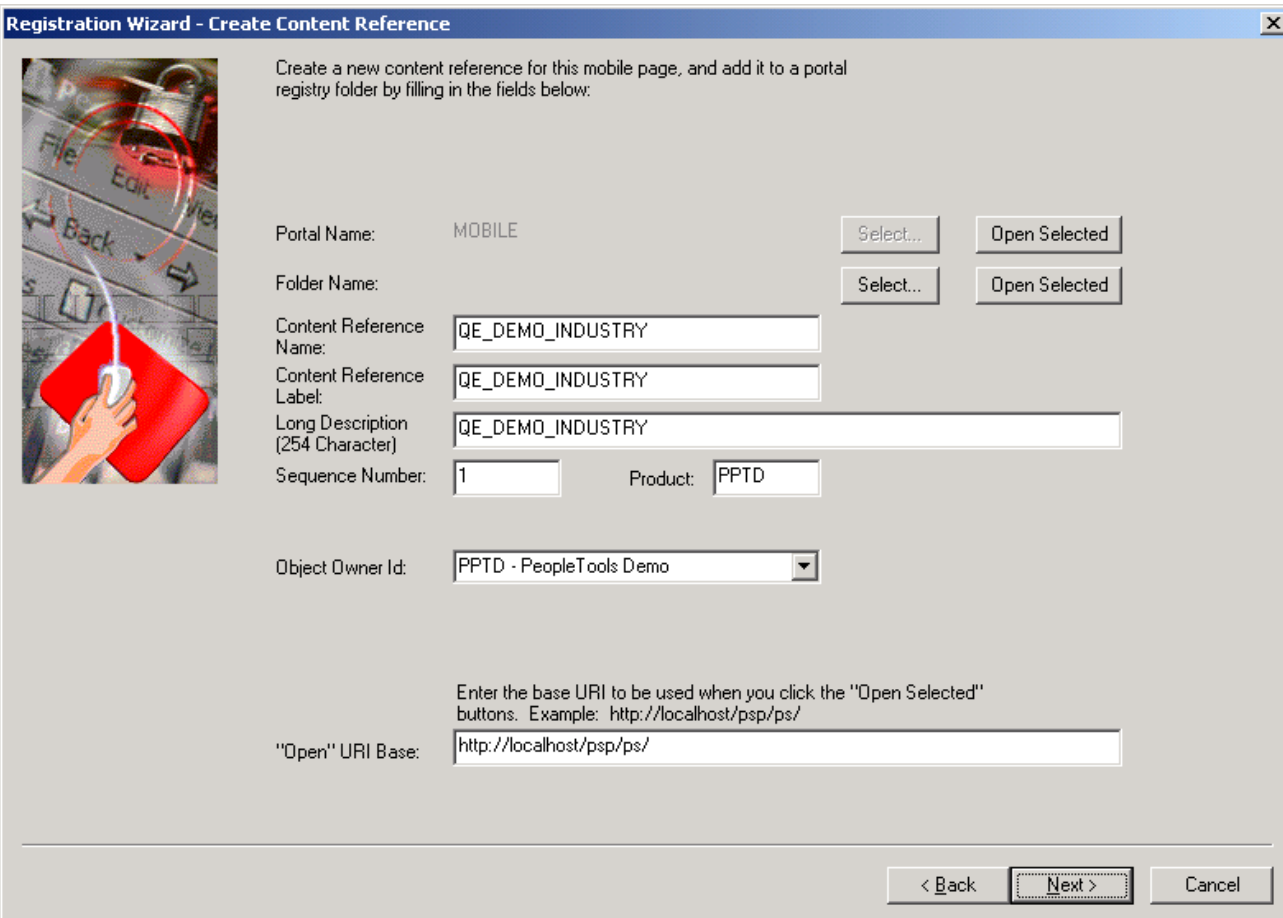
See *Enterprise PeopleTools 8.50 PeopleBook: PeopleTools Portal Technologies*.

---

## **Registering a Mobile Page**

As with a component or iScript, you can also register a mobile page in the portal using the Registration Wizard, though the wizard is slightly different. This wizard always assigns the mobile page to the MOBILE portal and does not require a node name.

**Important!** PeopleSoft Mobile Agent is a deprecated product. These features exist for backward compatibility only.



**Registration Wizard - Create Content Reference**

Create a new content reference for this mobile page, and add it to a portal registry folder by filling in the fields below:

Portal Name: MOBILE Select... Open Selected

Folder Name: Select... Open Selected

Content Reference Name: QE\_DEMO\_INDUSTRY

Content Reference Label: QE\_DEMO\_INDUSTRY

Long Description (254 Character): QE\_DEMO\_INDUSTRY

Sequence Number: 1 Product: PPTD

Object Owner Id: PPTD - PeopleTools Demo

Enter the base URI to be used when you click the "Open Selected" buttons. Example: http://localhost/ps/ps/  
 "Open" URI Base: http://localhost/ps/ps/

< Back Next > Cancel

### Registration Wizard for mobile page

To register a mobile page in the portal:

1. Open the mobile page that you want to register.
2. Open the Registration Wizard.

Use any one of three methods to do this:

- Click the Registration Wizard icon in the toolbar.
- Select Tools, Register Mobile Page from the menu.
- Select Register from the pop-up menu.

3. Set the Folder Name and content reference information.
4. Assign the Sequence Number and Product.
5. Select the object owner ID from the drop-down list box.

6. Click Next.

The final screen of the wizard opens enabling you to verify the settings that you have selected. The registration wizard also adds your registry entry to the current project. The Registry entry check box is selected by default.

7. Review your settings in the output window and click Finish.

## Chapter 15

# Creating Style Sheet Definitions

This chapter provides an overview of style sheets and style classes and discusses how to:

- Create standard and sub style sheets.
- Set style class attributes.
- Work with free form sub style sheets.
- Work with default style sheets.
- Override default page style properties.
- Modify page tabs.
- Modify push buttons in SWAN style sheets.

---

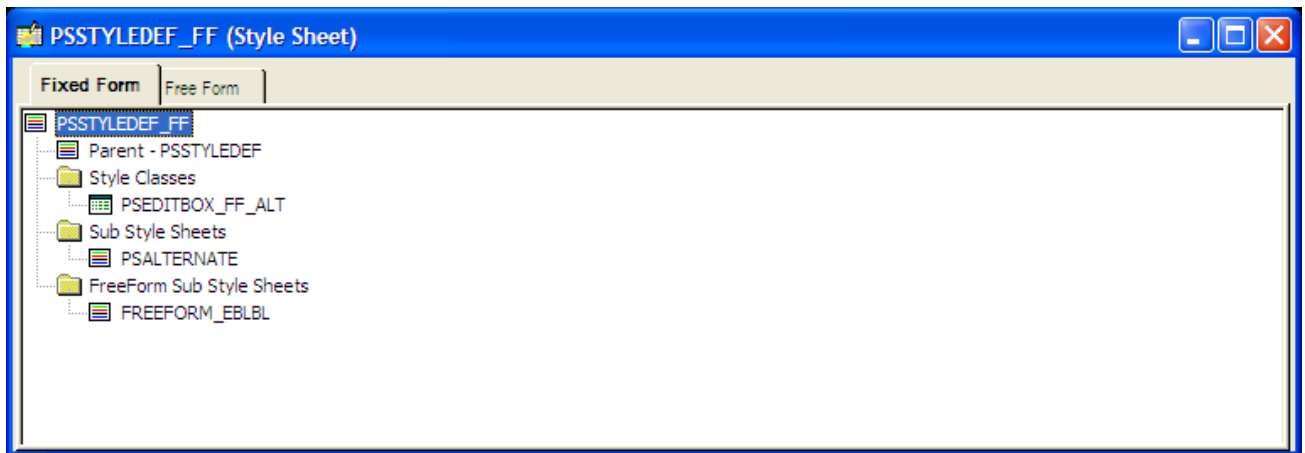
## Understanding Style Sheets and Style Classes

Style sheets are definitions that store margins, spacing, fonts, borders, color and other layout settings for a particular element of a page. When you use a style sheet, the system applies these predefined layout settings or *style classes* to every page in an application. By storing all page layout settings in a style sheet definition, you eliminate the need to redefine the same settings for each page separately, you can quickly reconfigure and apply new settings, and you maintain a consistent appearance throughout the application.

In PeopleSoft applications, a style sheet is a definition, like a record or page definition, that you create and maintain in PeopleSoft Application Designer. A style sheet is a standalone definition that can be upgraded, deleted, renamed, and so on.

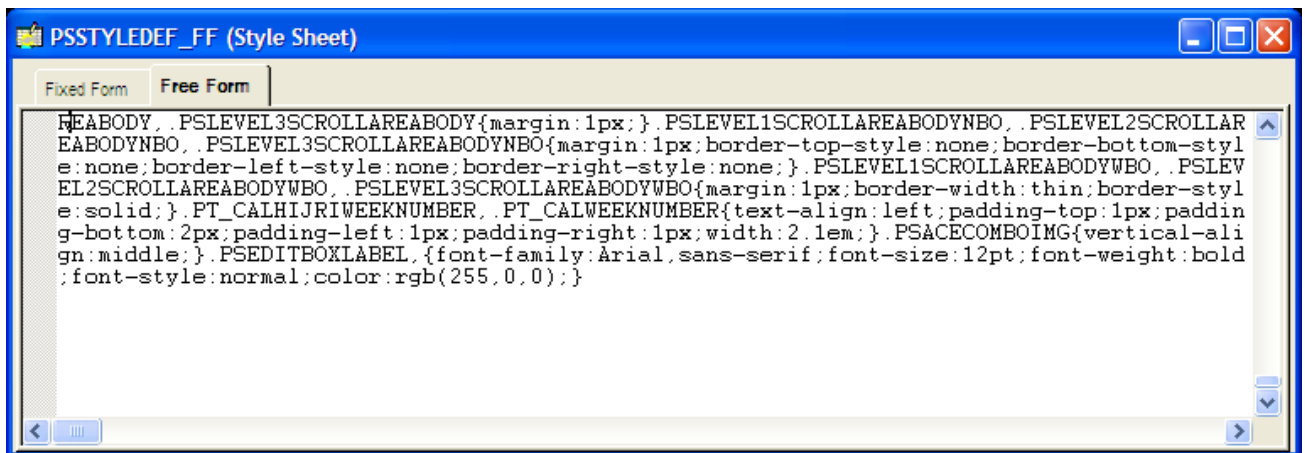
You define style sheets by using the Fixed Form and Free Form tabs.

In this example, you see the Fixed Form tab, which you access to insert style sheets, sub style sheets, free form sub style sheets, or style classes into the definition.



Style sheet definition showing the Fixed Form tab

In this example, you see the Free Form tab, which you access to enter text when you create a free form sub style sheet. For standard style sheets and sub style sheets, use this view to examine or copy the text that is generated by the style sheet definition.



Style sheet definition showing the Free Form tab

---

**Note.** For standard style sheets and sub style sheets, the Free Form tab text is *read-only*.

---

This section discusses:

- Style sheet types.
- Style classes.

## Style Sheet Types

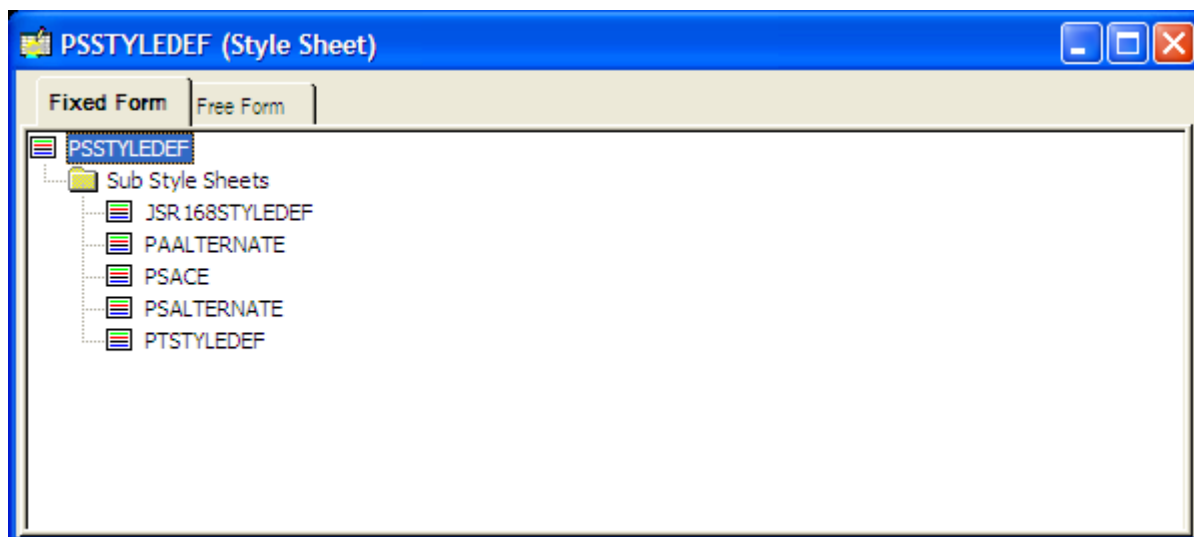
You can create three style sheet types using Application Designer:

- Standard Style Sheet
- Sub Style Sheet

- Free Form Sub Style Sheet

### ***Standard Style Sheet***

This is an example of a standard style sheet.



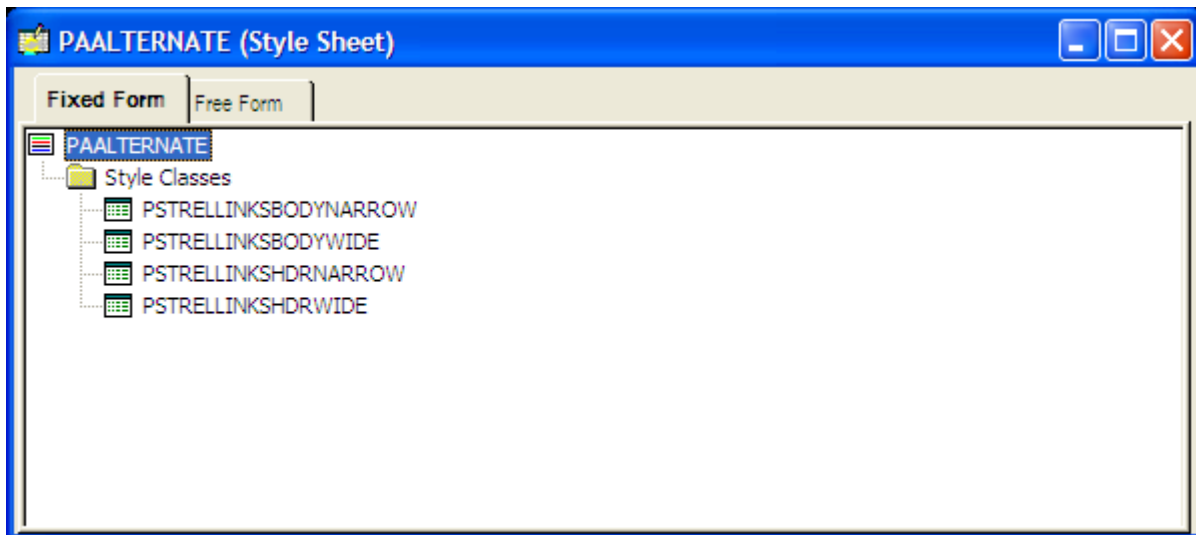
#### Standard Style Sheet

Standard style sheets are the primary organizers for working with style sheets and style classes. Standard style sheets are containers that can include sub style sheets, free form sub style sheets, and style classes in any combination; however standard style sheets cannot contain other standard style sheets.

Standard style sheets can inherit style classes from other standard style sheet definitions through the Parent Style Sheet property. The parent style sheet propagates all of its style classes to the child style sheet. Only standard style sheets can have parent style sheets, and be used as parent style sheets.

### ***Sub Style Sheet***

This is an example of a sub style sheet.



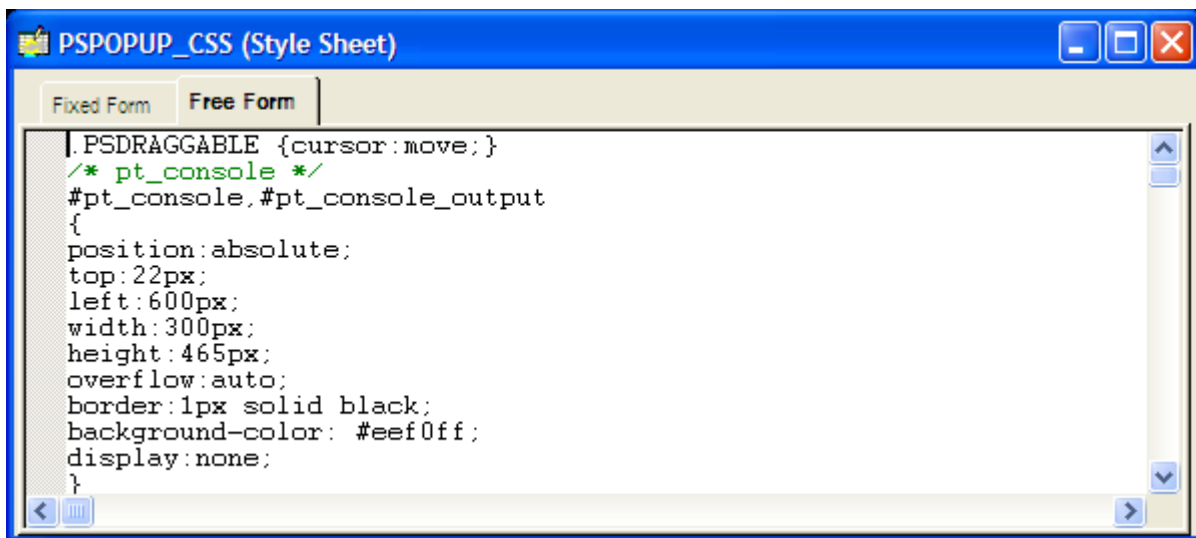
### Sub Style Sheet

In the same way that PeopleTools supports subrecords, you can define *sub style sheets*. The primary purpose of sub style sheets is to organize and reuse style classes. You can insert sub style sheets in standard style sheets as well as into other sub style sheets. Sub style sheets neither have parent style sheets nor can be used as parent style sheets.

To use sub style sheets, you must insert them into standard style sheets.

### Free Form Sub Style Sheet

This is an example of a free form sub style sheet:



### Free Form Sub Style Sheet

Free form sub style sheets are text-based sub style sheets that enable you to take advantage of Cascading Style Sheets— Level 2 (CSS2), AJAX, and DHTML features. With a free form sub style sheet, you can create the style sheet in a third-party editor and then copy the style sheet text into the Free Form tab of the free form sub style sheet. On free form sub style sheets, the fixed form tab is grayed out because free form sub style sheets do not support inheritance or aggregation. Free form sub style sheets cannot have parent style sheets nor can they be used as parent style sheets.

Note these important points regarding the use of free form sub style sheets:

- Like other sub style sheets, to use free form sub style sheets, you must insert them in a standard style sheet or in a sub style sheet that is part of a standard style sheet.
- When creating free form style sheets, style class names must be identical to the PeopleTools default style class names.

If you replace the default style sheet or override the page style sheet with a style sheet that includes only free form text, the free form sub style sheet must define and include *all* default style classes used by the application. Unexpected results will occur in both Application Designer and PeopleSoft Internet Architecture pages if a default style class is missing.

See [Chapter 15, "Creating Style Sheet Definitions," Default Style Classes, page 412](#).

- At runtime, style classes in free form sub style sheets take precedence over PeopleTools-formatted style classes of the same name.

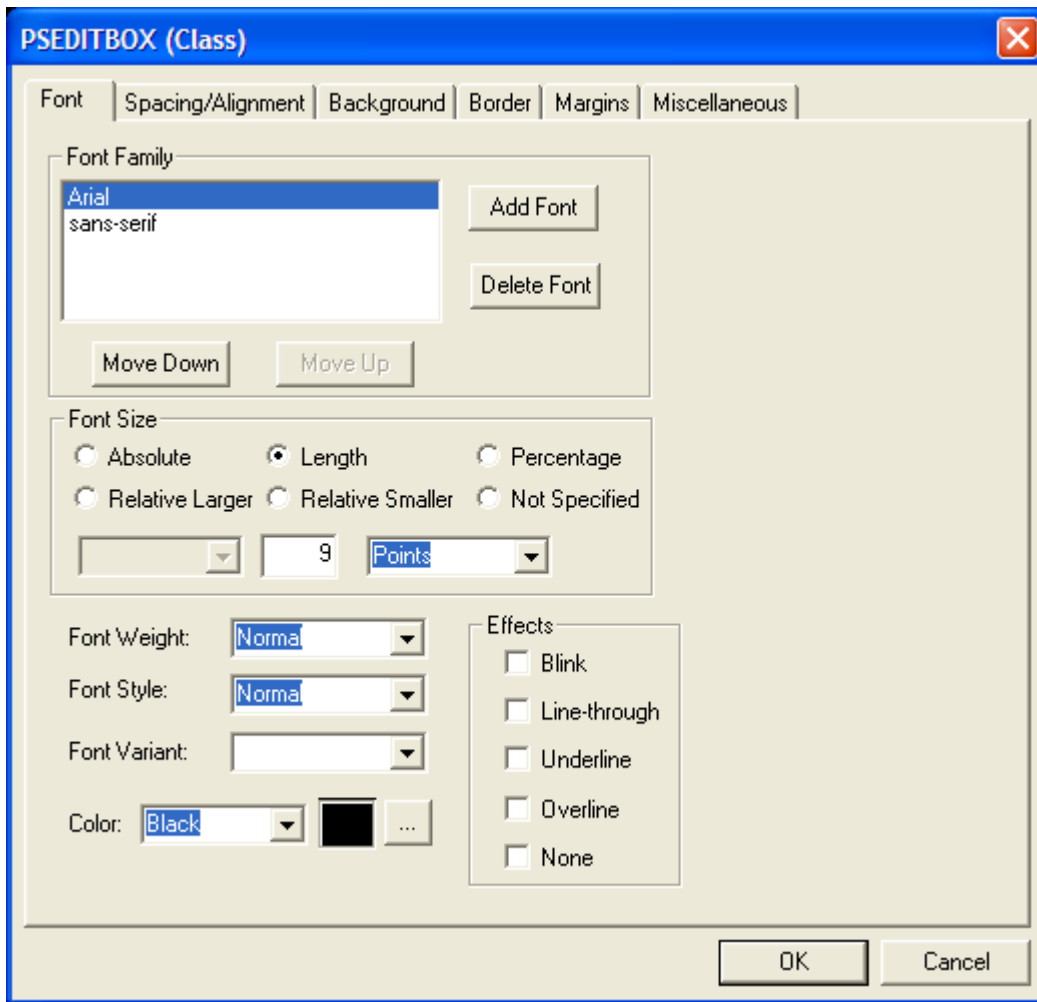
---

**Note.** Free form sub style sheets are supported in PeopleTools release 8.50 and higher. If you attempt to import a free form sub style sheet into an earlier release, the free form sub style sheet imports as an empty standard style sheet.

---

## Style Classes

Access a style class dialog box:



The PSEDITBOX style class properties dialog box

Standard style sheets and sub style sheets comprise standard *style classes*. The style class determines the visual properties of pages and page controls by controlling a broad range of characteristics, including font size, spacing, alignment, border width, weight, and color.

Style classes in different style sheets are independent of each other. You can copy existing style classes and paste, or insert, them into any style sheet. However, the most efficient way to share style classes across multiple style sheets is to group style classes into sub style sheets.

### Default Style Classes

Default style classes are the PeopleTools style classes that the system explicitly associates with a page element, such as the page, each page control type, page control labels, and so on. PeopleSoft applications use the default style classes to render the page and all page elements at design time and at runtime, unless you specify a style property override.

See [Chapter 15, "Creating Style Sheet Definitions," Overriding Default Page Style Properties, page 440.](#)

Note these points about default style classes:

- If you change an attribute of a default style class, the change affects *every* occurrence of the page element throughout an application, except the instances that do not use the default style.

- The application style sheet (or any page level override) must include *all* default style classes for the system to render pages correctly at design time and at runtime. The default style classes can be accounted for in standard or free form format in any combination. However, unexpected results will occur in both Application Designer and PeopleSoft Internet Architecture pages if a default style class is missing.

---

**Note.** PeopleTools style sheets include all default style classes. It is recommended that you do not modify any PeopleTools style sheets or the attributes of default style classes.

---

Most default style class names enable you to easily identify the page element with which they are associated. For example, PSEDITBOX is the name of the default style class for all edit box page controls. When you examine a specific edit box control, the value in the Style field, however, is not the *name* of the default style class. The Style field value of *\*\*\*Use Default Value\*\*\** identifies any page element using its default style class.

This PeopleBook includes an appendix that lists and describes the default style classes.

See [Appendix C, "Using Default Style Classes," Default Style Classes, page 553](#).

### **Custom Style Classes**

This list includes some examples of custom style classes:

- Creating new PeopleTools style classes, inserting them into standard style sheets, and using them as style overrides to create unique pages in an application.
- Creating and inserting override style classes which supplant style classes inherited from a parent style sheet.
- Overriding page style classes with non-default, existing PeopleTools style classes.
- Creating free form style sheets that reconfigure one or more default style classes.

---

## **Creating Standard and Sub Style Sheets**

This section discusses how to:

- Create standard style sheets or sub style sheets.
- Rename standard or sub style sheets.
- Delete standard or sub style sheets.
- Insert sub style sheets into style sheets.
- Insert style classes into style sheets.
- Inserting override style classes into style sheets.

### **Creating Standard Style Sheets or Sub Style Sheets**

To create a new standard style sheet or sub style sheet:

1. Select File, New.
2. Select *Style Sheet* from the Definition list and click the OK button.
3. Select the style sheet type.
4. Click the OK button.
5. Select File, Definition Properties.
6. Select the style sheet type and parent style sheet.

The Parent Style Sheet field is enabled if you create a standard style sheet only. The default parent style sheet is the application style sheet that you set on the PeopleTools Options page.

7. Enter a description.
8. Save the style sheet.

## Renaming Standard Style Sheets or Sub Style Sheets

To rename standard or sub style sheets:

1. Select File, Rename.
2. Select *Style Sheet* in the Definition drop-down list box and then enter search criteria to find the style sheet definition that you are deleting.
3. Click the Rename button; the definitions that match the search criteria appear in the list box.
4. Double-click the style sheet definition that you would like to rename.
5. Enter the new name of the style sheet definition in the highlighted area and press the Enter key.
6. Click the OK button to confirm the rename.

---

**Note.** When you rename a standard or sub style sheet, Application Designer automatically renames all references to the style sheet.

---

## Deleting Standard Style Sheets or Sub Style Sheets

To delete standard or sub style sheets:

1. Select File, Delete.
2. Select *Style Sheet* in the Definition drop-down list box and then enter search criteria to find the style sheet definition that you are deleting.
3. Press the Delete key; the definitions that match the search criteria appear in the list box.
4. Click the style sheet that you want to delete and press the Delete key.
5. Click the OK button to confirm the deletion or click the Cancel button to cancel the operation.

---

**Note.** When you delete a style sheet, Application Designer automatically deletes all references to the deleted style sheet. If you delete a style sheet that contains sub style sheets, the sub style sheets are *not* deleted; they become independent definitions.

---

## Inserting Sub Style Sheets into Style Sheets

You can insert a sub style sheet into a standard style sheet or another sub style sheet.

To insert a sub style sheet into style sheets:

1. Create or open the receiving style sheet.
2. Select Insert, Insert Sub Style Sheet.
3. In the drop-down list boxes, select the sub style sheet type and the sub style sheet that you want to insert.
4. Click the OK button.
5. Save the sub style sheet.

You can remove sub style sheets from within other style sheets by highlighting the sub style sheet and pressing the Delete key. This action does *not* delete the definition from the database; it removes the sub style sheet from the current definition only.

## Inserting Style Classes into Style Sheets

You can insert style classes into any standard style sheet or sub style sheet.

To insert a style class into a style sheet:

1. Open the style sheet in which you want to create the new style class.
2. Select Insert, Insert Style Class.
3. Enter the name of the new class.
4. Copy attributes from another style class by selecting the style class name from the Copy Values From Style Class drop-down list box.

Leave the Copy Values From Style Class field empty if you want to set all style class properties manually.

5. Click the OK button.
6. Configure style class attributes, as necessary.

See [Chapter 15, "Creating Style Sheet Definitions," Setting Style Class Attributes, page 416.](#)

You can remove a style class from a style sheet by highlighting the class and pressing the Delete key. This action *does* delete this instance of the style class from the database.

---

**Note.** For efficiency, clone style classes from other style sheets. Then, either use the Insert menu or paste the style class into the new style sheet. At that time, the system enables you to enter the new style class name. Be aware that when copying style classes, the copy or clone will not contain any of the related language data associated with the original.

---

## Inserting Override Style Classes into Style Sheets

When you create a style sheet and configure the Parent Style Sheet property, the new style sheet inherits all of the style classes of the parent. If you want to change the attributes of an inherited default style class in the new style sheet, you insert an override style class. To insert an override style class into a style sheet:

1. Open the style sheet.
2. Verify that the Parent Style Sheet property is configured correctly.
3. Select Insert, Override Style Class.
4. Select the style class name from the Style Class drop-down list box and click the OK button.

The class properties dialog box appears showing the old settings.

5. Configure the override style class as necessary.

See [Chapter 15, "Creating Style Sheet Definitions," Setting Style Class Attributes, page 416.](#)

6. Click the OK button.
7. Save the style sheet definition.

---

## Setting Style Class Attributes

This section provides overviews of class attributes and fonts and discusses how to:

- Access class attributes.
- Specify fonts.
- Set font attributes for a specific language.
- Specify spacing and alignment.
- Specify background attributes.
- Reference a background image URL.
- Specify border attributes.
- Specify margins.
- Specify miscellaneous attributes.

## Understanding Class Attributes

Class attributes are grouped into categories and are displayed in a tabbed dialog box for each class.

---

**Note.** PeopleSoft Application Designer does not render certain style class formatting attributes at design time. You might be able to see the attribute at runtime only. Also, not all browsers support every style class attribute.

---

PeopleSoft Pure Internet Architecture supports the attributes that are listed in the World Wide Web Consortium (W3), Cascading Style Sheets, Level 1.

See <http://www.w3.org/TR/REC-CSS1>.

Every class has each of these major attributes, which correspond to a tab in the Class dialog box.

<b>Font</b>	Controls the font in which text appears. If more than one font is specified, the browser displays the first font; if it cannot find it, it displays the next font.
<b>Spacing/Alignment</b>	Controls the spacing in addition to the default spacing. You can specify additional spacing between words and between letters. You can also specify the alignment, height, indent, and white space.
<b>Background</b>	Controls the background colors for the page control. If the background is an image, you can select how the image appears.
<b>Border</b>	Controls the border that is displayed around the element. You can specify width, color, and style.
<b>Margins</b>	Controls the margins and padding on each side of the element.
<b>Miscellaneous</b>	Controls the display of list-item markers, URLs, and cursor formats.

## Understanding Generic and Installed Fonts

When you add a font, the dialog box provides a choice between *generic* fonts and *installed* fonts.

Generic fonts can be displayed on all browsers. This table displays examples of some of the generic fonts:

<b><i>Generic Font</i></b>	<b><i>Example</i></b>
Cursive	Zapf-Chancery
Fantasy	Western
Monospace	Courier

<b><i>Generic Font</i></b>	<b><i>Example</i></b>
Sans serif	Helvetica
Serif	Times Roman

### ***Installed Fonts***

Installed fonts are installed on your computer. If you use an installed font for a style class, PeopleSoft recommends that you:

- Verify that *all* users of your application have the same installed font.
- Specify the installed font first, followed by a generic font.

This way, if the installed font is not available to the user of your application, the generic font can be used.

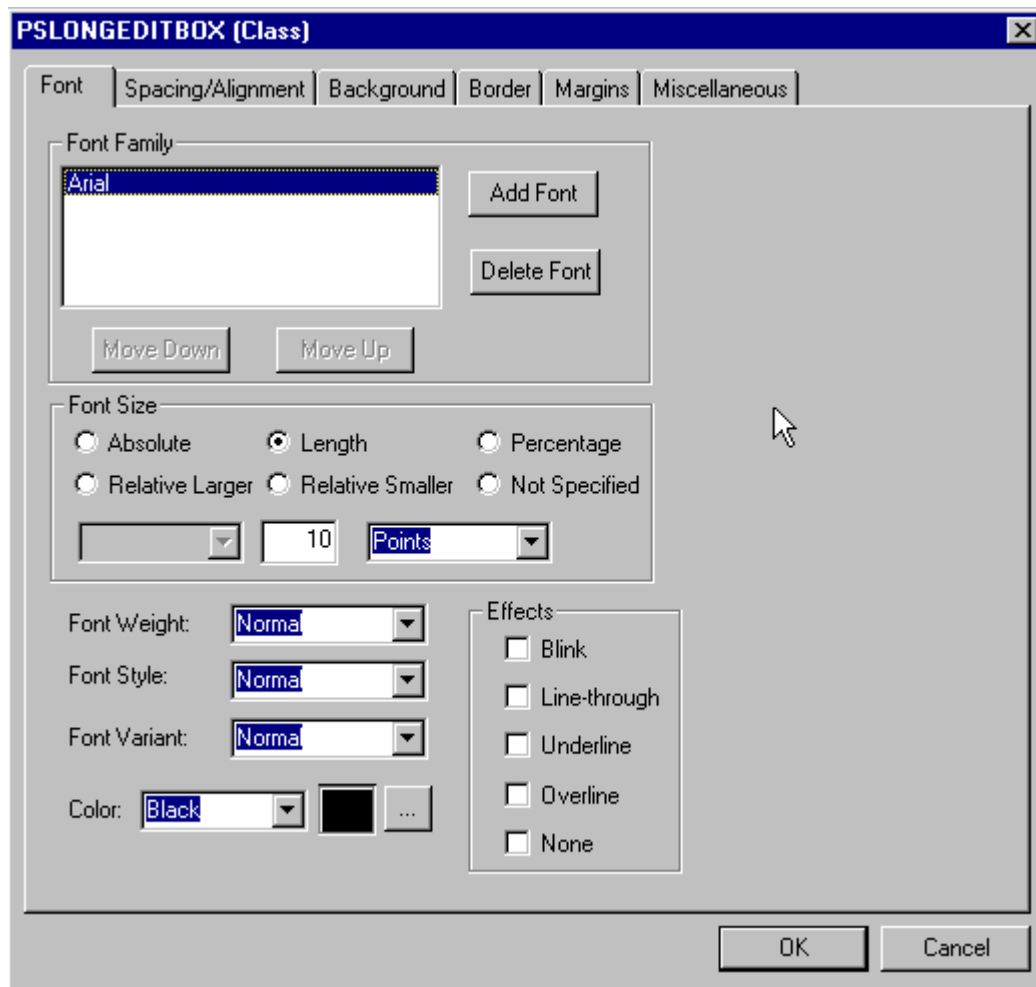
## **Accessing Class Attributes**

To access class attributes:

1. Open a style sheet.
2. Double-click the class name.

## **Specifying Fonts**

Access the class dialog box. Select the Font tab.



PSLONGEDITBOX (Class) dialog box: Font tab

### Font Family

Specify the font in which you want the text to appear. You can specify more than one font, so that if the browser cannot display the first font, it attempts to display the next font, and so on, until it finds a font that it can use.

### Font Size

#### Absolute

Select to use a size from the drop-down list box (*xx-small*, *x-small*, *small*, *medium*, and so on.) There is a scaling factor of 1.5 between adjacent sizes.

#### Length

Specify the size with a number, and select the measurement type from the drop-down list box (*em*, *ex*, *Pixels*, *Inches*, *Millimeters*, and so on).

#### Percentage

Specify a number, followed by a percent sign (%). Percentage values are always relative to the parent element.

<b>Relative Larger and Relative Smaller</b>	Select to interpret the font relative to the table of font sizes and the font size of the parent element. For example, if the parent element has a font size of medium, and you select Relative Larger, this font size is large.
<b>Font Weight</b>	Select the weight of the font. The values <i>100</i> to <i>900</i> form an ordered sequence, in which each number indicates a weight that is at least as dark as its predecessor. The value <i>Normal</i> is synonymous with <i>400</i> , and <i>Bold</i> is synonymous with <i>700</i> .
<b>Font Style</b>	Select the style of the font. Values are: <i>Normal</i> , <i>Italic</i> , or <i>Oblique</i> . These are matched to existing fonts with those names.
<b>Font Variant</b>	Select a variant. Values are: <i>Normal</i> or <i>Small Caps</i> . In a small caps font, the lowercase letters look similar to the uppercase ones, but they're in a smaller size and with slightly different proportions.
<b>Color</b>	Specify the color or select the color from a color palette.

## Setting Font Attributes for a Specific Language

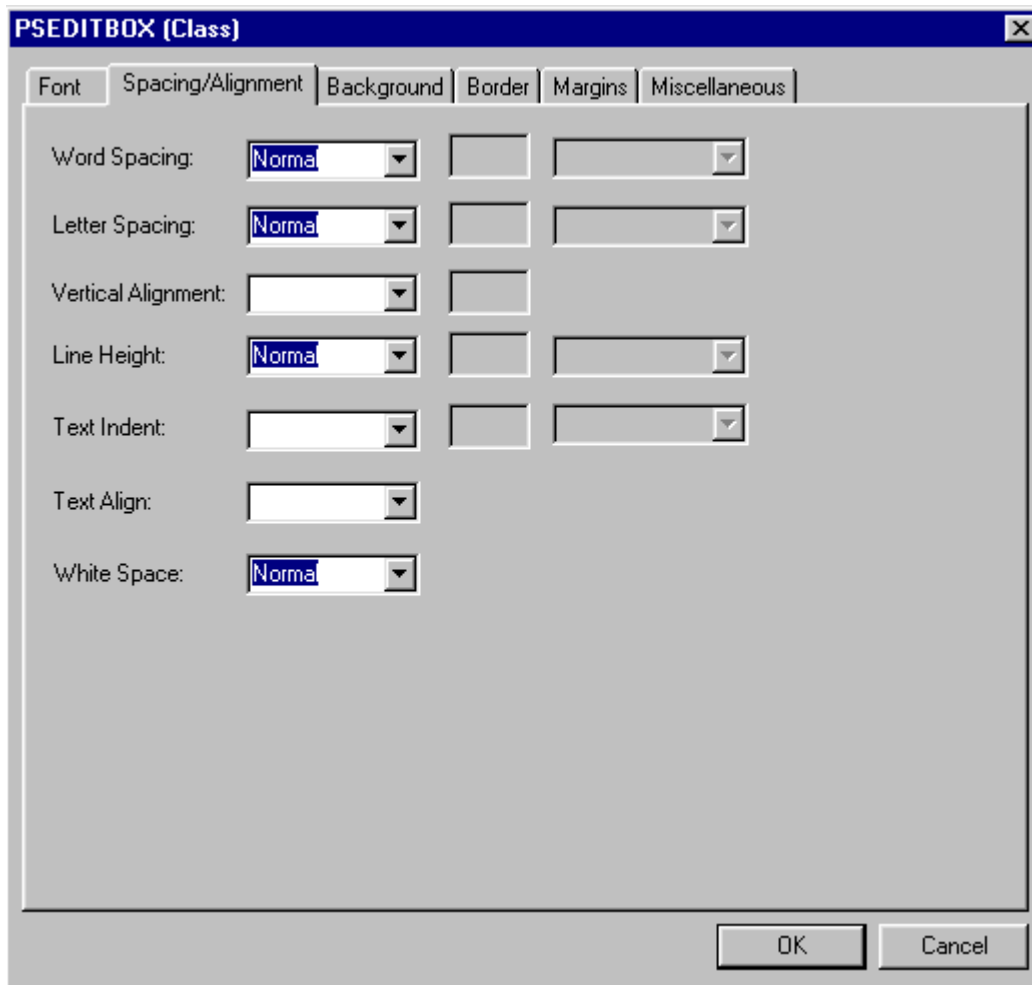
You can change the fonts that are available based on the base language settings. The default for the base language is English, providing generic fonts. If these fonts are not appropriate for a particular language, you must set the language preference before opening a style sheet. You can then set the font name list, font size, or font weight for the selected language.

To set font attributes for a specific language:

1. Select PeopleTools, Utilities, International, Preferences from your browser.
2. Select the appropriate language for the style sheet from the drop-down list box.
3. Save the new selected language as the base language.
4. Open the style sheet definition in which you want to save the font name list, font size, or font weight for the selected language.

## Specifying Spacing and Alignment

Access the class dialog box. Select the Spacing/Alignment tab.



PSEDITBOX (Class) dialog box: Spacing/Alignment tab

### **Word Spacing and Letter Spacing**

Select *Normal* or *Length*. If you select *Length*, specify a number to indicate an addition to the default space between words or letters. Values can be negative, but there might be implementation-specific limits.

**Vertical Alignment**

Select the vertical positioning of the element. Values are:

*Baseline:* Aligns the baseline of the element (or the bottom, if the element doesn't have a baseline) with the baseline of the parent.

*Sub:* (subscript) Aligns the top of the element with the baseline of the parent element.

*Super:* (superscript) Aligns the baseline of the element with the top of the parent element.

*Top:* Aligns the top of the element with the tallest element on the line.

*Text Top:* Aligns the top of the element with the top of the parent elements font.

*Middle:* Aligns the vertical midpoint of the element (typically an image) with the baseline plus half of the x-height of the parent.

*Bottom:* Aligns the bottom of the element with the lowest element on the line.

*Text Bottom:* Aligns the bottom of the element with the bottom of the parent elements font.

*Percentage:* Refers to the value of the line-height property of the element itself. It raises the baseline of the element (or the bottom, if it has no baseline) to the specified amount above the baseline of the parent. Negative values are possible. For example, a value of  $-100\%$  lowers the element so that the baseline of the element is where the baseline of the next line should have been. This allows precise control over the vertical position of elements (such as images that are used in place of letters) that don't have a baseline.

---

**Note.** Using the top and bottom alignments could cause a loop where there are element dependencies.

---

**Line Height**

Set the distance between the baselines of two adjacent lines. When a numerical value is specified, the line height is provided by the font size of the current element multiplied with the numerical value. This differs from a percentage value in the way it inherits. When a numerical value is specified, child elements inherit the factor itself, not the resultant value (as is the case with percentage and other units). Negative values are not allowed.

**Text Indent**

Specify the indentation that appears before the first formatted line. This value may be negative, but there might be implementation-specific limits. An indentation is not inserted in the middle of an element that was broken by another.

**Text Align**

Specify how text is aligned in the element. Because Text Align inherits, all block-level elements inside the DIV element with CLASS=center are centered.

---

**Note.** Alignments are relative to the width of the element, not the canvas.

---

**White Space**

Specify how to handle white space inside the element. Values are:

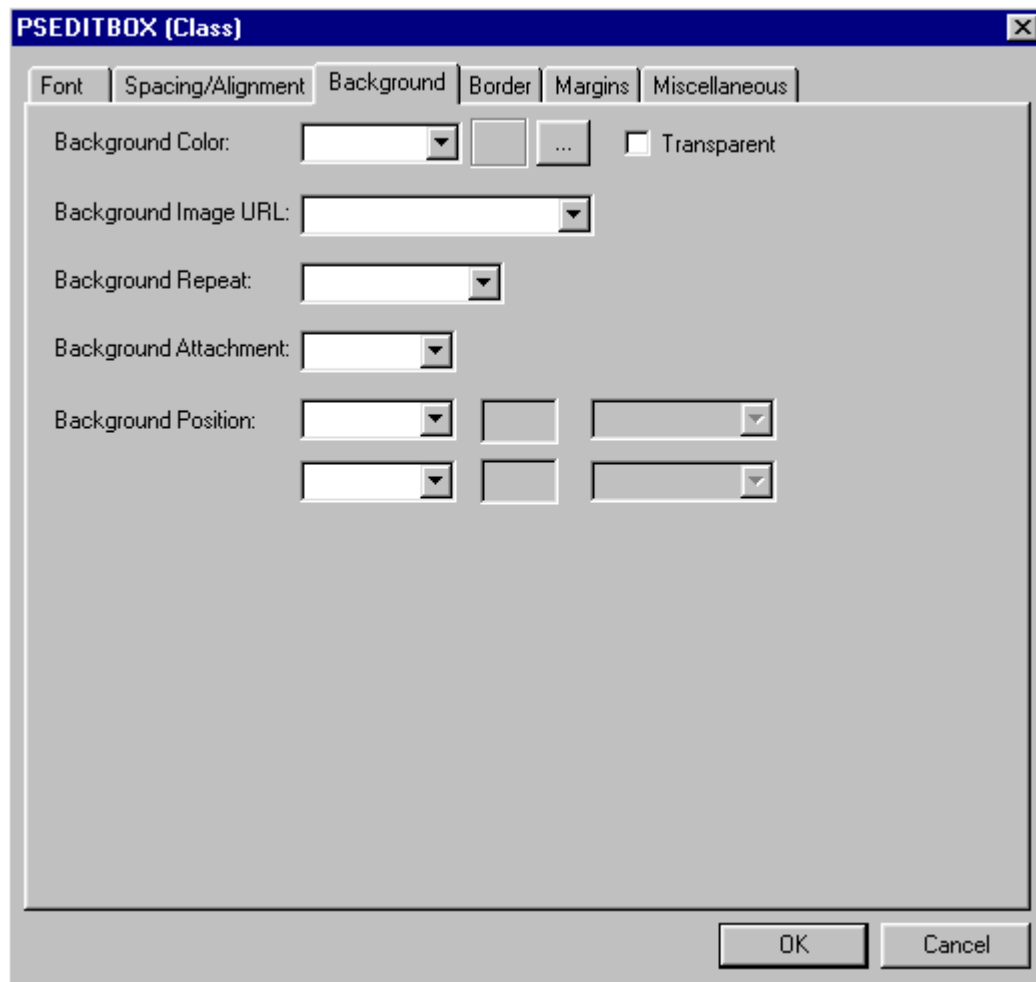
*Normal:* White space is collapsed.

*Pre:* Behaves like the PRE element in HTML.

*Nowrap:* Wrapping is done only through elements having a line break.

## Specifying Background Attributes

Access the class dialog box. Select the Background tab.



PSEDITBOX (Class) dialog box: Background tab

**Background Color** Specify the background color of an element.

**Background Image URL** Set the background image of an element to an image file that is previously defined. When you set a background image, also set a background color to use when the image is unavailable. When the image *is* available, it appears on top of the background color.

See [Chapter 15, "Creating Style Sheet Definitions," Referencing a Background Image URL, page 425.](#)

**Background Repeat** If you specify a background image, indicate whether the image is repeated, and if it is, how many times.

**Background Attachment** If you specify a background image, indicate whether it is fixed to the canvas or whether it scrolls along with the content.

**Background Position** If you specify a background image, set its initial position. Values are:

*Top*: Aligns the background to the top of the element.

*Left*: Aligns the background to the left of the element.

*Center*: Aligns the background to the midpoint of the element (typically an image).

*Bottom*: Aligns the background to the bottom of the element.

*Right*: Aligns the background to the right of the element.

*Length*: Specify the length of the background and the unit of measure.

*Percentage*: Refers to the value of the line-height property of the background itself. It raises the baseline of the element (or the bottom, if it has no baseline) the specified amount above the baseline of the parent. Negative values are possible.

For example, a value of *-100%* lowers the element so that the baseline of the element is where the baseline of the next line should have been. This allows precise control over the vertical position of elements without a baseline, such as images used in place of letters.

### **Examples of Percentage and Length Value Pairs and Keywords**

This table lists background positions, value pairs, and their effect:

<b>Background Position</b>	<b>Value Pairs</b>	<b>Effect</b>
Percentage	0 and 0	Places the upper, left-hand corner of the image in the upper, left-hand corner of the box that surrounds the content of the element (that is, not the box that surrounds the padding, border, or margin).
Percentage	100 and 100	Places the lower, right-hand corner of the image in the lower, right-hand corner of the element
Percentage	14 and 84	Places the point that is 14 percent across and 84 percent down the image at the point that is 14 percent across and 84 percent down the element.
Length	2cm and 2cm	Places the upper, left-hand corner of the image 2 centimeters (cm) to the right and 2 cm below the upper, left-hand corner of the element.

<b>Background Position</b>	<b>Value Pairs</b>	<b>Effect</b>
Top Left or Left Top	NA	The same as 0 percent and 0 percent.
Top Center or Center Top	NA	The same as 50 percent and 0 percent.
Right Top or Top Right	NA	The same as 100 percent and 0 percent.
Left Center or Center Left	NA	The same as 0 percent and 50 percent.
Center and Center	NA	The same as 50 percent and 50 percent.
Right Center or Center Right	NA	The same as 100 percent and 50 percent.
Bottom Left or Left Bottom	NA	The same as 0 percent and 100 percent.
Bottom Center or Center Bottom	NA	The same as 50 percent and 100 percent.
Bottom Right or Right Bottom	NA	The same as 100 percent and 100 percent.

---

**Note.** If you set only *one* percentage or length value, the system sets the horizontal position only and the vertical position becomes the default 50 percent. If you set two values, the first one is the horizontal position. You can set negative positions and combinations of length and percentage values, for example 50 percent and 2 cm.

---

## Referencing a Background Image URL

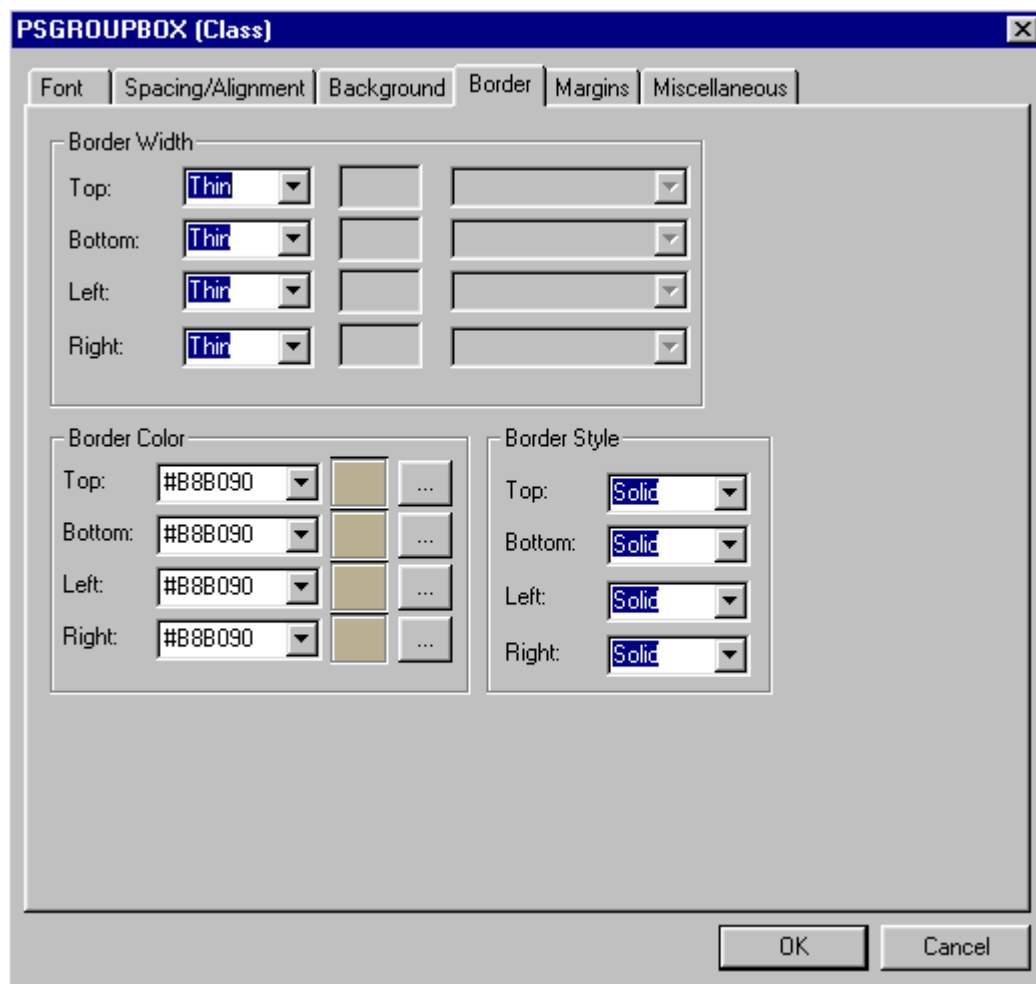
To specify a background image URL, select from a drop-down list box that is populated by entries in the URL Maintenance table. To reference a background image file on a designated server, add its location to the URL Maintenance table.

You reference a background image in the URL Maintenance table in the same way you would register any other URL.

See *Enterprise PeopleTools 8.50 PeopleBook: System and Server Administration*, "Using PeopleTools Utilities," URL Maintenance.

## Specifying Border Attributes

Access the class dialog box. Select the Border tab.



PSGROUPBOX (Class) dialog box: Border tab

### Border Width

Specify the width of each line in the border. Values are: *Thin*, *Medium*, or *Thick*, or you can specify *Length*, which enables you to specify a font size (with a number) and the measurement type from the drop-down list box (*Pixels*, *Inches*, *Millimeters*, and so on).

### Border Color

Specify the color of each line in the border. You can specify a hexadecimal value for the color or select the color from a color palette.

## Border Style

Specify the style of line in the border. Values are:

*None:* No border is drawn (regardless of the `<border-width>` value).

*Dotted:* Border is a dotted line that is drawn on top of the background of the element.

*Dashed:* Border is a dashed line that is drawn on top of the background of the element.

*Solid:* Border is a solid line.

*Double:* Border is a double line that is drawn on top of the background of the element. The sum of the two single lines and the space between equals the `<border-width>` value.

*Groove:* Border is a three-dimensional groove that is drawn in colors based on the `<color>` value.

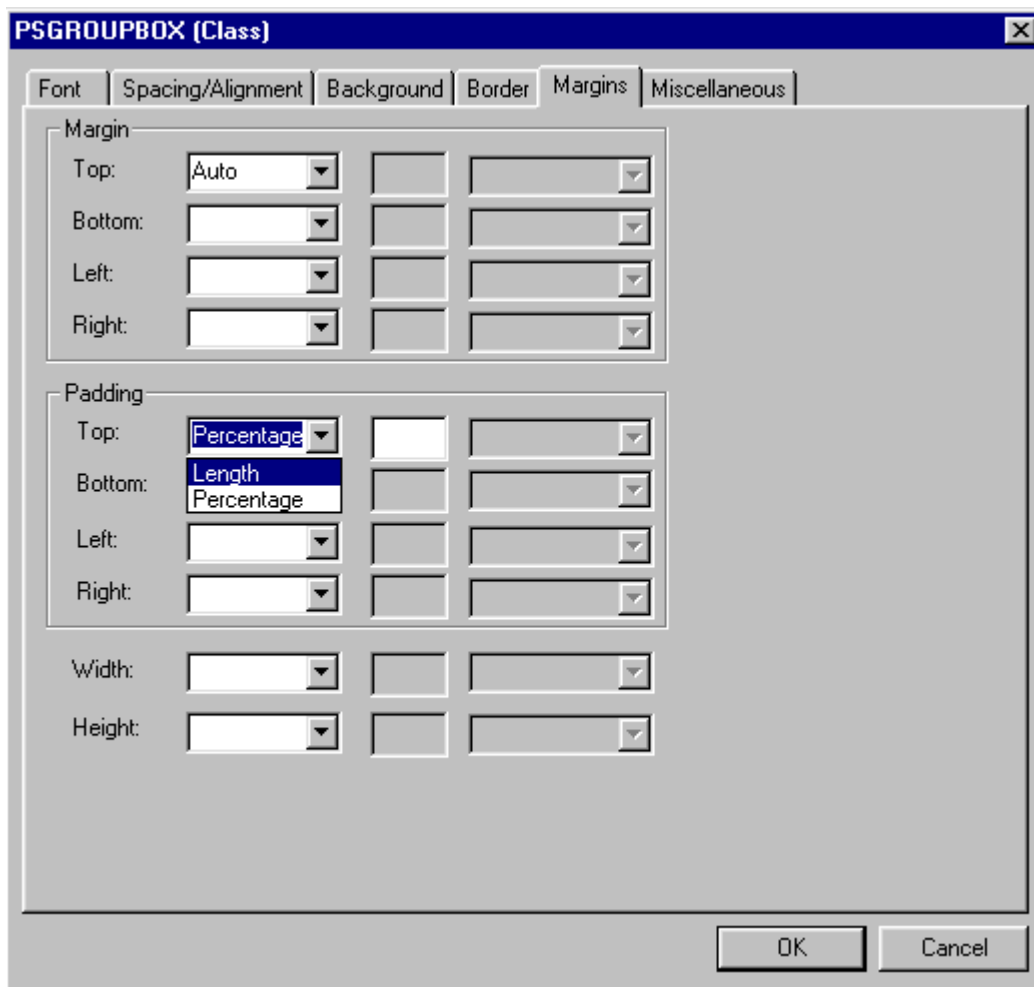
*Ridge:* Border is a three-dimensional ridge that is drawn in colors based on the `<color>` value.

*Inset:* Border is a three-dimensional inset that is drawn in colors based on the `<color>` value.

*Outset:* Border is a three-dimensional outset is that is drawn in colors based on the `<color>` value.

## Specifying Margins

Access the class dialog box. Select the Margins tab.



PSGROUPBOX (Class) dialog box: Margins tab

**Margin and Padding**

Specify the margin and padding for each side of the element. Values are:

*Length:* Specify a font size and the measurement type from the drop-down list box (pixels, inches, millimeters, and so on).

*Percentage:* Specify a number, followed by a percent sign (%). The percentage value is relative to the parent element.

*Auto:* The system calculates the width.

Do not use negative values for padding.

**Width and Height**

Set the width and height of text elements. These properties are most useful with replaced elements, such as images. The system scales the image to fit the value that you specify. If you set either property to *Auto*, and the image needs scaling, the system preserves the aspect ratio of the image. Do not use negative values.

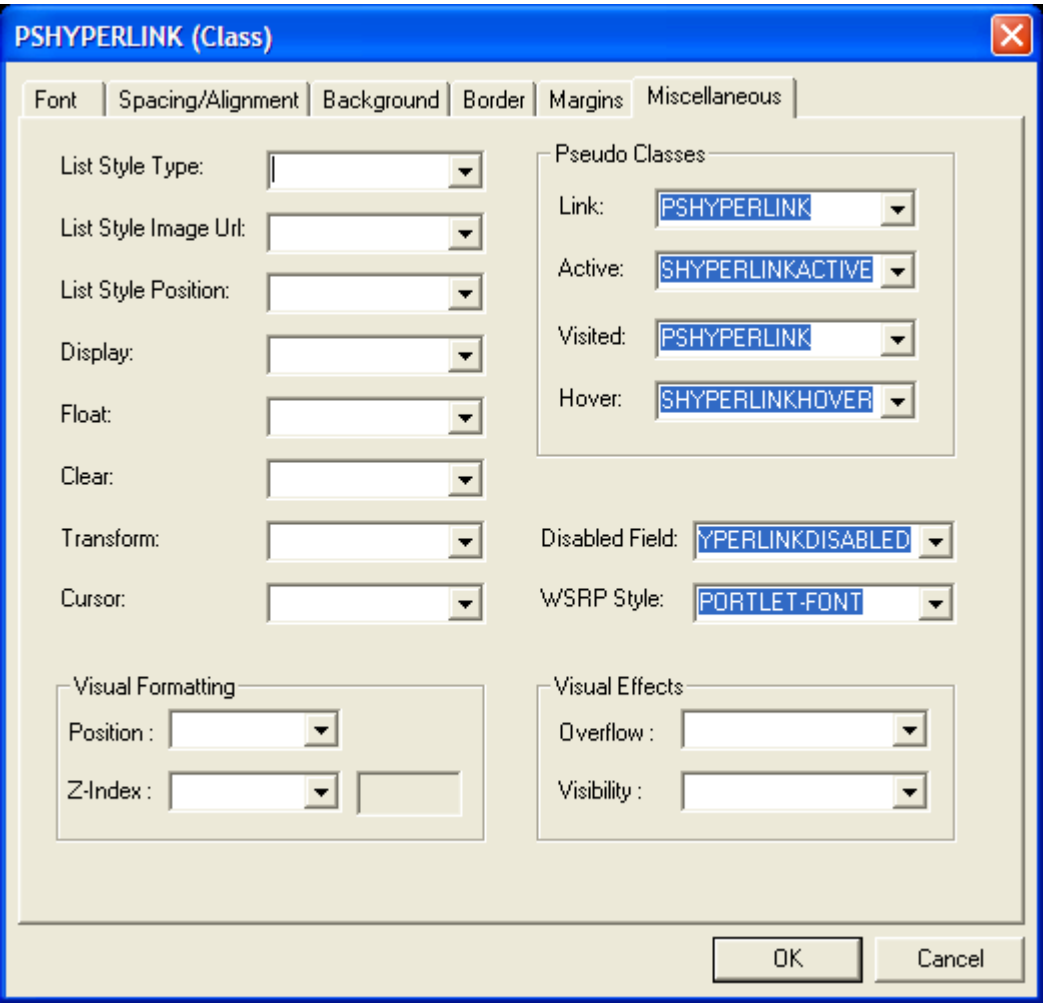
---

**Note.** If the width and height of a replaced element are both set to *Auto*, the system sets these properties to the intrinsic dimensions of the element.

---

# Specifying Miscellaneous Attributes

Access the class dialog box. Select the Miscellaneous tab.



PSHYPERLINK (Class) dialog box: Miscellaneous tab

- List Style Type**

Specify the appearance of the list-item marker if List Style Image URL is blank or if the system cannot display the image that is referenced by the URL.
- List Style Image Url** (list style image uniform resource locator)

Specify the URL containing the image to display as the list-item marker.
- List Style Position**

Specify how the list-item marker is drawn relative to the content. Values are: *Inside* and *Outside*.

<b>Display</b>	<p>Specify how an element is displayed on the canvas (which might be on a printed page or a computer display). Values are:</p> <p><i>Block</i>: Opens a new box. The box is positioned relative to adjacent boxes. Typically, elements like H1 and P are block types.</p> <p><i>ListItem</i>: Similar to block, except that a ListItem marker is added. In HTML, LI typically has this value.</p> <p><i>Inline</i>: Results in a new inline box on the same line as the previous content. The box dimensions are based on the formatted size of the content. If the content is text, it might span several lines, and there will be a box on each line. The margin, border, and padding properties apply to inline elements but do not affect the line breaks.</p> <p><i>None</i>: Deactivates the display of the element, including children elements and the surrounding box.</p>
<b>Float</b>	<p>Specify how the element floats with the text. Values are:</p> <p><i>None</i>: Causes the element to appear where it appears in the text.</p> <p><i>Left</i> and <i>Right</i>: Cause the element to move to the left or right, and the text wraps on the right-hand or left-hand side of the element, respectively. With a <i>Left</i> or <i>Right</i> value, the element is treated as block-level and the Display property is ignored.</p>
<b>Clear</b>	<p>Specify the sides of an element on which floating elements are not accepted. Values are:</p> <p><i>Left</i>: An element appears below any floating element on the left-hand side.</p> <p><i>Right</i>: An element appears below any floating element on the right-hand side.</p> <p><i>Both</i>: An element appears below any floating element on the left-hand and the right-hand side.</p> <p><i>None</i>: Floating elements are allowed on all sides.</p>
<b>Transform</b>	<p>Use for text elements only. Values are: <i>Capitalize</i>, <i>Uppercase</i>, <i>Lowercase</i>, and <i>None</i>.</p>
<b>Cursor</b>	<p>Specify how the cursor displays when passed over the element.</p>
<b>Disabled Field</b>	<p>Specify how the link should appear when the link (or field) is disabled.</p>
<b>WSRP Style</b>	<p>ENTER DATA HERE</p>

### **Pseudo Classes**

Pseudo classes are mechanisms that extend the expressiveness of style sheets. Using pseudo classes, you can change the style of page links based on whether and when the links have been visited. Pseudo classes do not exist in HTML; that is, they are not visible in the HTML code.

PeopleSoft uses pseudo classes to specify how a browser indicates to a user the status of links in a document that the user is viewing. For example, it is common for a browser to display a document link in a different color than the rest of the text.

<b>Link</b>	Specify how the link should appear normally.
<b>Active</b>	Specify how the link should appear when it is actively selected.
<b>Visited</b>	Specify how the link should appear if it has been visited.
<b>Hover</b>	Specify how the link should appear when it is designated but not activated. For example, when the cursor hovers over a box that is generated by the element.

## Visual Formatting

<b>Position</b>	<p>The position property places an element in a static, relative, absolute or fixed position.</p> <p><i>Static:</i> The page element always has the position the normal flow of the page gives it (a static element ignores any top, bottom, left, or right declarations). This is the default.</p> <p><i>Relative:</i> The page element moves relative to its normal position, for example, <i>left:20</i> adds 20 pixels to the element left position.</p> <p><i>Absolute:</i> The page element is positioned at the specified coordinates relative to its containing block. The element's position is specified with the <i>left,top,right</i>, and <i>bottom</i> properties.</p> <p><i>Fixed:</i> The page element is positioned at the specified coordinates relative to the browser window. <i>Absolute:</i> The page element is positioned at the specified coordinates relative to its containing block. The element's position is specified with the <i>left,top,right</i>, and <i>bottom</i> properties.</p>
-----------------	--

<b>Z-Index</b>	<p>Positioning of elements in CSS occurs in three dimensions, not just two. The placement of elements along this third z-axis is exactly what this property controls. The third dimension is perpendicular to the screen, giving the screen a sense of depth. Elements can be overlapped, with higher or closer elements obscuring elements that are lower or farther away (element transparency becomes an important issue with this capability.)</p> <p>Each element's rendering box is automatically assigned an integer z-index stacking level based on its context in the document. Boxes with greater z-axis numbers will appear in front of boxes with lower z-axis numbers ('0' and negative numbers are allowed.) Explicitly setting the z-index property for an element box not only sets its z-position relative to other element boxes in its current context, it also initiates a new stacking context hierarchy, in which the current element box and its child elements partake. If two or more boxes have the same stacking level within the same context, they are rendered back to front in the order of the document tree. If no z-index property is set for an element box, it inherits the stacking level of its parent element box.</p>
----------------	---

## Visual Effects

### Overflow

This property specifies whether the content of a block-level element is clipped when it overflows the element's box (which is acting as a containing block for the content).

*Visible:* This value indicates that content is not clipped; it may be rendered outside the block box.

*Hidden:* This value indicates that the content is clipped and that no scrolling mechanism should be provided to view the content outside the clipping region; users will not have access to clipped content. The size and shape of the clipping region is specified by the Clip property.

*Scroll:* This value indicates that the content is clipped and that if the user agent uses scrolling mechanism that is visible on the screen (such as a scroll bar), that mechanism should be displayed for a box whether any of its content is clipped. This avoids any problem with scroll bars appearing and disappearing in a dynamic environment. When this value is specified and the target medium is print or projection, overflowing content should be printed.

*Auto:* The behavior of this value is user agent-dependent, but should cause a scrolling mechanism to be provided for overflowing boxes.

### Visibility

This property specifies whether the boxes generated by an element are rendered. Invisible boxes still affect layout (set the Display property to *None* to suppress box generation altogether).

*Visible:* The generated box is visible.

*Hidden:* The generated box is invisible (fully transparent), but still affects layout.

*Collapse:* This property may be used in conjunction with scripts to create dynamic effects on rows and columns. If used on elements other than rows or columns, *Collapse* has the same meaning as *Hidden*.

---

## Working With Free Form Sub Style Sheets

This section discusses how to:

- Create free form sub style sheets.
- Insert free form sub style sheets into standard style sheets.
- Convert standard style sheets to free form sub style sheets.
- Rename free form sub style sheets.
- Delete free form sub style sheets.
- Include meta variables and comments in free form sub style sheets.
- Change font and color settings for free form sub style sheets.

## Creating Free Form Sub Style Sheets

To create free form sub style sheets:

1. Select File, New.
2. Select Style Sheet from the list.
3. Select the Free Form Sub Style Sheet style sheet type.
4. Select File, Definition Properties to enter a description for documentation purposes, when done, click the OK button.
5. Select the Free Form tab and enter the text for the style sheet.
6. Save the style sheet.

---

**Important!** The Style Sheet Editor does not perform any syntax checking on free form sub style sheets. You must verify and correct any style errors that occur at runtime.

---

## Inserting Free Form Sub Style Sheets into Style Sheets

To insert free form sub style sheets into style sheets:

1. Open the receiving standard or sub style sheet.
2. Click the Fixed Form tab.
3. Select Insert, Insert Sub Style Sheet.
4. Select *Freeform Sub Style Sheet* from the Sub Style Sheet Type drop-down list box.
5. Select the free form style sheet from the Sub Style Sheet drop-down list box.
6. Click the OK button and save the style sheet.

## Converting Style Sheets to Free Form Sub Style Sheets

To convert style sheets to free form sub style sheets:

1. Open a standard style sheet or sub style sheet.
2. Select File, Definition Properties.
3. In the Style Sheet Type drop-down list box, select *Freeform Sub Style Sheet* and click the OK button.

---

**Note.** You will be warned that once the style sheet is saved as a free form type the style sheet data cannot be converted back to the standard or sub style sheet type.

---

4. Dismiss the warning dialog box.
5. Save the style sheet.

---

**Note.** It is recommended that you do not modify a PeopleTools style sheet; clone the style sheet and make changes to the copy.

---

## Renaming Free Form Sub Style Sheets

To rename free form sub style sheets:

1. Select File, Rename.
2. Select *Style Sheet* in the Definition drop-down list box and then enter search criteria to find the style sheet definition that you are renaming.
3. Click the Rename button; the definitions that match the search criteria appear in the list box.
4. Double-click the style sheet definition that you would like to rename.
5. Enter the new name of the style sheet definition in the highlighted area and press the Enter key.
6. Click the OK button to confirm the rename.

---

**Note.** When you rename a free form sub style sheet, Application Designer automatically renames all references to the style sheet.

---

## Deleting Free Form Sub Style Sheets

To delete free form sub style sheets:

1. Select File, Delete.
2. Select *Style Sheet* in the Definition drop-down list box and then enter search criteria to find the style sheet definition that you are deleting.
3. Press the Delete key; the definitions that match the search criteria appear in the list box.
4. Click the style sheet that you want to delete and press the Delete key.
5. Click the OK button to confirm the deletion or click the Cancel button to cancel the operation.

---

**Note.** When you delete a style sheet, Application Designer automatically deletes all references to the deleted style sheet.

---

## Including Meta Variables and Comments in Free Form Sub Style Sheets

This section discusses:

- Meta variables.
- Comments.

## **Meta Variables**

Free form sub style sheets support these existing PeopleTools meta variables:

- %AlignStart
- %AlignEnd
- %AlignAmount
- %Image

See *Enterprise PeopleTools 8.50 PeopleBook: PeopleCode Language Reference*, "Meta-HTML."

See *Enterprise PeopleTools 8.50 PeopleBook: Global Technology*, "Implementing Bidirectional Language Support," Developing HTML Areas.

---

**Note.** The system ignores any other meta variables.

---

Meta variables in the free form sub style sheet are resolved at runtime before the style sheet is downloaded to the client. Meta variable processing does *not* occur at design time. In Application Designer, if you select Layout, View in Browser, you do not see any results from meta variable processing.

## **Comments**

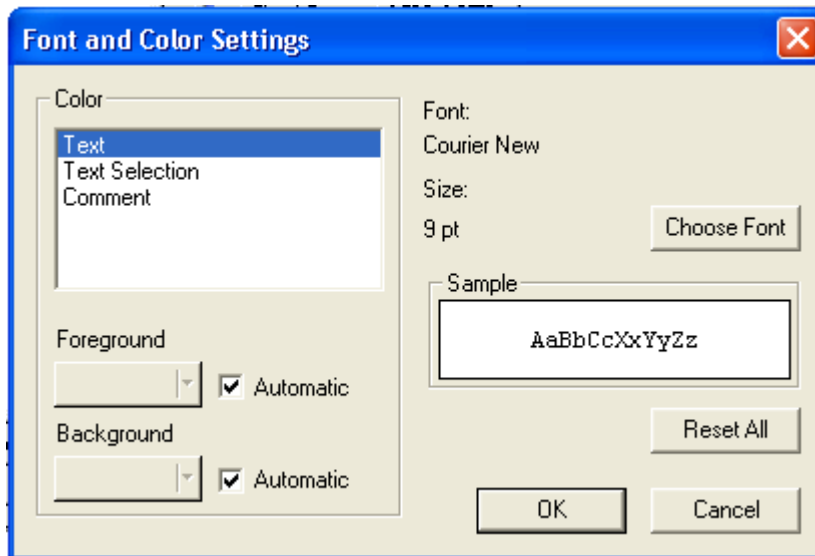
To insert comments, include the text between `/*` and `*/`.

The use of `//` is not supported.

## **Changing Font and Color Settings For Free Form Sub Style Sheets**

When entering code in the Free Form tab of free form sub style sheets, the appearance of the various elements of the code, such as comments is based on the default font and color settings in the Font and Color Settings dialog box.

Access the Font and Color Settings dialog box (Open a free form sub style sheet, select Edit, Display Font and Colors).



Font and color Settings dialog box

This table describes the font and color settings dialog box:

<b>Color</b>	Select the item whose color you wish change. Select from these choices: <i>Text</i> <i>Text Selection</i> <i>Comment</i>
<b>Foreground</b>	Click to select the color of the font.
<b>Background</b>	Click to select the color of the background behind the font.
<b>Choose Font</b>	Click to select the font.
<b>Reset All</b>	Click to reset all defaults to the original settings.

You can change the default font and color settings for free form sub style sheets.

### ***Changing the Default Font and Color Settings***

To change the default font and color settings:

1. In the Color group box, select an item in the list box. Select from these choices:  
*Text*  
*Text Selection*  
*Comment*
2. Select a color from the Foreground drop-down color box.
3. Select a color from the Background drop-down color box.

4. Click the Choose Font button to select another font.
5. Click Reset All to restore the settings to the default settings.
6. Click the OK button.

---

## Working with the Default Style Sheet

Applications use a default style sheet which contains all default style classes for PeopleSoft application. You specify the default application style sheet on the PeopleTools Options, the System Options, and the Registry Options pages. When you install your PeopleSoft applications, *PSSTYLEDEF* is the application style sheet for PeopleSoft applications prior to and including application release 9.0. *PSSTYLEDEF\_SWAN* is the default application style sheet when you install your PeopleSoft applications for releases 9.1 and newer.

This section discusses:

- Configuring PSSTYLEDEF as the default style sheet.
- Configuring PSSTYLEDEF\_SWAN as the default style sheet.
- Configuring other default style sheets

---

**Note.** Only standard style sheets can be used as the default application style sheet.

---

**Important!** Oracle has updated the styles that define the PeopleSoft user interface. This PeopleSoft PeopleTools release delivers the classic (old) style as well as two new styles: a dark blue style and a light blue style. PeopleSoft 8.4 applications and pre-8.50 PeopleSoft PeopleTools system databases use the classic style, but all other applications use the new dark blue style. The dark blue style is set as the default. To use another interface, you have to delete the sub style sheets associated with the dark blue style and replace them with either the classic or light blue substyle sheet. The classic and light blue styles are considered deprecated as of PeopleTools 8.50.

To enable a deprecated user interface consult your upgrade and installation documentation.

---

## Configuring PSSTYLEDEF as the Default Style Sheet

PSSTYLEDEF is the default application style sheet. The PSSTYLEDEF style sheet comprises all default style classes and consists of these sub style sheets:

- JSR168STYLEDEF: Defines PeopleSoft Portlet default style classes.
- PAALTERNATE: Defines PeopleSoft Portal Solutions alternatives to PS default style classes.
- PSACE: Defines analytic calculation engine and analytic grid default style classes.
- PSALTERNATE: Defines application alternatives to PS default style classes.
- PSCHARTSTYLE: Defines charting default style classes.
- PSORGCHART: Defines organization chart default style classes.

- **PSRATEBOX**: Defines rating box default style classes.
- **PTSTYLEDEF**: Defines the standard PeopleSoft Pure Internet Architecture and portal default style classes.
- **PSNAV2**: Defines the 8.50 Menu 2.0 navigation style classes.

This is a free form sub style sheet.

- **PSPOPUP\_CSS**: Defines the pop-up dialog box page style classes.

This is a free form sub style sheet.

- **PSTAB\_PTCSS**: Defines the page tab style classes.

This is a free form sub style sheet.

---

**Note.** It is recommended that you do not modify **PSSTYLEDEF** or other PeopleTools style sheets; clone existing style sheets and make changes to the copy.

---

To change the default application style sheet to **PSSTYLEDEF**:

---

**Note.** **PSSTYLEDEF** is the default; you should only perform these steps if you are reverting to **PSSTYLEDEF** after implementing another style sheet.

---

1. In the browser, select PeopleTools, Utilities, Administration, PeopleTools Options.
2. Enter *PSSTYLEDEF* in the Style Sheet Name field.
3. Save the page.
4. Select PeopleTools, Portal, Portal Utilities, System Options.
5. Enter *PSSTYLEDEF* in the Style Sheet Name field.
6. Save the page.
7. In the browser, select PeopleTools, Portal, Portal Utilities, Registry Options.
8. Select the desired registry.
9. Enter *PSSTYLEDEF* in the Style Sheet Name field.
10. Save the page.
11. Run the *UPGPT850SWAN* Application Engine program.
12. Stop the web server, delete all cache files and restart the web server.
13. Stop the application server, delete all cache files and restart the web server.

## Configuring **PSSTYLEDEF\_SWAN** as the Default Style Sheet

The **PSSTYLEDEF\_SWAN** style sheet is comparable to **PSSTYLEDEF** in its sub style sheets and style classes. The page styles provided by **PSSTYLEDEF\_SWAN**, is the standard design for all Oracle Applications Unlimited products.

---

**Note.** The My Reports pagelet contains exceptions to the SWAN style sheet and might appear to look different from the standard SWAN style.

---

- JSR168STYLEDEF\_SWAN: Defines PeopleSoft Portlet default style classes.
- PAALTERNATE\_SWAN: Defines PeopleSoft Portal Solutions alternatives to PS default style classes.
- PSACE\_SWAN: Defines analytic calculation engine and analytic grid default style classes.
- PSALTERNATE\_SWAN: Defines application alternatives to PS default style classes.
- PSCHARTSTYLE\_SWAN: Defines charting default style classes.
- PSORGCHART\_SWAN: Defines organization chart default style classes.
- PSRATEBOX\_SWAN: Defines rating box default style classes.
- PTSTYLEDEF\_SWAN: Defines the standard PeopleSoft Pure Internet Architecture and portal default style classes.
- PSNAV2\_SWAN: Defines the 8.50 Menu 2.0 navigation style classes

This is a free form sub style sheet.

- PSPOPUP\_CSS\_SWAN: Defines the pop-up dialog box page style classes.

This is a free form sub style sheet.

- PSTAB\_PTCSS\_SWAN: Defines the page tab style classes.

This is a free form sub style sheet.

---

**Note.** It is recommended that you do not modify PSSTYLEDEF\_SWAN or other PeopleTools style sheets; clone existing style sheets and make changes to the copy.

---

To change the default application style sheet to PSSTYLEDEF\_SWAN:

1. In the browser, select PeopleTools, Utilities, Administration, PeopleTools Options.
2. Enter *PSSTYLEDEF\_SWAN* in the Style Sheet Name field.
3. Save the page.
4. Select PeopleTools, Portal, Portal Utilities, System Options.
5. Enter *PSSTYLEDEF\_SWAN* in the Style Sheet Name field.
6. Save the page.
7. In the browser, select PeopleTools, Portal, Portal Utilities, Registry Options.
8. Select a registry.
9. Enter *PSSTYLEDEF\_SWAN* in the Style Sheet Name field.

10. Save the page.

---

**Note.** Repeat steps 7 through 10 for each portal registry.

---

11. Run the *UPGPT850SWAN* Application Engine program.
12. Stop the web server, delete all cache files and restart the web server.
13. Stop the application server, delete all cache files and restart the application server.

## Configuring Other Default Style Sheets

You can use other style sheets as the default, including those which comprise free form text only, as long as the style sheet contains all default style classes.

To change the default application style sheet:

1. In the browser, select PeopleTools, Utilities, Administration, PeopleTools Options.
2. Enter the style sheet in the Style Sheet Name field.
3. Save the page.
4. Select PeopleTools, Portal, Portal Utilities, System Options.
5. Enter the style sheet name in the Style Sheet Name field.
6. Save the page.
7. In the browser, select PeopleTools, Portal, Portal Utilities, Registry Options.
8. Enter the style sheet name in the Style Sheet Name field.
9. Save the page.
10. Stop the web server, delete all cache files and restart the web server.
11. Stop the application server, delete all cache files and restart the application server.

---

## Overriding Default Page Style Properties

Unless you override the default style sheet at the page level, the browser renders all pages and page fields based on the default style classes that comprise the application style sheet. If you want to use different styles to create subtle changes or entirely unique pages, you can override styles at the page and page field levels.

This section discusses how to:

- Override the default page style sheet.
- Override the default page background style.
- Override default page field styles.

## Overriding the Default Page Style Sheet

You can override the default page style sheet by accessing the page properties dialog box of any PeopleSoft page. When you override the default page style sheet, you are not changing the default style classes for the page; you are indicating to the system to select the default style classes from a non-default style sheet. Note these points when overriding the page style sheet:

- Changing the page style sheet overrides the default style sheet for only the selected page.
- Only the custom and default style classes which comprise the current page style sheet—whether default or override—are available for use by the page and its elements. Style classes on other style sheets are inaccessible.
- If you select a page style sheet that comprises only free form sub style sheets, the system constrains all subordinate page elements to their default style classes.
- If you select a page style sheet that comprises only free form sub style sheets, the page designer renders the page using the application style sheet, not the free form style sheet.

To override the default page style sheet:

1. Open the page in PeopleSoft Application Designer.
2. Select File, Definition Properties to access the Page Properties dialog box.
3. Select the Use tab.
4. Click the Page Style Sheet drop-down list box and select a style.
5. Click the OK button and save the page.
6. Select Layout, View in Browser to preview the page.

Unlike the page designer view, this option renders the page with all style classes, including free form styles.

## Overriding the Default Page Background Style

The page background derives from the Background tab of the PSPAGE style class contained in the current page style sheet—whether default or override. You can override the default page background style.

Note these points about changing the default value of the page background style:

- Changing the page background style value for a specific page overrides the background properties of only the current page.
- If the current page style sheet comprises only free form sub style sheets, the system constrains the page background style to the default style class.
- The Page Background Style drop-down list box displays only the style classes which comprise the current page style sheet—whether default or override. Style classes that do not appear in the current page style sheet don't appear in the list.

To override the default page background style:

1. Open the page in PeopleSoft Application Designer.
2. Select File, Definition Properties to access the Page Properties dialog box.
3. Select the Use tab.
4. Click the Page Background Style drop-down list box and select a style.
5. Click the OK button and save the page.
6. Select Layout, View in Browser to preview the page.

Unlike the page designer view, this option renders the page with all style classes, including free form styles.

## Overriding the Default Page Field Styles

You can override the default style of most page field controls, such as edit boxes or drop-down list boxes, and so on. Many page field controls include a separate *label* style which enables you to override the label style independent of the control style itself.

Note these points about changing the default style of a page field:

- The Style drop-down list box displays only the style classes which comprise the current page style sheet—whether default or override. Style classes that do not appear in the current page style sheet do not appear in the list.
- If the page style sheet comprises only free form sub style sheets, the system constrains all page field controls to the default style despite the Style field value.

To override the default page field style:

1. Open a page in PeopleSoft Application Designer.
2. Double click the page field control to access the Properties dialog box.
3. Select the Record tab.
4. Click the Style drop-down list box and select the new style.
5. Click the Label tab.
6. Click the Style drop-down list box and select the new style.
7. Select Layout, View in Browser to preview the page.

Unlike the page designer view, this option renders the page with all style classes, including free form styles.

8. Save the page.

In addition to setting the style property at design time, you can also use the PeopleCode Field object Style property to programmatically change the style class of a field at runtime.

See *Enterprise PeopleTools 8.50 PeopleBook: PeopleCode Developer's Guide*, "PeopleCode and PeopleSoft Pure Internet Architecture," Using the Field Object Style Property.

---

## Modifying Page Tabs

This section provides an overview of tab definitions and discusses how to:

- Create tab images.
- Customize page tab style sheets.

## Understanding Cascading Style Sheet Tabs

To provide a greater degree of and simpler user-level configuration, PeopleSoft applications use the sliding door technique for tabs. Cascading style sheets (CSS) have the advantage of layering background images which enables them to slide over each other to create visually interesting effects.

The default style sheet and the SWAN style sheet contain CSS tabs. You can, however, change the background colors of these CSS folder and grid tabs. To change the background colors of tabs, it is helpful to have a basic understanding of how tab definitions are constructed and stored in the database.

### ***Sliding Doors Technique***

Sliding door tabs expand and contract based on the size of the text by using just two separate background images: one image to render the left side of a tab and the other image to render the right. Think of these two images as the sliding doors that complete one opening. The doors slide together and overlap more to fill a narrower space, or slide apart and overlap less to fill a wider space. The background images only show in the available doorway of the element to which they are applied (content area + padding). The two images are anchored to the outside corners of their respective elements. The visible portions of these background images fit together inside the doorway to form a tab-like shape.

### ***Sliding Door Tab Images***

These are the image names that you need to know if you are modifying the style sheets that ship with the application.

For the dark blue look and feel, folder tabs and grid tabs use these images:

- PSTAB\_LEFT.gif.
- PSTAB\_RIGHT.gif.

For the SWAN look and feel, the folder tabs and the grid tabs use these images:

- PSTAB\_LEFT\_SWAN.gif.
- PSTAB\_RIGHT\_SWAN.gif.

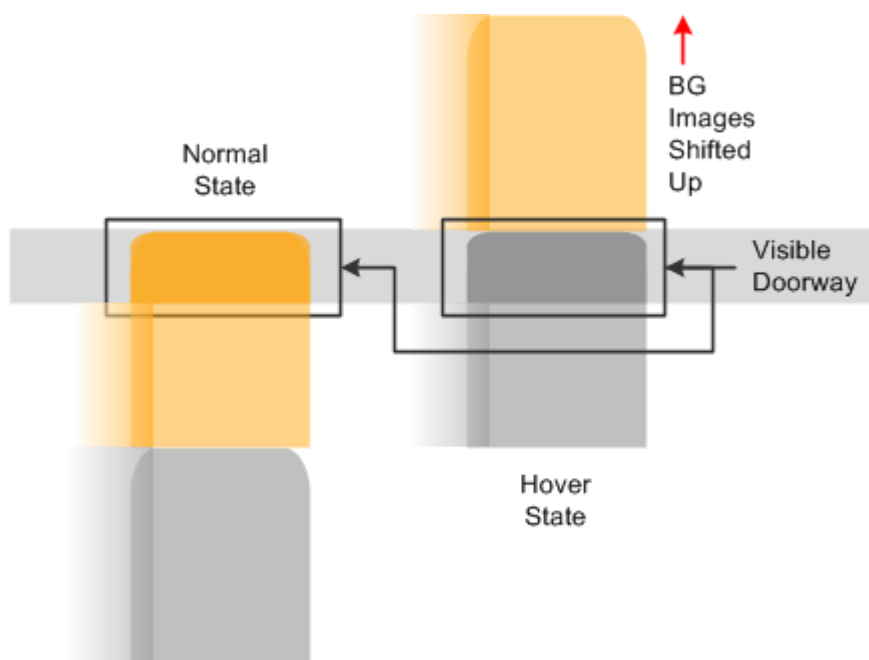
The CSS code behind the rendering of the tab image is:

```

#PSTAB a
{
background:transparent url(%IMAGE(PSTAB_LEFT)) no-repeat left top;
padding:0 2px 0 5px;
}
#PSTAB a span
{
font-size:9pt;
background:transparent url(%IMAGE(PSTAB_RIGHT)) no-repeat right top;
padding:4px 14px 2px 5px;
}

```

When working with the sliding door tabs, the images load faster, are simpler to maintain, and the text within them can be scaled up or down significantly in size without breaking the page. Along with these new images, applications take advantage of the CSS background-position property to move into view the appropriate portion of the combined-state background image when the user hovers over a tab.



Style sheet sliding door tab preload technique

Applications use the shifted background positions from the active, inactive and hover states.

## Creating Tab Images

Before you can assign a new background color to the active, inactive, and hover states of the tabs, you must create the necessary tab images in that color. To produce tabs in a different color combination:

1. In Application Designer, open the image definitions for the tabs that you want to change.
2. Select File, Export Image and in the Save Image dialog box, select the directory in which you want to place the images.

3. Use any image editor to alter the image colors to the same as the new background color you choose.

---

**Note.** Do not replace any of the transparent pixels with solid pixels. Additionally, it is recommended that you leave the size and shape as they are even though the style sheets allows you to alter the size and shape of the images.

---

4. Select File, New, Image to add the new image definitions to the database.
5. Save the new image files under the appropriate names.
6. Customize the tabs.

See [Chapter 15, "Creating Style Sheet Definitions," Customizing Page Tab Style Sheets, page 445.](#)

## Customizing Page Tab Style Sheets

Once you create the new images, you must change the properties of the style classes rendering the new sliding door tabs. The style classes that render the folder tabs and the grid tabs are PSTAB and PTGRIDTAB, respectively. Both style classes are part of the PSTAB\_PTCSS free form style sheet.

To make modifications to the tab properties:

1. Open the appropriate free form style sheet: PSTAB\_PTCSS if you are using the default style sheet or PSTAB\_PTCSS\_SWAN if you are using the SWAN style sheet.
2. For folder tabs, modify the PSTAB style class. For grid tabs, modify the PTGRIDTAB style class.

For example, if you have two new folder tab images, the left and the right image of the tab, each has a different name and you must change the code as you see next:

```
#PSTAB a
{
background:transparent url(%IMAGE(<newlyCreatedLeftSideTabImageName>)) no-repeat=
  left top;
....
}
#PSTAB a span
{
font-size:9pt;
background:transparent url(%IMAGE(<newlyCreatedRightSideTabImageName>)) no-repeat=
  right top;
....
}
```

---

**Note.** You must replace <newlyCreatedLeftSideTabImageName> and <newlyCreatedRightSideTabImageName> with the names of the new images that you create. You might have to modify the position property if the new images varies in size from the original images.

---

3. Because the image contains all active, inactive and hover color properties, it is enough to change the pixel properties and background positioning, if there is any change in size and shape of the images from what have been delivered.
4. You can also modify style class properties such as border, padding, font and text to create the required look of the tabs you are implementing.

5. You can override the tab bottom line by changing the following code:

```
#PSTAB #TABBOTTOMLINE
{
  Color: #809EBA;
  background: repeat-x right 18px;
}
```

6. You can adjust the hover properties as seen in the following example.

```
#PSTAB a:hover a
{
  color: #555555;
}

#PSTAB a:hover span
{font-size:9pt;
background-position:100% -50px;
}
```

---

**Note.** The `<span>` element renders the main part of the tab. Adjust any necessary properties, such as `background-position` in the `<span>` element to render the image correctly. The left image is small and you need adjust only the color and possibly the border properties.

---

7. Save the style sheet.
8. Stop the application server, clear the cache, and restart the application server.
9. Stop the web server, clear the cache, and restart the web server.

---

## Modifying Push Buttons in SWAN Style Sheets

This section provides an overview of push buttons in the SWAN style sheet and describes how to:

- Create SWAN style sheet push button images.
- Customize SWAN push buttons in HTML areas.

## Understanding Cascading Style Sheet Push Buttons

PeopleSoft applications provide the PTPUSHBUTTON free form style sheet as part of the SWAN style sheets. You can edit all properties in the PTPUSHBUTTON style sheet to change the look and feel of the classic yellow push button. Like sliding door tabs, push buttons expand and collapse to accommodate the button text and the buttons come with a left and right image. The left image is associated with the `<a>` element and the right image with the `<span>` element. The `<span>` element contains the text and font properties for the button. The PTPUSHBUTTON free form style sheet overrides the existing properties provided in the PSPUSHBUTTON style class in the PSSTYLEDEF\_SWAN style sheet. You edit push buttons similarly to the way that you edit the sliding door tabs.

### See Also

[Chapter 15, "Creating Style Sheet Definitions," Sliding Door Tab Images, page 443](#)

## Creating SWAN Style Sheet Push Button Images

All the steps for altering the sliding door tab images are the same for the sliding door buttons. Do not alter the size of the button as this affects the way background position is defined for the buttons.

### See Also

[Chapter 15, "Creating Style Sheet Definitions," Sliding Door Tab Images, page 443](#)

## Customizing SWAN Push Buttons in HTML Areas

If there are any custom defined push buttons in HTML areas that display in the yellow button background, you can change them to appear in the SWAN push button style. To customize SWAN push buttons in HTML areas:

1. Open the HTML area that contains the push button you want to modify.
2. Locate the `<input type=button.../>` tag:
3. After the `<input type...>`tag, insert the following tags: `<a class='PSPUSHBUTTON'><span> ... </span></a>`.



## Chapter 16

# Creating Image Definitions

This chapter provides an overview of images and discusses how to:

- Create new image definitions.
- Open an image definition.
- Update an image definition.
- Convert images.
- Consolidate images.
- Use the catalog of image definitions.

---

## Understanding Images

Images improve the look and usability for the user. PeopleSoft Internet Architecture comes equipped with several different types of image-related features, each of which serves a different function. The following table provides a brief description of each image type:

<i><b>Image Type</b></i>	<i><b>Description</b></i>
Image field	Use for storing images in a user-defined format, such as bitmap (BMP) or PostScript (EPS). Image fields cannot be changed at runtime.
ImageReference field	Use to change an image dynamically at runtime by using PeopleCode.
Image definition	Use as the source from which all static images and image fields are taken and used elsewhere in the system.
Image control	Place on a page when you want a variable image that is taken from a record field in the database.
Static image control	Use to place a static image definition on the page, such as on a push button.

---

## Creating New Image Definitions

This section provides an overview of image definitions and discusses how to:

- Create an image definition.
- Set image properties.
- Import and create alternate image types.

## Understanding Image Definitions

Image definitions are managed PeopleSoft definitions that you can associate with a variety of page controls. They are primarily for aesthetic purposes, but can also demonstrate a simple function, such as an arrow on a push button. Many organizations store images along with the rest of their employee, customer, and supplier data as part of their ongoing business operations. For example:

- Retailers often store images of each product with the standard merchandise information that they use to manage inventory.

Online retailers can display product pictures on their customer web sites.

- Many organizations store images of each employee as part of their standard human resources information.
- Consumer banks might store images of their customers' canceled checks.

PeopleSoft Application Designer enables you to create an image definition from any type of image file and store it in a central PeopleTools image catalog. In this process, you convert the image files into image definitions and store them in the image catalog so that you can refer to them from a PeopleTools application. After you create the image definition, it is available for use throughout the system, such as in a static image page control or on a push button or link.

### ***Image Types***

When creating new image definitions to use with applications, you might want to use different image types for different purposes. Some developers believe, for example, that JPG is the best format for photographic images because it provides the greatest compression of any bitmap format in common use. However, some believe that JPG is not as effective in compressing text and drawings as it is at compressing photographs. GIF images, on the other hand, seem to be the most widely used format for image storage and continue to be the preferred format for storing text and drawings. Regardless of your image type preferences, PeopleSoft enables you to store all image types in the image catalog.

However, WBMP image types cannot be viewed in PeopleSoft Application Designer during design time, but they can be viewed through the mobile device at runtime. By providing an alternate image type of the image, such as JPG, PeopleSoft applications enable you to import images for viewing purposes only in PeopleSoft Application Designer. The image that you view at runtime in the wireless application is still the original WBMP image.

## Creating an Image Definition

To create an image definition from an image file:

1. Select File, New.
2. Select Image.
3. Click OK to access the Open Image File dialog box.
4. Select an image file type.

You can select from several file types, such as bitmap (BMP), wireless bitmap (WBMP), device independent bitmap (DIB), Graphics Interchange Format (GIF), and Joint Photographic Experts Group (JPEG). You can create an image definition by using any file type. WBMP images cannot be viewed in Application Designer unless you import an alternate image type.

---

**Note.** Application Designer supports animated gif files. In Application Designer, the file is not animated at design time, however when you place an animated gif image on a page definition, the file is animated on the page at runtime.

---

See [Chapter 16, "Creating Image Definitions," Importing and Creating Alternate Image Types, page 452.](#)

---

**Note.** Some browsers do not support all image types.

---

See Customer Connection, "Troubleshooting Browser Limitations."

5. Select an image file name.

PeopleSoft Application Designer displays a warning message if the image size is greater than 32 kilobytes. The maximum image size depends on the database platform that you are using. Some database platforms support much larger image sizes while others limit the size.

6. Select Open.

The image is now open as an image definition in the definition workspace.

7. Select File, Save to save the image definition in the PeopleSoft image repository.

Upon saving, the Image Properties dialog box appears.

## Setting Image Properties

After you create a new image definition, you can set the image properties.

To set image properties:

1. Access the Image Properties dialog box.  
If not already open, select File, Definition Properties.
2. (Optional) Enter a description on the General tab.

3. Select the Use tab to view image use properties.

<b>Image Format</b>	Specify the main and alternate image formats. The main image format, which appears at runtime, is the format in which the image was imported, such as BMP. The alternate format is the optional image format that you can specify if you want a WBMP image as the main image format that appears at runtime.
<b>Image URL</b> (image uniform resource locator)	Specify a reference to an image on a web server. This can be used as an alternative to storing an image definition in the database once an image definition is saved. When the user opens a page containing a control that is associated with this image definition, the application retrieves the image from the URL, rather than from the database. Set up URLs in the URL Maintenance utility.  <i>See Enterprise PeopleTools 8.50 PeopleBook: System and Server Administration, "Using PeopleTools Utilities," URL Maintenance.</i>
<b>Mirror this image in right-to-left contexts</b>	Select this property to have the system flip the image horizontally when a language that is read right-to-left (RTL) is in use.  Pages where the languages have an RTL reading order, for example, Arabic might also display images whose directionality depends on the directionality of the language. For example, an arrow in a margin that points to the text in a left-to-right (LTR) language, such as English, will point to the right, but in an RTL language the arrow should be pointing to the left.
<b>Used in Workflow Maps</b>	Select to filter images that appear in a list box for the Change Icon function for PeopleSoft Workflow designers.

## Importing and Creating Alternate Image Types

To import a WBMP image type:

1. Select File, New.
2. Select Image.

The Open Image File dialog box appears.

3. Select *WBMP* from the Files of Type drop-down list box.
4. Locate the image that you want to import.
5. Click the Open button.

The Content Repository Interface dialog box appears, prompting you to select an alternate image.

6. Select Yes.

The Open Alternate Image File dialog box appears, prompting you to select an alternate image type. You must have the same image stored in the directory under a different file type, such as JPG, to view the image in PeopleSoft Application Designer. If you select No, you can still open the WBMP image definition, but you cannot see it. Instead, you receive a message that the image cannot be viewed in PeopleSoft Application Designer.

7. Select the alternate image format and the appropriate file.
8. Click Open.
9. Select File, Save As.

Upon saving, the Image Properties dialog box appears. If you select the Use tab, the Image Format group box shows the main image format as WBMP and the alternate as JPG. Now you can view the image in PeopleSoft Application Designer as a JPG image and still display the image at runtime as WBMP.

---

## Opening an Image Definition

To open an image definition:

1. Select File, Open.
2. Select *Image* from the Definition drop-down list box.
3. Click Open to view the full list of image definitions and brief descriptions.

---

## Updating an Image Definition

This section discusses how to:

- Update an image definition.
- Change the image display size.
- Specify the image storage format.

### Updating an Image Definition

Occasionally, the original image file from which you created an image definition might change. To keep the image definitions current, you might want to update them rather than create entirely new image definitions.

To update an image definition:

1. To change the image for a saved image definition, right-click the open definition and select Update Image.

The Open Image File dialog box appears, in which you can select the changed image file to replace the open image definition.

2. Click Open.

This replaces the previous image in the image definition with the new image that you selected.

3. Select File, Save to save the current image definition with the new image.

## Changing the Image Display Size

If the image file is smaller than you want it to appear in the open image definition, you can increase the size by zooming in on the definition workspace. This does not alter the size of the image in the image catalog.

To change image definition display size:

1. Open the image.
2. Right-click the image and select Zoom to see the different size ratios.
3. Select a new image display percentage.

The image automatically changes to that selection.

## Specifying the Image Storage Format

You can specify a default format in which image definitions are stored. For example, if JPG is specified as the preferred storage format, then a BMP image is stored as a JPG after it is converted to an image definition.

To specify an image definition storage format:

1. Select Tools, Options to open the Options dialog box.
2. Select the Image tab.

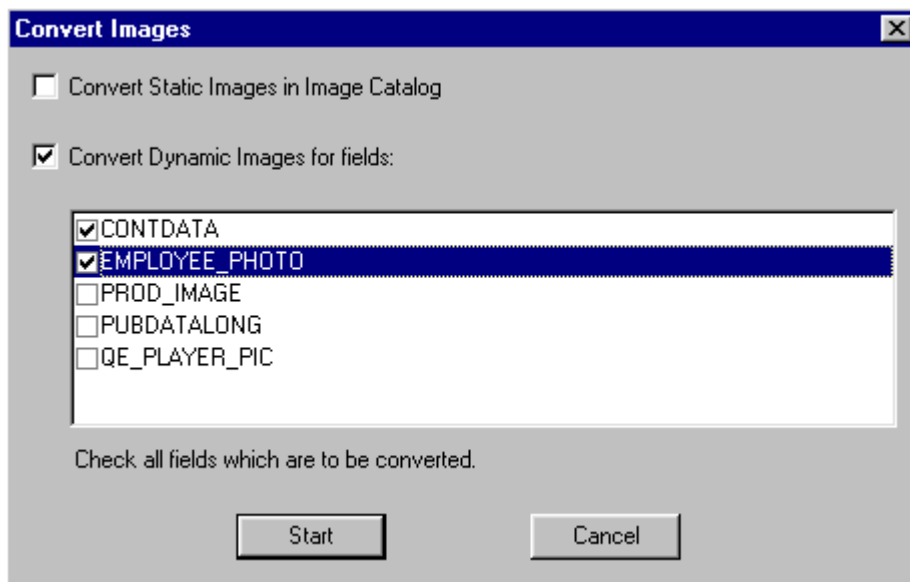
<b>No conversion</b>	Select to import all image definitions in their original formats. This is the default.
<b>DIB and JPG</b>	Select to convert and store imported image definitions as either DIB or JPG in the PeopleSoft Application Designer image catalog. GIF and WBMP images cannot be converted to DIB or JPG.

3. Click OK.

---

## Converting Images

Not all browsers support all image formats, but most browsers support the JPG image type. Therefore, PeopleSoft has a utility to convert all application images to JPG.



Convert Images dialog box

To convert images to JPG format:

1. Select Tools, Upgrade, Convert Images.
2. Select one of these check boxes:

**Convert Static Images in Image Catalog**      Converts all image definitions that are stored in the image catalog of the PeopleSoft database.

**Convert Dynamic Images for fields**      Converts all images that are dynamically referenced by PeopleCode to appear in image fields.

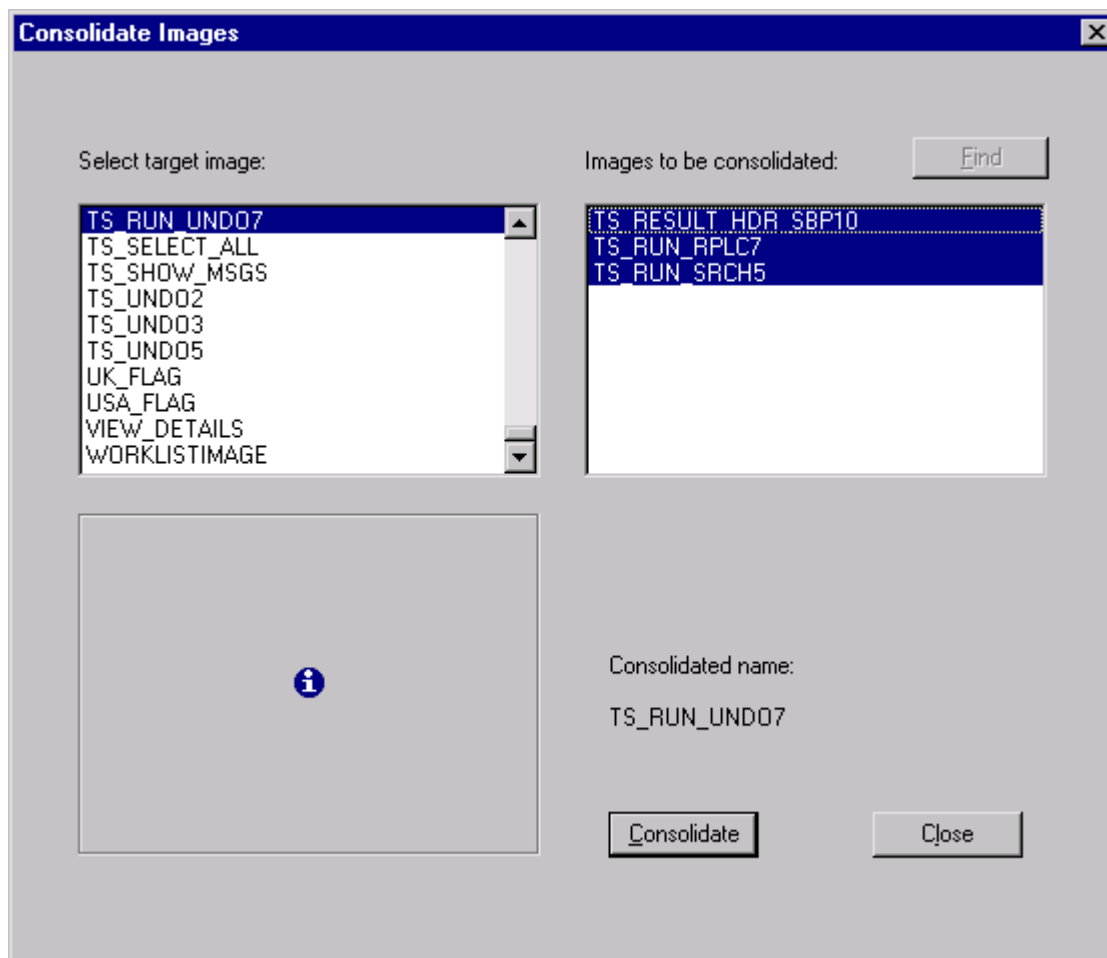
3. Select the fields to convert.
4. Click the Start button.

When the process is complete, a confirmation message appears in the Image tab of the output window.

---

## Consolidating Images

Use this tool if you create custom image definitions and use them in multiple record definitions. Consolidating images helps you avoid having the same image stored in several places. All image definitions included in the PeopleSoft system have already been consolidated.



Consolidate Images dialog box

To consolidate custom image definitions:

1. Open an existing image definition in the definition workspace.
2. Select Tools, Consolidate Images.
3. Select a target image from the list.
4. Click Find to gather all of the consolidated image candidates.
5. Select all of the images.
6. Click Consolidate.

The selected image definitions are removed from the Images to be consolidated list box and consolidated into the target image.

7. Click Close.

## Using the Image Catalog

In addition to the image definitions that you create, the system comes with many predefined image definitions. You can use these images to identify an action that a user must perform on a page or for aesthetic purposes only. For example, you can access an image either by inserting it onto a page as a static image or by specifying it as a label on a push button to accompany a specific function.

The following tables list some of the common image definitions in the image catalog that are used by PeopleTools in the deployment of internet applications. PeopleSoft does *not* recommend changing any of these images. You can also use Structured Query Language (SQL) to view a full list of different images:

```
SELECT DISTINCT CONTNAME FROM PSCONTDEFN WHERE CONTTYPE = 1.
```

For a list of language-dependent versions of images, use the following SQL:

```
SELECT CONTNAME, LANGUAGE_CD, CONTTYPE CONTFMT DESCR FROM  
PSCONTDEFNLANG WHERE CONTTYPE = 1
```

---

**Note.** Before logging into PeopleSoft Application Designer in a different language, the language support must first be enabled. Use PeopleTools Utilities to enable support.

---

In general:

- All PeopleTools image definitions start with *PT\_*.
- Any image definition with *\_D* at the end of the name is the disabled mode of the image.
- All images that start with *PT\_TAB* are used for tab construction and colors.

It is important to use these images consistently as you create new pages and update pages in the applications. The images are categorized by function and listed alphabetically by image name.

---

**Note.** This list contains a sample of the common images in PeopleSoft Application Designer, but it is not a complete list of PeopleTools images. For a complete list of images, select File, Open, Image. Click Open to see the full list of images.

---

### Toolbar

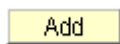
In previous releases, the entire toolbar button (Save, Next in List, and so on) was an image, including the text in the button. Currently, the button is an HTML button and the text is specified in HTML; only the icon on the button is an image. This enables you to change the color of the button using a style class. Toolbar buttons are rendered as HTML buttons, not images.

### Scroll or Grid Actions

Use these images when creating scroll areas, scroll bars, and grids.



PT\_ADD: Add a row button.



PT\_ADD\_LARGE: Large add a row button. Alternative to PT\_ADD if there is enough room on the page.



PT\_DELETE: Delete a row button.



PT\_DELETE\_LARGE: Large delete a row button. Alternative to PT\_DELETE if there is enough room on the page.



PT\_GRID\_NO\_TABS: Expand grid button. Expand grid columns horizontally so that grid tabs are no longer showing.



PT\_GRID\_SHOW\_TABS: Tabbed grid button. Return an expanded grid to a tabbed grid.



PT\_NEXTROW: View the next row button.



PT\_PREVIOUSROW: View the previous row button.

### ***Tab Images***

Use these images to change the look of both folder and grid tabs.



PT\_TAB\_LSCROLL: Scroll left to the previous tab in a page or grid.



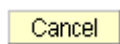
PT\_TAB\_RSCROLL: Scroll right to the next tab in a page or grid.



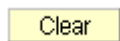
PT\_TABxxx through PT\_TABxxxxxxxx: Selection of over 80 images that can be used for creating folder tabs.

### ***Look-up and Search Page Images***

Use these images on look-up and search pages.



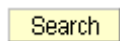
PT\_CANCEL: Cancel a look-up page.



PT\_CLEAR: Clear search criteria from the page (appears only on the Advanced Search/Lookup page).

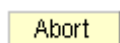


PT\_LOOKUP: Begin looking up data.

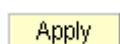


PT\_SEARCH: Start a search on the search page.

### ***General***



PT\_ABORT: Abort a transaction. Used in some PeopleCode message boxes.



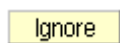
PT\_APPLY: Applies changes that were made to the database. Usually found when transferring to another page through PeopleCode.



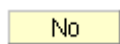
PT\_CALENDAR: Opens the calendar prompt.



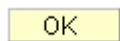
PT\_COPYURL: Used in the pagebar. Copies contents of address bar to the clipboard.



PT\_IGNORE: Used in some PeopleCode message boxes.



PT\_NO: Used in some PeopleCode message boxes.



PT\_OK: Accept the input on a secondary page and return to the main page.



PT\_POPUP: Opens a pop-up menu from which the user can select a page to transfer to.



PT\_POPUP\_SKIP: Use if a pop-up menu has only one menu item to select from, so the user is brought directly to the page.



PT\_PRINT: Open the print dialog box for the designated item.



PT\_PROCESSING: Appears as flashing text while accessing a server or database.



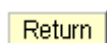
PT\_PROMPT\_LOOKUP: Look-up button for a field prompt table.



PT\_SPELLCHECK: Opens the spell check page for the current field.



PT\_REPORT\_DIST\_ICN: Usually found on the Process Scheduler Request page, this button takes you to the Distribution Detail page. Use to select roles and users who can access or receive an email of report output.



PT\_SEC\_RETURN: Return from a secondary page to the main (calling) page.



PT\_WF\_ACTIVITY: Used for a subprocess of the business task for workflow, consisting of one or more steps.



PT\_WF\_BUSPROC: Depict a complete business task for workflow.



PT\_WF\_STEP: Show a discrete step in the business process, corresponding to a single transaction that is performed on an application page or through an external program.

## Query Images



PT\_QUERY\_ADD\_CRITERIA: Add criteria to a query.



PT\_QUERY\_KEY: Denotes a query key field.



PT\_QUERY\_SORT: Sort query fields alphabetically.

## Tree Images

Use these tree images with PeopleSoft Tree Manager.

Tree node and leaf images include:



PSTREEMGR\_COL\_BRANCH: Identifies the root node for a branch.



PSTREEMGR\_COL\_NODE: Standard node image.



PSTREEMGR\_COL\_SKNODE: Tree node that has skipped a level.



PSTREEMGR\_END\_NODE: Tree node that has no children; that is, a terminal node.



PSTREEMGR\_EXP\_BRANCH: Root node on a branched tree.



PSTREEMGR\_EXP\_NODE: Expanded node.



PSTREEMGR\_EXP\_SKNODE: Expanded node that has skipped a level.



PSTREEMGR\_LEAF: Lowest-level detail value of a tree.

Node and leaf command button images include:



PTTREE\_ADDCHILD: Add a new child node.



PTTREE\_ADDLEAF: Add a new detail value, such as a leaf.



PTTREE\_ADDSIB: Add a new node as a sibling to the selected node.



PTTREE\_CUT: Cut a node or leaf and place it on the clipboard.



PTTREE\_DELETELEAF: Delete a leaf.



PTTREE\_DELETENODE: Delete a node.



PTTREE\_DISP\_AS\_ROOT: Redisplay the tree, starting with the currently selected node being displayed as the root node.



PTTREE\_EDITDATA: Display the underlying user data page.



PTTREE\_MAKEBRANCH: Create a new tree branch, starting with the selected node.



PTTREE\_PASTECHILD: Paste the node on the clipboard as a child of the currently selected node.



PTTREE\_PASTESIB: Paste the node on the clipboard as a sibling of the currently selected node.



PTTREE\_UNBRANCH: Unbranch a branch.



PTTREE\_UPDATELEAF: Update the values and properties for a leaf.



PTTREE\_UPDATENODE: Update the value or properties for a node.

### ***Portal Images***

Use these images to perform a specific action. Additional portal images, such as PT\_PORTAL\_HEADER\_BG, are purely aesthetic. In general, all images that are reserved for the portal contain the word *PORTAL* in the image name, such as PT\_PORTAL\_SEPARATOR.



PT\_PORTAL\_DOWN\_ARROW\_Y: Move a pagelet down when personalizing the portal layout.



PT\_PORTAL\_IC\_CLOSE: Remove a pagelet from the homepage.



## Chapter 17

# Creating HTML Definitions

This chapter provides an overview of HTML definitions and discusses how to:

- Create an HTML definition.
- Open an HTML definition.
- Reference HTML definitions dynamically.

---

## Understanding HTML Definitions

An HTML definition is a blank window where you can enter HTML. Use HTML definitions for storing long strings of HTML text or HTML text that you use repeatedly in your application. You can later access the definition dynamically from an HTML area control by using the GetHTMLText PeopleCode function.

See *Enterprise PeopleTools 8.50 PeopleBook: PeopleCode Language Reference*, "PeopleCode Built-in Functions," GetHTMLText.

You can reference HTML definitions from an HTML area control statically or dynamically. However, the most common usage is to reference HTML definitions dynamically.

---

## Creating an HTML Definition

To create an HTML definition:

1. Select File, New.

The New Definition dialog box opens enabling you to select the definition you want.

2. Select *HTML* from the scroll area.
3. Click OK to open a new HTML definition.
4. Type or paste the HTML text directly into the open definition window.
5. Select File, Save to save the HTML definition.

The HTML definition is saved in the current database.

---

## Opening an HTML Definition

To open an HTML Definition:

1. Select File, Open to access the Open Definition dialog box.
2. Select *HTML* from the Definition drop-down list box.
3. Click Open to show all of the HTML definitions that are currently stored in the database.

You can also enter a letter or word in the Name edit box to filter the display. For example, entering the letter *H* reveals all of the HTML definitions that begin with *H* in the database.

Naming a group of HTML definitions with the same introductory phrase can help keep them together for easy selecting.

---

## Referencing HTML Definitions Dynamically

To reference an HTML definition dynamically:

1. Add an HTML area control to a page definition.
2. Open the HTML Area Properties dialog box.
3. Select the *Constant* option.
4. On the HTML tab, enter the GetHTMLText PeopleCode function for the HTML definition that you want to reference.
5. Click OK.

### See Also

[Chapter 9, "Using Page Controls," Populating an HTML Area, page 242](#)

[Chapter 11, "Creating Page Definitions," Adding Page Controls, page 305](#)

*Enterprise PeopleTools 8.50 PeopleBook: PeopleCode Developer's Guide*, "PeopleCode and PeopleSoft Pure Internet Architecture," Using HTML Definitions and the GetHTMLText Function

## **Part 3**

# **Maintaining Data**

### **Chapter 18**

#### **Constructing File Layouts and Performing Data Interchanges**

### **Chapter 19**

#### **Using Command Line Parameters**

### **Chapter 20**

#### **Using Change Control**



## Chapter 18

# Constructing File Layouts and Performing Data Interchanges

This chapter provides an overview of file layouts and discusses how to:

- Construct file layouts.
- Specify file layout, record, and field properties.
- Perform data interchanges.
- Produce a sample file layout.

---

## Understanding File Layouts

This section discusses:

- File layouts.
- Field formats in a flat file.
- Fixed-position (FIXED) format files.
- Comma-separated value (CSV) format files.
- XML-tagged (XML) format files.

## Understanding File Layouts

A file layout is a definition that represents the data structure of a flat (text) file to be processed. When reading from or writing to a flat file, the PeopleCode can use a file layout as a template to identify and correctly process the file's records and fields. File layouts work with hierarchical and nonhierarchical data, and they can handle files that combine data records with nondata (audit or control) records.

To access data in a file, you don't have to create a file layout. PeopleTools supports reading and writing to plain text files, as well as to files that have a format based on a file layout.

- If the file is a plain text file, data is read or written by using text strings.
- If the file is based on a file layout, you can use text strings, rowsets, or record objects.

Using a file layout greatly simplifies reading, writing, and manipulating hierarchical transaction data with PeopleCode. It can facilitate transactions between a PeopleSoft application and a third-party system when the third-party system doesn't support PeopleSoft Integration Broker or component interfaces.

---

**Note.** Do not create or save files in Excel and then use those files for file processing.

File layouts do not have any related-language processing abilities.

---

### ***Applying File Layouts to Data Interchanges***

In addition to manipulating transaction data, you can employ file layouts to move data between a PeopleSoft database and external systems (data interchange) by using flat files as the transmission medium. File layouts enable you to:

- Export hierarchical PeopleSoft data to several flat file formats.
- Map incoming hierarchical file data directly to PeopleSoft tables.
- Preview and troubleshoot the input data and its formatting before importing it.
- Automatically generate the Application Engine and PeopleCode programs that are needed to perform data import.
- Use batch processes to perform large-volume data imports and exports.

### **See Also**

Chapter 18, "Constructing File Layouts and Performing Data Interchanges," Performing Data Interchanges, page 487

*Enterprise PeopleTools 8.50 PeopleBook: PeopleCode API Reference, "File Class"*

## **Field Formats in a Flat File**

A flat file, in the simplest sense, is a collection of fields in text format. The file must be formatted in a way that enables the PeopleCode to locate each field. PeopleSoft file layouts support three formats:

- **FIXED:** Each field has a starting position and a length that together specify its location in the file.

This is the default format for new file layouts.

- **CSV:** Fields are located in the file by their sequence, separated by commas.
- **XML:** A field is located not by its position or sequence within a record, but by the named XML tags surrounding it.

To preserve relationships between fields, you group fields into logical collections. In relational databases, these collections are records. Each line within a file is a collection of fields.

---

**Note.** With some file formats, the logical concept of a record may actually span multiple physical lines, but the concept of collections of fields remains.

---

The conceptual structure of all three file formats is represented in a file layout as follows:



### File layout structure

A file layout is a collection of file records, which in turn are a collection of fields, each of which has a describable location.

---

**Note.** To avoid confusion with the standard terms *record* and *field*, when working with a file layout we refer to a collection of fields as a *file record* and to fields as *file fields*.

---

## FIXED Format Files

This is the most common type of flat file that is currently processed by PeopleSoft EDI Manager. Almost all electronic data interchange (EDI) type processing uses this file type where each data element is oriented by a fixed, or column-dependent, position within the file.

The following table describes the properties of FIXED format files:

<i><b>Property</b></i>	<i><b>Description</b></i>	<i><b>PeopleSoft EDI Manager Equivalent</b></i>
File Layout Format	File format (FIXED).	None.
File Record ID	A group of numbers that can be used to identify the file record.	RowID.
ID Start Position	The column starting position of the file record ID.	Treated as a field within each map.
(Record) ID Length	The length of the file record ID.	Treated as a field within each map.
File Record Name	A user-specified name for the file record.	PeopleSoft record name.
File Field	A user-specified name for the file field.	PeopleSoft record's field name.
(Field) Start Position	The column starting position of the file field.	Starting position.
Field Length	The length of the file field.	Length of the field.

<b>Property</b>	<b>Description</b>	<b>PeopleSoft EDI Manager Equivalent</b>
Field Format	The formatting options for both inbound and outbound field processing.	Based on the field type.

### **Considerations for FIXED Format Files**

Consider the following when working with FIXED format files:

- Be careful when you change the length or starting position of any file fields, or when you insert a new file field between two existing ones.

It's possible to overlay fields. Results are unpredictable.

- When you insert a record into a file layout, fields of the Long type are converted to the Character type, with a length of 0.

You must set the field length to a number greater than 0 before you can save the file layout.

## **CSV Format Files**

In this type of file, each data element is surrounded with a separator, a delimiter, or both. File record IDs can be used to determine which table data is moved to or from; however, in most cases this type of file contains homogenous records.

In the following example, the qualifier is a double quote (") and the delimiter is a comma (,):

"NAME " , " ADDRESS " , " PHONE "

File layout definitions store the file record ID (when used) and the relative sequence number of the field. In the preceding example, "PHONE" is sequence number 3.

The following table describes the properties of CSV format files:

<b>Property</b>	<b>Description</b>	<b>EDI Manager Equivalent</b>
File Format	File format (CSV).	None.
File Record ID	A group of numbers that can be used to identify the file record.	RowID.
ID Sequence Number	The sequence number of the field that contains the file record ID.	Treated as a field within each map.

<i>Property</i>	<i>Description</i>	<i>EDI Manager Equivalent</i>
Qualifier	The character that surrounds a field to mark its extent. This can be set at the file layout, file record, or file field level.	Delimiter.
Delimiter	The character used to separate fields from each other. This can be set at the file layout or file record level.	Separator.
File Record Name	A user-specified name for the file record.	None.
File Record Field	A user-specified name for the file field.	None.
Field Format	The formatting options for both inbound and outbound field processing.	Based on the field type.

### ***Considerations for CSV Format Files***

Consider the following when working with CSV format files:

- Both the qualifier and the delimiter accept a blank as a valid value.
- If a field is null, you don't have to use qualifiers.

In the following example, Field2 is null:

```
Field1,,Field3,Field4. . .
```

- The system automatically pads the decimal places with zeros when amounts do not have values beyond the decimal point.

For example, 100 is automatically written as 100.00.

### ***Parsing Support for CSV Format Files***

The following list summarizes parsing support in preview and runtime for all CSV file types:

- All blank lines are ignored.  
In the past, blank lines came in as empty records.
- All empty fields are returned as empty strings or zeros.
- Fields, including commas or new lines, are surrounded in quotes.

- When a field is in quotes, any embedded quotes must be escaped by two quotes.
- When non-Microsoft Excel format backslashes are embedded within a string, they must be double-backslashes to be interpreted as a backslash; otherwise, they are interpreted as an escape character.

### ***Parsing Support for Microsoft Excel CSV Format***

The following table summarizes the differences between Microsoft Excel CSV parsing and non-Microsoft Excel CSV parsing:

<b><i>Item</i></b>	<b><i>Microsoft Excel CSV Parsing</i></b>	<b><i>Non-Microsoft Excel CSV Parsing</i></b>
White space	Leading and trailing white space is significant for strings.	Leading and trailing whites paces are ignored, including converted tabs.
Backslashes	Not interpreted as special characters, except for new lines.	Interpreted.
Quotes inside quoted strings	Double-quoted.	Escaped.
Tabs	Converted to a single space.	Interpreted as tabs and ignored.

## **XML Format Files**

This type of file contains data that is represented in a hierarchical or tree-type structure. A tag surrounds each data element. A file record tag might group multiple entries.

File layout definitions tie the identifier along with parent and child relationships to the file record and file field.

There is no PeopleSoft EDI Manager equivalent for this format.

The following table describes the properties of XML format files:

<b><i>Property</i></b>	<b><i>Description</i></b>
File Format	The file format (XML).
File Record ID	The tag name representing the file record.
Field Identifier	The tag name representing the file field.
Field Format	The formatting options for both inbound and outbound field processing.
File Record Name	A user-specified name for the file record.

<i>Property</i>	<i>Description</i>
Field Name	A user-specified name for the file field.

### **Considerations for XML Format Files**

Consider the following when working with XML format files:

- The XML input file must contain *at least* the fields that are specified in the file layout definition that you're using.

If the file is missing any fields, the input rowset doesn't contain any data.

- If the XML input file contains extra fields that aren't specified in the file layout definition, the ReadRowset method ignores the extra fields.
- When you insert a record into a file layout, fields of the Long type are converted to the Character type, with a length of 0.

You must set the field length to a number greater than 0 before you can save the file layout.

- The system automatically pads the decimal places with zeros when amounts do not have values beyond the decimal point.

For example, 100 is automatically written as 100.00.

- When importing XML files, the longest tag name supported is thirty characters. This is only an issue when importing XML files, as that limit is automatically enforced by PeopleSoft Application Designer.

---

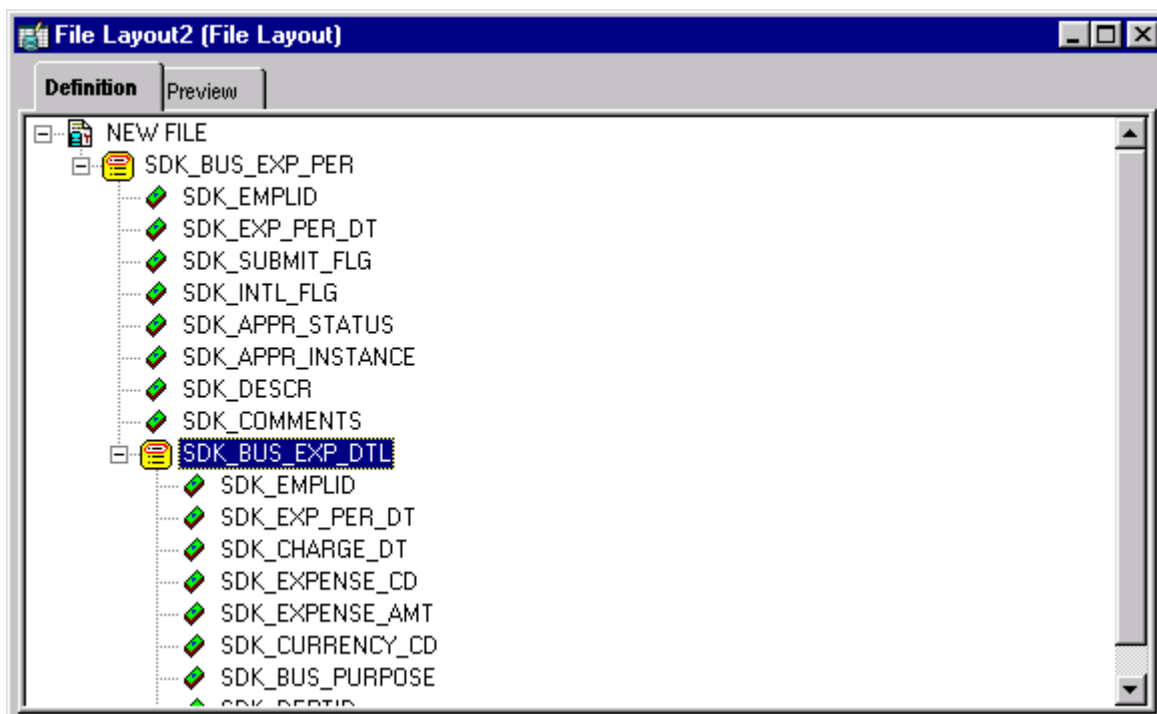
## **Constructing File Layouts**

This section discusses how to:

- Create a new file layout definition.
- Add file records and file fields.
- Reorder file layout nodes.
- Name file layouts, records, and fields.
- Apply field formats.

### **Creating a New File Layout Definition**

In PeopleSoft Application Designer, select File, New, File Layout.



File layout with two records in a hierarchy

A new file layout contains only one element—the root node, to which file records are attached. The default root node name is NEW FILE.

Save the new file layout definition. You're prompted to name the file layout, which also becomes the name of the root node.

---

**Note.** The default file layout format is FIXED.

---

### See Also

[Chapter 18, "Constructing File Layouts and Performing Data Interchanges," Specifying File Layout Properties, page 478](#)

## Adding File Records and File Fields

You can add file records and file fields to your file layout in two ways:

- Base them on existing record and field definitions in the database.
- Construct them directly in the file layout, without reference to any database records or fields.

A file record that you construct in this way is added as a *segment*, but the result is an ordinary file record.

You can use both methods in any combination when creating file records and file fields in a file layout definition. Each file record and file field is generically referred to as a *node*.

---

**Important!** Regardless of the method that you use, file records and file fields exist only as definitions within a file layout and have no connection with any database record or field. Even with file records based on records in the database, a change to the database record definition is *not* reflected in the file layout.

After any update to PeopleTools, we recommend all file layout definitions be regenerated.

---

### ***Using a Segment Instead of a Record***

Suppose that in the file that is provided to you, some of the file records contain new data and must be inserted, while others contain data that updates existing data. You can add a segment with a single field (like `AUDIT_ACTION`) that indicates whether the file record is new or changed. When you process the file, you can use PeopleCode to look at this field and, based on its value, perform the appropriate action.

As another example, suppose that you want to include two fields from the `PERSONAL_DATA` table in the file, but not all of the other fields. You have two choices: insert the `PERSONAL_DATA` table and manually delete all of the unwanted fields, or insert a segment, name it `PERSONAL_DATA`, and then insert the two fields that you want.

### ***Using Segments in Data Interchanges***

If you're creating a file layout for a data interchange, you can use segments, but each file record must correspond to a record with the same name in the PeopleSoft database, and its file fields must have the same names and data formats as the database record's fields. The file record may contain a subset of the fields in the corresponding database record. It may also contain a subset or a superset of the fields that are provided in the corresponding file data.

### ***Adding File Records***

To add a file record to the file layout definition, use one of the following methods:

- Drag and drop a database record.

Drag a record definition from the Project window into the file layout's Definition window, dropping it on the root node or on any existing file record or file field. The new file record appears at the same level as the node that you dropped it on, following all of the other file records at that level. All of the record's constituent fields are inserted, as well.

- Insert a database record.

With the root node or a file record highlighted, select **Insert, Record**. When you select a record, the new file record appears following the highlighted file record, at the same level. All of the record's constituent fields are automatically inserted, as well.

- Insert a segment.

With the root node or a file record highlighted, select **Insert, Segment**, and enter a file record name. When you click **OK**, the new file record appears following the highlighted file record, at the same level.

- Insert a child segment.

With a file record highlighted, select **Insert, ChildSegment**, and enter a file record name. When you click **OK**, the new file record appears one level below the highlighted file record, before any others at that level.

---

**Note.** When you add a file record at the root level, it appears immediately below the root node, before all of the other file records.

---

### ***Adding File Fields***

To add a file field to the file layout definition, use one of the following methods:

- Drag and drop a database field.

Drag a field definition from the Project window into the file layout's Definition window, dropping it on any existing file record or file field. Confirm the field name or enter a different one, and click OK. The new file field appears following the node that you dropped it on.

- Insert a database field.

With a file record or file field highlighted, select Insert, Database Field. Confirm the field name or enter a different one, and click OK. The new file field appears following the highlighted node.

- Insert a file field.

With a file record or file field highlighted, select Insert, FileField, and enter a file field name. When you click OK, the new file field appears following the highlighted node.

---

**Note.** Each file field must have a unique name within its parent file record, but file fields in different file records can have the same name.

---

## **Reordering File Layout Nodes**

The file layout definition provides a set of directional arrow buttons in the toolbar, which you can use to reposition any file record within the hierarchy of the file layout or any file field within its parent file record.

The up and down arrows don't change the level of the selected item, just its order among other items at that level. The right and left arrows move the selected item lower and higher in the file layout hierarchy.

---

**Note.** When you reposition a file record in the file layout, its child records are also repositioned, and their child records are repositioned, and so on.

---

## **Naming File Layouts, Records, and Fields**

File layout names can be 30 characters in length, and file record and file field names can be 15 characters in length. All names should follow PeopleSoft naming standards.

Each file record within a file layout must have a unique name, but one file record can have the same name as the file layout. Each file field within a given file record must have a unique name, but file fields in different file records can have the same name.

### ***Using WriteRecord, ReadRowset, and WriteRowset***

If you use the WriteRecord, ReadRowset, or WriteRowset file layout methods for writing to or reading from records, the application record and the file record *must* have the same name, and the application record fields and the file fields *must* have the same names. These methods write only to like-named records and like-named fields within a given record. If you rename a record or a field after you use it to create a file layout definition, you must rename the file record or file field to the exact same name.

In a file layout definition containing more than one record, records and fields that aren't like-named are ignored. Like-named records don't have to contain all of the same fields, and like-named fields don't have to be the same length. Like-named fields should, however, be of the same type.

### **See Also**

Chapter 6, "Creating Record Definitions," Naming Record Definitions, page 117

## **Applying Field Formats**

You must consider several issues when specifying field formats.

### ***Outputting Numbers to FIXED Files***

When you write numeric data to a FIXED format flat file, all numbers are right-justified in the file field. Numbers with decimal places specified are written with zeros padding the unused decimal places.

For example, a sequence of records with numbers of varying precision is written this way:

```
001      53.2700BUY
002   2174.0933SELL
003    108.0000SELL
```

### ***Date, Time, and Datetime Field Considerations***

In accordance with the International Organization for Standardization (ISO) 8601 standards, the field lengths for date, time, and datetime fields are fixed in the file layout, regardless of the file format:

- Date fields have a fixed length of 10.
- Time fields have a fixed length of 20.
- Datetime fields have a fixed length of 31.

---

**Note.** The PeopleSoft system complies with ISO 8601 guidelines for representing date field data. For example, while a hyphen (-) may be the date-time separator in your data, when exporting data the results will show a "T" in place of the hyphen. Refer to the ISO 8601 guidelines for more information.

---

### ***Considerations for Using Dates with the ReadRowset Method***

Single digits in dates in the form MMDDYY or MMDDYYYY must be padded with zeros. That is, if the date in the data is February 3, 2002, the format must be:

- 02/03/2002
- 02/03/02

The format 2/3/02 is *not* valid.

When using the date format DDMMYY, single digits should also be padded with zeros and the month value should be in all capitals. For example, if the date in the data is February 2, 2003, the format must be 02-FEB-03; 2-Feb-03 is *not* valid.

### See Also

*Enterprise PeopleTools 8.50 PeopleBook: PeopleCode API Reference*, "File Class," File Layout Error Processing

---

## Specifying File Layout, Record, and Field Properties

This section discusses how to:

- Specify file layout properties.
- Specify file record properties.
- Specify file field properties.

---

**Note.** Some properties are available only for a specific file layout format. For example, a file definition tag is available only for an XML format file. When a property is available only for a particular format, the format is noted in parentheses after the name of the property (such as File Definition Tag [XML]).

---

**Note.** Each property that appears in the Preview tab of the File Layout Designer is only another view of the corresponding property of the same name that appears in the Properties dialog box when working with the Definition tab. Changing any property value in one view, changes the same property value in the other view.

---

## Specifying File Layout Properties

The File Layout Definition Properties dialog box contains all of the information that is stored at the file layout (root) level.

Access the dialog box by using one of the following methods:

- Select File, Object Properties.
- Press Alt+Enter.
- Double-click the topmost (root) node of a file layout definition.
- Right-click an open file layout, and select Data Object Properties.

The General tab of the dialog box contains description information for the file layout. The Use tab contains specific information for the file layout.

The image shows a Windows-style dialog box titled "File Layout Definition Properties" with a blue title bar and a close button (X) in the top right corner. The dialog has two tabs: "General" and "Use", with "Use" currently selected. Inside the "Use" tab, there are several input fields and checkboxes. The "File Layout Name" field contains "PSPING\_FILELAYOUT". The "File Layout Type" field contains "CSV". Below these is a "File Layout Format" dropdown menu currently set to "CSV". Further down is a "File Definition Tag" text box, which is empty. Below that is a "Buffer Size" text box containing the number "0". There is a checkbox labeled "Imply Decimal Place" which is unchecked. Below this is a section labeled "Excel Format:" with an unchecked checkbox. Underneath are four more checkboxes: "Qualifier Optional" (unchecked), "Convert Tabs to Spaces" (unchecked), "Interpret Backslash" (unchecked), and "Strip White Space" (unchecked). At the bottom of this section is a label "Embedded Strings Interpretation:" followed by two radio buttons: "Double Quote" (selected) and "Backslash Quote" (unselected). At the very bottom of the dialog are "OK" and "Cancel" buttons.

File Layout Definition Properties dialog box - Use tab

**File Layout Format** Displays the type of file layout. Values are *FIXED*, *CSV*, and *XML*.

**File Definition Tag** Enter the XML tag name that is associated with this layout (or transaction). This tag can be 30 characters in length. This tag must be unique in the file layout.

This field is active when the file layout format is XML.

---

**Note.** If you leave this value blank, the system inserts `<start>` into the file as the default tag value. Enter any value to prevent the system from inserting the default `<start>` tag into the file.

---

**Buffer Size** Displays the size of the input buffer that is used at runtime.

This field is active when the file layout format is XML.

---

**Note.** Do not edit this value directly.

---

<b>Imply Decimal Place</b>	<p>Select to automatically add a decimal place based on the decimal value that is specified in the field definition when importing a file into a PeopleSoft database and no decimal place is specified for numeric values in the file.</p> <p>If you select this option, make sure that the length that you specify for the Number field includes the decimal point, even though it is not visible. In this case, the decimal point is counted as a space. For each occurrence of numeric data that is missing a decimal point character in the input file, a space character is also required in the input file (either immediately before the numeric characters or immediately after them).</p> <p>This field is active when the file layout format is FIXED.</p>
<b>Excel Format</b>	<p>Select to parse the field by using Microsoft Excel CSV specifications.</p> <p>This field is active when the file layout format is CSV.</p>
<b>Qualifier Optional</b>	<p>Select to parse the field based on delimiters. When a qualifier exists, the field is parsed by using a closing matched qualifier. All characters are ignored before and after the qualifier.</p> <p>This field is active when the file layout format is CSV.</p> <p>If you do not select this check box, the field <i>must</i> have qualifier pairs. All characters are ignored before and after the qualifier.</p>
<b>Convert Tabs to Spaces</b>	<p>Select to convert all tabs within the field to spaces. If you do not select this option, all tabs are ignored.</p> <p>This field is active when the file layout format is CSV.</p> <hr/> <p><b>Note.</b> If the Strip White Space option is also selected, the system removes trailing and leading tabs.</p> <hr/>
<b>Interpret Backslashes</b>	<p>Select to interpret backslashes within field data as escape characters.</p> <p>The following characters have special meanings:</p> <ul style="list-style-type: none"> <li>• \t represents a tab.</li> <li>• \n represents a new line.</li> </ul> <p>If you do not select this check box, backslashes are considered regular text data.</p> <p>This field is active when the file layout format is CSV.</p>
<b>Strip White Space</b>	<p>Select to remove all leading and trailing white space within field data from the output data.</p> <p>This field is active when the file layout format is CSV.</p> <hr/> <p><b>Note.</b> If you select this option, trailing and leading tabs are also removed, regardless of whether you have selected the Convert Tabs to Spaces option.</p> <hr/>

**Double Quote**

Select to interpret two double quotes (" "), as one double quote (") in output data.

For example, if you enable this option, `abc "def "` is interpreted as `abc"def .`

This field is active when the file layout format is CSV.

**Backslash Quote**

Select to interpret backslash quotes (\") as one double quote (") in output data.

For example, if you enable this option, `"abc\"def "` is interpreted as `abc"def .`

This field is active when the file layout format is CSV.

**See Also**

Chapter 18, "Constructing File Layouts and Performing Data Interchanges," Understanding File Layouts, page 467

## Specifying File Layout Segment Properties

The File Layout Segment Properties dialog box contains information that is stored at the file record level.

The image shows a Windows-style dialog box titled "File Layout Segment Properties". It has a tab labeled "Use" which is currently selected. Inside the dialog, there are several input fields and a dropdown menu. The "File Layout Name" is set to "PSPING\_FILELAYOUT" and the "File Layout Type" is set to "CSV". The "File Record Name" field contains "TP\_TST\_CASE\_VW". The "ID Seq No" is "1", "Max Rec Length" is "106", "File Record ID" is empty, "ID Start Position" is "0", "ID Length" is "0", "Default Qualifier" is empty, "Seg Terminator" is empty, "Field Delimiter" is set to "Comma" (with a dropdown arrow), "Record Tag" is "PTP\_TST\_CASE\_V", and "Record Description" is empty. At the bottom right, there are "OK" and "Cancel" buttons.

File Layout Segment Properties dialog box - Use tab

Access the dialog box by using one of the following methods:

- Double-click the file record node.
- Right-click the file record node, and select Selected Node Properties.

**File Record Name** Enter a file record name that is associated with this file record. This name is used when accessing the file record from PeopleCode. Every file record in a file layout must have a unique name.

**ID Seq No. (CSV)** Enter a sequence number for the field that contains the file record ID.  
(identification sequence number)

<b>Max Rec Length</b> (maximum record length)	Displays the default maximum length of the combined field sizes of the record. This value is automatically updated.
<hr/> <b>Warning!</b> Any inbound or outbound data is truncated beyond this value. <hr/>	
<b>File Record ID</b>	Enter a number to uniquely identify the file record in the file layout. You can use this number in processing the file. This number is automatically written to the file if you use the WriteRecord or WriteRowset methods and if the file type is FIXED or CSV.
<b>ID Start Position (FIXED)</b>	Enter the column or starting position in the file record where the file record ID starts.
<b>ID Length (FIXED, CSV)</b>	Displays the length of the file record ID. This number is automatically generated when you enter the file record ID.
<hr/> <b>Note.</b> This value shouldn't be edited directly. <hr/>	
<b>Default Qualifier (CSV)</b>	Enter a qualifier that is used for the file record ID and as the default for fields when no field qualifier is specified. This value overrides the definition qualifier that is specified in the File Layout Definition Properties dialog box. When you first create a file layout, this property is blank.
<b>Seg Terminator (FIXED, CSV)</b>	Enter characters to be placed at the end of each segment when the layout is written to a file.
<b>Field Delimiter (CSV)</b>	Enter a delimiter that is used for all fields in the file record. This overwrites the definition delimiter that is specified on the File Layout Definition Properties dialog box.
<b>Record Tag (XML)</b>	Enter an XML tag name for this file record. The default value is the file record name.
<hr/> <b>Note.</b> Although each record name in a file layout must be unique, record tags do not have to be unique. <hr/>	
<b>Record Description</b>	Enter a description of the record for documentation purposes only.

## Specifying File Layout Field Properties

The File Layout Field Properties dialog box contains information that is stored at the file field level.

The image shows the 'File Layout Field Properties' dialog box with the 'Use' tab selected. The dialog is titled 'File Layout Field Properties' in a blue header bar. Below the header, there is a 'Use' tab. The main area contains several fields and checkboxes. At the top, 'Layout Name' is 'PSPING\_FILELAYOUT' and 'Sequence No' is '1'. Below that, 'Layout Type' is 'CSV'. The 'Field Name' is 'PTP\_TEST\_CASE\_ID'. 'Start Position' is '1' and 'Length' is '30'. 'Field Type' is 'Character'. There are checkboxes for 'Trim Spaces' (unchecked), 'Pad Field' (unchecked), and 'Upper Case' (unchecked). 'Qualifier' is an empty field. 'Decimal' is '0'. 'Field Tag' is 'PTP\_TEST\_CASE\_ID'. There is a 'Propagate' section with '<<<' and '>>>' buttons and a '0' in the middle. 'Strip Characters' is an empty field. 'Field Description' is a large empty text area. 'Default Value' is an empty field. 'Field Inheritance' has 'Record' set to 'DO-NOT-INHERIT' and 'Field' is an empty dropdown. At the bottom are 'OK' and 'Cancel' buttons.

File Layout Field Properties dialog box - Use tab

Access the dialog box by using one of the following methods:

- Double-click the file field node.
- Right-click the file field node, and select Selected Node Properties.

Most individual properties are usable by all field types. However, some are specific to a particular field type—for example, the Upper Case check box is applicable for character fields only, while the Date Separator field is applicable for date fields only, and so on. The preceding dialog box shows the properties for a character type of field. However, the following descriptions include all possible properties.

**Date Format (Date)** Select a date format, such as MMDDYY, DDYYMM, and so on.

**Date Separator (Date)** Enter a character that is used to separate date values. The default value is /.

<b>Decimal</b>	<p>Enter the number of decimal positions (to the right) of the decimal point. This property is valid only for fields that are defined as number or signed number.</p> <hr/> <p><b>Note.</b> You're allowed only 31 characters plus a decimal point.</p> <hr/>
<b>Default Value</b>	<p>Enter the default value to be used by a call to ReadRowset when the corresponding data is imported from a file via a file layout definition.</p> <hr/> <p><b>Note.</b> The system variables related to date and time (for example, %DATE, %TIME, %DATETIME) cannot be used to specify the value of the default-value property of a file layout field.</p> <p>The value of the default-value property of a file layout field has no relationship to the value of the default-value property of any corresponding record field (one having the same record name and field name). When data is imported from a file via a file layout definition, the value used for a file layout field that is empty in the input file is determined by the default value of the corresponding file layout field and not the default value of any corresponding record field (even if the value of the default-value property of the file layout field is empty and even if that record field is required).</p> <hr/>
<b>Field Description</b>	Enter a description of the field for documentation purposes only.
<b>Field Inheritance</b>	<p>Optionally select a parent file record and field, from which the current field's value is to be inherited. If you're writing to a file, this means that the value is written only in the parent file record, not in the child (inheriting) file record (that is, the value isn't written more than once to a file). If no value is present in the parent field, the default value that is specified here is used.</p> <p>For example, the following file sample shows both the EMPLID (8113) and EFFDT (08/06/1999) written only once to a file, though these fields are repeated in the third file record (with a file record ID of 102):</p> <pre> 100 8113 Frumman,Wolfgang 101      08/06/1999      000001      219 Going to London office 102      100 000015 I 08/06/1999 102      200 000030 I 08/06/1999 102      300 000009 I 08/06/1999 102      400 000001 I 08/06/1999 102      500 000011 I 08/06/1999 </pre>
<b>Field Name</b>	Enter the name that is associated with this file field. This name is used when accessing the file field from PeopleCode. Every field within a file record must have a unique name; however, two different file records can contain the same file field.
<b>Field Tag (XML)</b>	<p>Enter an XML tag name to be used around the field. The default value is the name of the field.</p> <hr/> <p><b>Note.</b> Although each field name in a file record must be unique, each field tag does not have to be unique.</p> <hr/>
<b>Field Type</b>	Select the data type of the file field.

<b>Length</b>	<p>Enter the maximum number of characters of this field.</p> <hr/> <p><b>Note.</b> You're allowed only 32-character precision for number and signed number fields, that is, a total of 32 characters both to the right and left of the decimal. Other fields, such as character fields, can be longer.</p> <p>You can't set the field length for Date, Time, and Datetime type fields. These field lengths are automatically set to the ISO standards for such fields.</p> <hr/>
<b>Pad Field (CSV)</b>	Select to enable field padding. The field will be padded with space characters up to the value of the length property. Character data will be left justified, numeric data right justified.
<b>Propagate (FIXED)</b>	If a field position or length is changed, enter an amount here to increment (positive number) or decrement (negative number) the current field and all fields before it ( <<< ) or after it (>>>).
<b>Qualifier (CSV)</b>	Enter the qualifier for the field, that is, the character that surrounds this field, separating it from other fields. Specifying this value overwrites the value that is specified in the file layout properties and file record properties.
<b>Start Position (FIXED)</b>	<p>Enter the starting position (column) of the field within the file record.</p> <hr/> <p><b>Important!</b> If you specify a start position for a field that overwrites a previous field, no data is written to the file. Use Propagate to change the start positions for your file fields.</p> <hr/>
<b>Strip Characters</b>	<p>Specify any characters to be removed from the input buffer. Use this to preprocess input strings. For example, if a field in the input file contains hyphens, but you want to remove the hyphens before processing the field, you can enter a hyphen here, and it is stripped out while being read. You can specify more than one character to be stripped out. Be sure to not separate the strip characters. For example, the following strips out all semicolons and hyphens:</p> <pre>; -</pre> <p>The following strips out all semicolons, hyphens, and spaces:</p> <pre>; -</pre>
<b>Trim Spaces</b>	<p>Select to remove trailing spaces only from the input string. This option is active only for FIXED and CSV formats.</p> <p>This is different from the Strip Characters field, which removes all spaces from the entire input field if you specify a space.</p>
<b>UpperCase (Char)</b>	Select to convert lowercase text to uppercase during inbound processing. This is used primarily when customer data may be in lowercase and the PeopleSoft software requires the data to be in uppercase.

---

**Note.** Numbers that contain thousand separators (as in "," or ".") are not accepted by PeopleSoft. For example, if you exchange data with Microsoft Excel, you need to disable the "Use 1000 Separator (,)" property.

---

**See Also**

[Chapter 4, "Creating Field Definitions," page 61](#)

---

## Performing Data Interchanges

This section provides an overview of the import process and discusses how to:

- Preview input data.
- Generate and run the import program.
- Export data.

## Understanding the Import Process

To help you troubleshoot and import flat file data, the file layout definition provides a data preview page. It can also generate an Application Engine program with the associated PeopleCode that is necessary to import the data.

---

**Note.** The Application Engine program that the system automatically generates should be viewed as containing sample PeopleCode that you need to review and modify, especially when a single file layout involves multiple records.

---

### ***Data Import Activities***

Importing data by using a file layout requires the following sequence of activities:

1. Provide the import data in a properly formatted flat file.

Each record in the file must correspond to a record with the same name in the PeopleSoft database, and its fields must have the same names and data formats as the database record's fields. Each record in the file must end with a new line character.

2. Create a file layout definition to match the record and field structure of the data.

Insert the appropriate record definitions into the file layout, then reposition the file records and file fields to match the record and field positions in the file.

3. Preview and troubleshoot the input data format and content.
4. Generate the data import Application Engine program and PeopleCode.
5. Run the Application Engine program to import the data.

### ***Field Inclusion and Exclusion***

The fields in the data file's records can be a subset of the database record's fields—define your file layout with only the file fields that you expect to receive.

The fields in the data file's records can be a superset of the database record's fields; you must define your file layout to suppress or ignore the extra fields. For FIXED files, don't define a field at the corresponding position in the file layout definition. For CSV files, the file layout must have the same number of fields in each record as there are in the corresponding file record—for each field that you don't want to import, define a field in the file layout at that position that *doesn't* correspond to any field in that database record. For XML files, any extra fields are ignored automatically.

Your data file can contain a subset or a superset of the records that are defined in the file layout. Only a file record with a matching file record ID in the file layout is imported.

### ***Record Hierarchy***

In theory, you can ignore rowset hierarchy when importing file data, because the PeopleSoft database stores each record independently of the others, and rowsets aren't used in the import process. However, many records are designed with hierarchical dependencies in mind. The input file might omit inherited field values or order the data records in a way that reflects such dependencies.

If the input file omits inherited field values, make sure that the inheriting fields' records in the file layout are children of the ones from which they inherit their values, and make the appropriate Field Inheritance settings.

If the records to be imported contain key fields that reflect a rowset hierarchy, they might be in an order in the file that also reflects the hierarchy. Make sure your file layout reflects that hierarchy, as well.

---

**Important!** The completed file layout must have exactly one file record at the root level; all other file records must be below that level.

---

### **See Also**

[Chapter 18, "Constructing File Layouts and Performing Data Interchanges," Reordering File Layout Nodes, page 476](#)

## **Previewing Input Data**

In PeopleSoft Application Designer, select File, New, File Layout, Preview.

**SDK\_BUS\_EXP (File Layout)**

Definition **Preview**

Segment Name: **SDK\_BUS\_EXP\_PER** ☐ Trim Spaces

Default File Name: **H:\Work\Tint\8.40\FLO\_In\_01.txt**

	Field Name	Field Length	Start Position	Propagate	Field Type	Date Separator
1	SDK_EMPLID	10	16	<<< 0 >>>	Date	
2	<b>SDK_EXP_PER_DT</b>	Field Length			Date Format	
3	SDK_SUBMIT_FLG				MMDDYYYY	
4	SDK_INTL_FLG	Strip Characters			Description	
5	SDK_APPR_STATUS	Default Value				
6	SDK_APPR_INSTANCE	Field Inheritance				
7	SDK_DESCR	Record	DO-NOT-INHERIT			
8	SDK_COMMENTS	Field				

	SDK_EMPLID	SDK_EXP_PE	SDK_SUBMIT	SDK_INTL_	SDK_APPR	SDK_APPR_IIS	SDK_DESCR	SDK_COMM
1	7777	06/11/1989	N	N	P	0	Customer Go-Live Celebration	
2	7777	08/31/1989	Y	N	A	0	Customer Focus Group Meeting	
3	7777	03/01/1998	Y	Y	P	0	Attend Asia/Pacific Conference	
4	7777	05/29/1998	N	N	P	0	Annual Subscription	Subscription
5	7777	08/22/1998	N	N	P	0	Regional Users Group Meeting	Attending

File layout preview

Initially no information is displayed on the preview tab.

**Note.** With an XML file, the data can't be previewed, although you can still generate and run an import program.

To preview the input data:

1. Click Browse and select the import file.

The name of the import file appears in the Default File Name text box.

2. Select a file layout segment from the Segment Name drop-down list box.

The file layout fields corresponding to the selected segment appear in the list below the segment name. This list comes from the file layout definition, and does not depend on you selecting a valid import file.

- Click the Refresh button on the toolbar to refresh the preview data.

At the bottom of the window, a preview of the first five rows of the selected segment in the selected import file appears in the grid.

You can also select View, Automatically Read Preview Data. While this menu item is selected, the file layout refreshes the preview automatically; when it's cleared, you need to click the Refresh Preview Data button to see the effects of changes you make to the property settings or data.

---

**Note.** Preview file data is available only for FIXED and CSV file layouts. The preview data appears only if you've selected an import file, if the file format matches the format specified for the file layout definition, and if it contains a file record ID that matches the file record ID of the selected segment.

---

- Select one of the fields on the Field Name list.

The properties of the field that you selected appear in the appropriate display fields. These properties are actually the field's file layout field properties; you can change all field property settings from this view.

---

**Note.** Only the field properties that are appropriate to the file layout format that you specified in the File Layout Definition Properties dialog box are visible in this view; for example, the Start Position and Field Length fields are available only for a FIXED format file layout.

---

### ***Examining the Input Data for Errors***

The following table shows some examples of input data errors:

<b><i>Symptom</i></b>	<b><i>Possible Reason</i></b>	<b><i>Solution</i></b>
The preview grid doesn't appear.	The input file's format doesn't match the file layout format that you specified in the File Layout Definition Properties dialog box.  A CSV file doesn't use the definition qualifier you specified in the File Layout Definition Properties dialog box.	Change the file layout format to match the input file.  Change the definition qualifier to match the one that is used in the input file.
Only the first column of the preview grid is populated.	A CSV file doesn't use the definition delimiter you specified in the File Layout Definition Properties dialog box.	Change the definition delimiter to match the one that is used in the input file.
The preview grid appears for some records, but not for others.	The file's record IDs for the missing records don't match their file record IDs specified in the File Layout Segment Properties dialog box.	Specify file record IDs that match the input file records.

<b>Symptom</b>	<b>Possible Reason</b>	<b>Solution</b>
Data for a field appears truncated in the preview grid.	With a FIXED file, the field length that you specified in the File Layout Field Properties dialog box is too short to accommodate the field data.	Increase the field length to accommodate the input data.
A field appears to start in the middle of the data.	With a FIXED file, the start position you specified in the File Layout Field Properties dialog box is too great to include the start of the field data.	Decrease the start position and adjust the field length to match the input data start position and length.

## Generating and Running the Import Program

This part of the process imports the file data.

To generate and run the import program:

1. After you preview the file layout and examine the input data, click the AE button on the toolbar.

This generates the Application Engine import program with its associated import PeopleCode.

2. Enter a name for the Application Engine program, and click OK.

The program is automatically saved and is ready to run as soon as its definition appears in PeopleSoft Application Designer.

3. Click the Run Program button on the toolbar.

The file data is imported into the database.

## Exporting Data

The method that you use to export data from the PeopleSoft system depends on the target application's requirements. To export data to a flat file, you create a file layout definition, then write PeopleCode to transfer the data to a file. The PeopleCode can be initiated from Application Engine or from any event. It should populate text strings, rowset objects, or record objects, and it should apply the WriteRecord or WriteRowset method from the File class to transfer the data to the file by using the file layout definition to position the records and fields as required by the target application.

---

**Note.** To generate valid XML files, be sure to use the Close method in the File class when you finish writing to the file.

---

### See Also

*Enterprise PeopleTools 8.50 PeopleBook: PeopleCode API Reference, "File Class"*

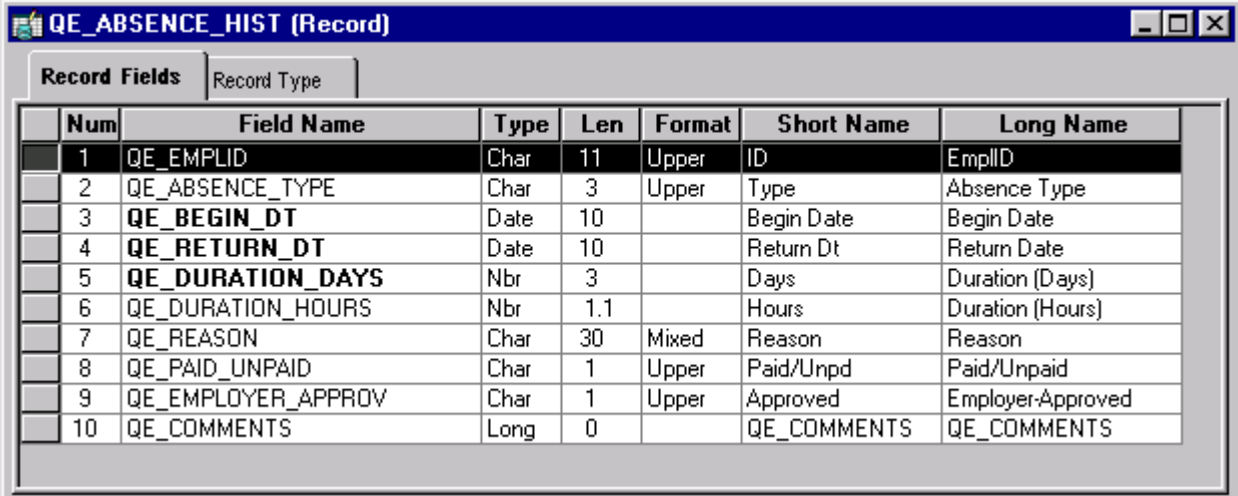
## Producing a Sample File Layout

This section describes how to:

- Create the file layout definition.
- Adjust file layout properties.
- Insert a segment and a field.

## Creating the File Layout Definition

This example illustrates how to create a file layout that can be used with the QE\_ABSENCE\_HIST record:



Num	Field Name	Type	Len	Format	Short Name	Long Name
1	QE_EMPLID	Char	11	Upper	ID	EmplID
2	QE_ABSENCE_TYPE	Char	3	Upper	Type	Absence Type
3	QE_BEGIN_DT	Date	10		Begin Date	Begin Date
4	QE_RETURN_DT	Date	10		Return Dt	Return Date
5	QE_DURATION_DAYS	Nbr	3		Days	Duration (Days)
6	QE_DURATION_HOURS	Nbr	1.1		Hours	Duration (Hours)
7	QE_REASON	Char	30	Mixed	Reason	Reason
8	QE_PAID_UNPAID	Char	1	Upper	Paid/Unpd	Paid/Unpaid
9	QE_EMPLOYER_APPROV	Char	1	Upper	Approved	Employer-Approved
10	QE_COMMENTS	Long	0		QE_COMMENTS	QE_COMMENTS

QE\_ABSENCE\_HIST record definition

For simplicity, suppose that each row in the FIXED format file has the following structure:

```

888 A
000 8001 VAC 1981-09-12 1981-09-26 14 .0 P Y
888 A
000 8001 VAC 1983-03-02 1983-03-07 5 .0 P Y
888 A
000 8001 VAC 1983-08-26 1983-09-10 13 .0 P Y
888 A
000 8516 MAT 1986-06-06 1986-08-01 56 .0 P Y
888 C
000 8516 SCK 1988-08-06 1988-08-07 1 .0 P Y
888 A
000 8516 VAC 1987-07-14 1987-07-28 14 .0 P Y
888 A
000 8553 JUR 1990-12-12 1990-12-17 5 .0 Local Jury Duty P N
888 A
000 8553 MAT 1992-02-20 1992-10-01 224 .0 Maternity Leave U N
888 A
000 8553 MAT 1994-08-19 1995-03-01 194 .0 Maternity U Y
888 A
000 8553 PER 1993-04-15 1993-04-19 4 .0 U N Personal
Day required
888 C
000 8553 SCK 1987-01-28 1987-01-30 2 .0 Hong Kong Flu P N
888 A
000 8553 SCK 1988-08-02 1988-08-03 1 .0 Sick P N
888 A
000 8553 SCK 1995-09-12 1995-09-13 1 .0 P N
888 C
000 G001 MAT 1991-07-02 1991-09-28 88 .0 3-month Maternity P Y Maternity will
be paid as 80% of Claudia's current salary.

```

000 is the file record ID for ABSENCE\_HIST, and each field appears in the same order as in the ABSENCE\_HIST database record. 888 is the file record ID for an extra segment called CHANGE\_ACTION, containing an AUDIT\_ACTION field with the following meanings:

- A: Row inserted.
- C: Row updated, but no key fields changed.

---

**Note.** The end of file (EOF) character must be on a separate line and not on a line containing data for any incoming file, regardless of the file type. Each data line needs to be terminated with an end of line (EOL) character, which is different from an EOF.

---

To create a file layout definition:

1. Use the QE\_ABSENCE\_HIST record definition as a template for the file layout.

Create a new file layout, then drag the QE\_ABSENCE\_HIST record into the open file layout.

2. Save the file layout.

Save it with a name of ABS\_HIST.

The name of the first node changes from NEW FILE to ABS\_HIST.

## Adjusting File Layout Properties

In the file layout definition, open the file layout, record, or field properties as appropriate for each step in the following procedure.

To adjust layout, record, and field properties:

1. Change the file layout properties.

Double-click the topmost node in the file layout, `ABS_HIST`, to display the file layout properties.

Enter a short and long description of the file layout you're creating. For this example, we're creating a `FIXED` file layout, so you don't need to make any changes on the `Use` tab.

2. Change the file record properties.

Double-click the `QE_ABSENCE_HIST` file record to display its properties. Enter a record ID of `000`, and a starting position of `1`. The ID length is automatically set.

When you click `OK`, you see a message asking if you want to increment the start positions for all fields.

Click `Yes`. This automatically increments the start position numbers for every field to take the length of the file record ID that you just added into account. If you don't click `Yes`, you must manually increment the start position for all fields.

You've just created the file record ID for the `QE_ABSENCE_HIST` record:

000	8001	VAC	1981-09-12	1981-09-26	14	.0	P	Y
000	8001	VAC	1983-03-02	1983-03-07	5	.0	P	Y

3. Change the file field properties.

When a record definition is used as a template for a file layout, the default starting position for each field is based on the order in which it appears in the record as well as its length.

Double-click the `QE_EMPLID` file field to display its properties.

The start position is automatically incremented to 4 (because the file record ID is three characters long). However, in the example there's an extra space between the end of the file record ID and the first field. Therefore, you need to change the start position of this field and all of the fields after this field. This increments the starting position of this field and all fields following this field by 1.

To do this:

- Click the up arrow under `Propagate` to change that number from 0 to 1.
- Click the button with the arrows pointing right (`>>>`).

4. (Optional) Adjust other fields.

The last field (QE\_COMMENTS) has a length of 0 because it's based on a Long field type.

---

**Note.** When a Long field is inserted into a file layout, it's converted to a Character field with a length of 0.

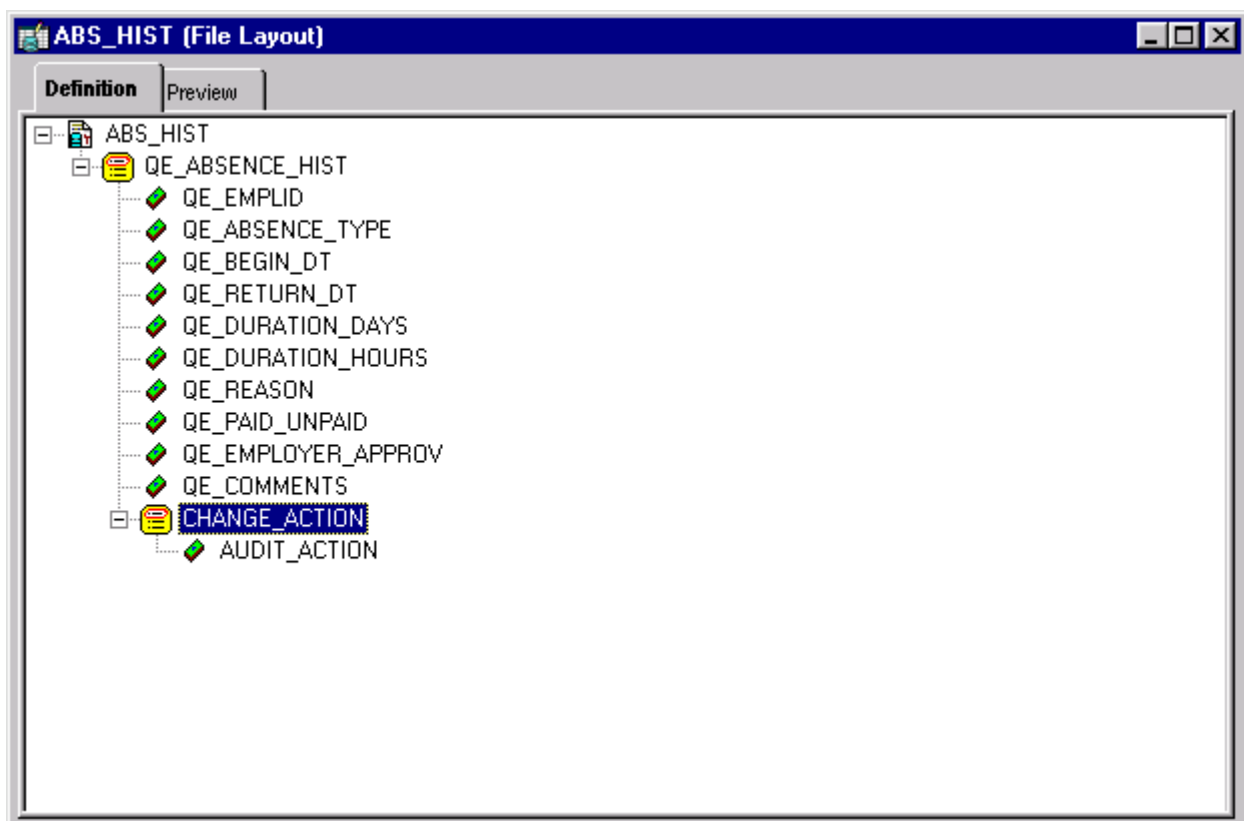
---

Because the format for this file layout is **FIXED**, you have to change the field length of that file field so it's long enough to accommodate the data that you expect from the file.

You don't have to propagate this change because this is the last field in the record.

## Inserting a Segment and a Field

Open the file layout definition based on the ABS\_HIST record.



File layout with new segment and field added

To insert a segment and a field in that segment:

1. Insert a segment.

Insert a segment that is a sibling, (that is, at the same level), as the QE\_ABSENCE\_HIST record. Insert a segment by selecting Insert Segment from the pop-up menu, or by selecting Insert, Segment. The Insert New Segment dialog box appears.

This dialog box is identical to the File Layout Segment Properties dialog box. Enter the file record name, file record ID, and ID start position. When you click OK, the segment is inserted.

---

**Note.** If you provide a Record File ID when adding a segment to a file layout that is either CSV or fixed, the system prepends this Record File Id to every row of data that gets written.

---

2. Insert a file field.

Insert a file field by selecting Insert FileField from the pop-up menu. The Insert New Field dialog box appears. It's identical to the File Layout Field Properties dialog box. Enter the field name, start position, and field length; then click OK.

---

**Note.** The start position isn't automatically set when you add a file field to the file record; for this example the AUDIT\_ACTION field requires a start position of 5.

---

3. Save your work.

Be sure to save the changes that you've made to the file layout by selecting File, Save, or clicking the save icon in the toolbar.

Now that you've created and saved a file layout, you must use PeopleCode to access the data. File layouts rely solely on PeopleCode as the engine behind the actual data access and movement.

### See Also

*Enterprise PeopleTools 8.50 PeopleBook: PeopleCode API Reference, "File Class"*

## Chapter 19

# Using Command Line Parameters

This chapter provides an overview of PeopleSoft Application Designer command line parameters and discusses how to:

- Start PeopleSoft Application Designer and PeopleSoft Data Mover.
- Build projects.
- Copy definitions.
- Compare definitions.

---

## Understanding PeopleSoft Application Designer Command Line Parameters

PeopleSoft Application Designer offers a variety of command-line parameters that you can use to control the database to which it connects. By using these parameters, you can automatically navigate to the part of the system that you need.

Use the values from the following tables when specifying certain parameters.

### ***Syntax***

The following command line syntax applies to all actions identified in this chapter:

PSIDE [-parameter value [-parameter value . . .]]

You can include as many or as few parameters as you need.

Each parameter starts with a hyphen (-) or a forward slash (/). The value for each parameter follows the hyphen or slash, separated by zero or more spaces. In general, the value doesn't need to have quotation marks around it, even if it has internal spaces—the system treats all text following the parameter as part of the value, up to the next parameter or the end of the command line.

---

**Note.** You must enclose a value in quotation marks only when it includes a hyphen or forward slash, or to include leading or trailing spaces. If the value itself includes a quotation mark character, precede the double quote with a backslash (\).

---

**See Also**

*Enterprise PeopleTools 8.50 PeopleBook: Data Management*, "Using PeopleSoft Data Mover," Using the PeopleSoft Data Mover Command-Line Interface

---

## Starting PeopleSoft Application Designer Command Line Interface

Use the command line syntax and available parameters to start PeopleSoft Application Designer.

**Command Line Syntax**

Use the command line syntax described previously to start PeopleSoft Application Designer.

See [Chapter 19, "Using Command Line Parameters," Understanding PeopleSoft Application Designer Command Line Parameters, page 497.](#)

**Available Parameters**

The available parameters for PSIDE.EXE fall into two general categories:

- Parameters providing login information.
- Parameters setting general options.

This table lists all of the available parameters:

<i><b>Parameter</b></i>	<i><b>Value</b></i>	<i><b>Description</b></i>
-CT	Database type	Enter the type of database to which you are connecting. Valid values are ORACLE, INFORMIX, SYBASE, MICROSOFT, DB2ODBC, and DB2 UDB for Linux, UNIX, and Microsoft (note the spelling of MICROSOFT).
-CS	Server name	Enter the name of the database server for the database that you're connecting to. This setting is required for some database types.
-CD	Database name	Enter the name of the database to which you are connecting, as you would enter it into the PeopleSoft Signon dialog box.

<b><i>Parameter</i></b>	<b><i>Value</i></b>	<b><i>Description</i></b>
-CO	User ID	Enter the PeopleSoft user ID to log in with.
-CP	Password	Enter the password for the specified user ID. If you omit this parameter the PeopleSoft Signon dialog box appears and prompts the user for the password.
-CI	Connect ID	Enter the ID that is used to connect to the database server.
-CW	Password	Enter the password for the specified connect ID.
-SS	NO	Use to suppress the display of the PeopleSoft splash screen. Any other value causes the splash screen to display.
-SN	NO	Use to suppress the sound that plays when you sign in to the PeopleSoft system. Any other value causes the sound to play.
-QUIET	(none)	Use to run in quiet mode, so that no message boxes appear. The system writes messages to the log file or trace file depending on whether you also use the -LF parameter. Quiet mode is recommended for use with processes that require no human interaction.
-HIDE	(none)	Use to hide the PeopleSoft Application Designer interface.
-LF	Log file path name	Path to the log file when used in conjunction with the -QUIET parameter. If used, the system writes any messages generated during processing to the log file.
-FP	Filename	(PeopleSoft Data Mover only) Enter the name of the Data Mover script to run.

**Example**

This command line statement starts PeopleSoft Application Designer and signs in to the DEP7TST database:

```
PSIDE -CT MICROSOFT -CS SEPNTDB05 -CD EP7TST -CO VP1 -CP VP1
```

---

## Building Projects

Use the same command line syntax as identified previously to build a project.

See [Chapter 19, "Using Command Line Parameters," Understanding PeopleSoft Application Designer Command Line Parameters, page 497.](#)

Use the command line statement for the project build to:

- Connect the project to the source database.
- Use the build settings from the Windows registry to generate the SQL script for the create or alter processes.

Before running the build command line, verify that:

- The project's system components are created and loaded.
- The build options are set in the Windows registry.

**Available Parameters**

This table lists all of the available parameters for the project build statement:

<i>Parameter</i>	<i>Description</i>	<i>Error Handling</i>
-HIDE	Use to hide the PeopleSoft Application Designer interface. Always use this parameter when performing a command line Copy process.	Required for all project command line processes.
-CT	Database type. Enter the type of the database to which you are connecting (MICROSOFT, ORACLE, SYBASE, and so on).	Required. If you omit this parameter, the last database type is taken from the registry. If it fails, the process stops and error messages are written to the log file.
-CS	Server name. Enter the name of the source database server for the database to which you are connecting.	Required for some database types. If you omit this parameter, the process stops and error messages are written to the log file.

<b><i>Parameter</i></b>	<b><i>Description</i></b>	<b><i>Error Handling</i></b>
-CD	Database name. Enter the name of the source database to which you are connecting, as you would enter it into the PeopleSoft Signon dialog box.	Required. If you omit this parameter, the process stops and error messages are written to the log file.
-CO	User ID. Enter the PeopleSoft user ID that is needed to sign in to the source database.	Required. If you omit this parameter, the last database type is taken from the registry. If it fails, the process stops and error messages are written to the log file.
-CP	User password. Enter the password for the specified user ID for the source database.	Required. If you omit this parameter, the PeopleSoft Signon dialog box appears and prompts the user for the password. If the password fails, the process stops and error messages are written to the log file (if the log file name parameter is specified).
-PJB	Project name. Enter the name of the project to be built. This project should be available in the database before starting the command line project build.	Required. This is the main parameter and is used internally to decide whether the user can build a project. If you omit this parameter and if you supply all of the source database login parameters, the program only starts the application.

### ***Example***

For this example, assume that the:

- Project name is CJR1.
- Database type is Microsoft.
- Database name is CJR810G.
- User ID is PTDMO.
- Password is PTDMO.
- Path name of the SQL script file is c:\temp\psbuild.sql.
- Path name of the log file is c:\temp\psbuild.log in the Windows registry.

For this example, you would enter the following at the command line:

PSIDE.EXE -CT MICROSOFT -CD CJR810G -CO PTDMO -CP PTDMO -PJB CJR\_PRJ -HIDE -QUIET

## Copying Definitions

Use the same command line syntax as identified previously to copy definitions.

See [Chapter 19, "Using Command Line Parameters," Understanding PeopleSoft Application Designer Command Line Parameters, page 497.](#)

Use the command line statement for Upgrade Copy processes to:

- Connect to the source database.
- Connect to the target database.
- Copy the project and its objects from the source database to the target database.

If the same project already exists in the target database, you can set the option to overwrite the older project with the new project.

### Available Parameters

The following table lists all available parameters for the upgrade copy statement:

<i>Parameter</i>	<i>Description</i>	<i>Error Handling</i>
-HIDE	Use to hide the PeopleSoft Application Designer interface. Always use this parameter when performing a command line Copy process.	Required for all project command line processes.
-CT	Source database type. Enter the type of source database to which you are connecting (for example, ORACLE, SYBASE, and so on).	Required. If the process stops, error messages are written to the log file (if the log file name parameter is specified).
-CS	Source server name. Enter the name of the source database server for the database to which you're connecting.	Required for some database types. If you omit this parameter, the process stops and error messages are written to the log file (if the log file name parameter is specified).

<b><i>Parameter</i></b>	<b><i>Description</i></b>	<b><i>Error Handling</i></b>
-CD	Source database name.  Enter the name of the source database to which you're connecting, as you would enter it into the PeopleSoft Signon dialog box.	Required. If you omit this parameter, the process stop and error messages are written to the log file (if the log file name parameter is specified).
-CO	Source user ID.  Enter the PeopleSoft user ID to use to sign in to the source database.	Required. If you omit this parameter, the user ID is taken from the registry. If it fails, the process stops and error messages are written to the log file (if the log file name parameter is specified).
-CP	Source user password.  Enter the password for the specified user ID for source database.	Required. If you omit this parameter, the PeopleSoft Signon dialog box appears and prompts the user for the password. If the password fails, the process stops and error messages are written to the log file (if the log file name parameter is specified).
-TS	Target server name.  Enter the name of the target database server for the database to which you're connecting.	Required for some database types and for a Database Copy process (but not used for a Copy To File or Copy From File process). If you omit this parameter, the process stops and error messages are written to the log file (if the log file name parameter is specified).
-TD	Target database name.  Enter the name of the target database to which you're connecting, as you would enter it into the PeopleSoft Signon dialog box.	Required for a database Copy process (but not used for a Copy To File or Copy From File process). If you omit this parameter, the process stops and error messages are written to the log file (if the log file name parameter is specified).
-TO	Target user ID.  Enter the PeopleSoft user ID to use to sign in to the target database.	Required for a Database Copy process (but not used for a Copy To File or Copy From File process). If you omit this parameter, the process stops and error messages are written to the log file (if the log file name parameter is specified).

<b><i>Parameter</i></b>	<b><i>Description</i></b>	<b><i>Error Handling</i></b>
-TP	Target user password.  Enter the password for the specified user ID for the target database.	Required for a Database Copy process (but not used for a Copy To File or Copy From File process). If you omit this parameter, the process stops and error messages are written to the log file (if the log file name parameter is specified).
-LF	Log file name.  Enter the name of the file in which error messages are logged during the command line Upgrade Copy process.	Not required. If you omit this parameter, a file is created with the specified path and name, and all of the processing and error messages are written to that file.
-PJC	Source project name.  Enter the name of the project that is to be copied from the source database to the target database. This project should be available in the source database before starting the command line Upgrade Copy process to target database.	Required for a Database Copy process. This is the main parameter that is used internally by the executable file to identify an Upgrade Copy process. If you omit this parameter and if you supply all of the source database login parameters, the executable file starts the application.
-PJTF	Copy Project To File project name.  Enter the name of the project to be copied from the source database to the file.	Required for a Copy To File process. Do not include the file path to the project with this parameter. Use -FP to specify the file path.
-PJFF	Copy Project From File project name.  Enter the name of the project to be copied from the file into the source database.	Required for a Copy From File process. Do not include the file path to the project with this parameter. Use -FP to specify the file path.
-FP	Project file path.  Enter the path to the project for a Copy to File or Copy From File process.	Required for a Copy To File or Copy From File process (but not for a Database Copy process). For the Copy From File process, the directory must exist. For the Copy To File process, any project with the same name is overwritten unless you use the -OVW parameter. A directory specification is created if it does not exist. Do not include the name of the project in the file path. Use -PJTF or -PJFF to specify the project name.

<b>Parameter</b>	<b>Description</b>	<b>Error Handling</b>
-OVD	<p>Override dependencies (0 or 1).</p> <p>Enter 1 to override the checks for unapplied dependency incidents during the Copy From File process and to perform the copy even if the project's dependency incidents have not been applied to the source database.</p>	<p>Not required, and not applicable for processes other than the Copy From File process. The default is 0. When you enter the default and if the project's dependency incidents have not been applied to the source database, an error message is written to the log file listing which incidents need to be applied. The Copy From File process also stops. When you enter 1, a warning message is written to the log for unapplied incidents and processing continues.</p>
-CL	<p>Commit limit.</p> <p>Enter the Commit limit (a number greater than 0) for the number of objects to be copied or compared before a commit is issued. Example: -CL 150</p>	<p>Not required. The default is 50 if the user does not set this parameter.</p>
-AF	<p>Audit flags on records (0 or 1).</p> <p>Enter 1 to keep the target audit flags as is. Enter 0 to take them from the source database. Example: -AF 0</p>	<p>Not required. The default is 1.</p>
-DDL	<p>Data definition language (DDL) on records and indexes (0 or 1).</p> <p>Enter 1 to keep the target database DDL flags as is. Enter 0 to take them from the source database. Example: -DDL 0</p>	<p>Not required. The default is 1.</p>
-PPL	<p>Portal registry structures permission list references.</p> <p>Enter 1 to keep the target portal registry permission list references as is. Enter 0 to take them from the source database.</p>	<p>Not required. The default is 1.</p>
-EXP	<p>Export project definition (0 or 1).</p> <p>Enter 1 to export the project definition to the target database when using the Database Copy process. Not applicable for Copy Project to File process.</p>	<p>Not required. The default is 1. Any project with the same name is overwritten with the new project definition.</p>

<b><i>Parameter</i></b>	<b><i>Description</i></b>	<b><i>Error Handling</i></b>
-OBJ	<p>Object types to copy</p> <p>List the object types as numbers with commas as delimiters. For example, if you choose records and indexes alone for copying, use the appropriate numbers for records and indexes from the Definition Type Selection table. For records, the number is 0, and for the indexes, the number is 1. Example: -OBJ 0,1</p> <p>See <a href="#">Appendix B, "Referencing Selections for Command Line Parameters," Definition Type Selections, page 546.</a></p>	Not required. If you omit this parameter, then all of the objects are copied by default.
-RST	<p>Reset done flags (0 or 1).</p> <p>Enter 1 to reset the done flags before initiating the Copy process. Enter 0 to not reset them. Example: -RST 0</p>	Not required. The default is 1.
-OVW	<p>Overwrite existing project (0 or 1).</p> <p>Enter 1 to overwrite any existing projects with the same name in the specified path during the Copy Project To File process, or in the database in the Copy Project From File process. Enter 0 to not overwrite existing projects. This applies to the Copy Project To and From File processes only.</p>	Not required. The default is 1.

<b>Parameter</b>	<b>Description</b>	<b>Error Handling</b>
-LNG	<p>Copy languages.</p> <p>List the language codes with commas as delimiters, or enter ALL to select all available languages. For new languages that are not listed in the Language Selection table, use the xlat short name from the LANGUAGE_CD field in the translate table. For example, if you choose English and Spanish for copying, enter the appropriate codes for English and Spanish from the Language Selection table. For English, the code is ENG, and for Spanish, the code is ESP. Example: -LNG ENG,ESP</p> <p>See <a href="#">Appendix B, "Referencing Selections for Command Line Parameters," Language Selections, page 550.</a></p>	<p>Not required. If you omit this parameter, then the languages that are already set in the project are used as the default. The COMMON language code refers to nonlanguage attributes that are not translated. Translated attributes can be copied separately from nontranslated attributes. If you do not specify the COMMON code when using -LNG, you copy only the translated language attributes.</p> <p>Changing languages on the command line changes the values that are stored with the project definition. Subsequent Copy and Compare processes on the modified project use the new values unless they are explicitly changed again in the upgrade options.</p>
-CFD	<p>Keep Target ChartField PageField Display Size.</p> <p>(Number 1 or 0). This parameter indicates whether or not the target chartfield page field display size attributes are to be kept, or whether they should be set from source during copy. See Upgrade Options for more information.</p> <p>Enter 1 to keep the target chartfield page field display size attributes.</p> <p>Enter 0 to set these attributes from the source values.</p>	<p>Not required. Default is 1</p>
-CFF	<p>Keep Target ChartField Field Format.</p> <p>(Number 1 or 0) This parameter indicates whether or not the target chartfield field format attributes are to be kept, or whether they should be set from the source during Copy. See Upgrade Options for more information.</p> <p>Enter 1 to keep the target chartfield field format attributes.</p> <p>Enter 0 to set these attributes from the source values.</p>	<p>Not required. Default is 1</p>

**Example: Database Copy Process**

These are some command line statements for the Database Copy process:

- Copy the project PPLTOOLS from the source Microsoft SQL Server database, PTDMO, to the target database, PTTST.
- Log process and error messages to c:\temp\copy.log.
- Set the commit limit to 150.

Copy the audit flags and record and index DDLs from the source database.

- Copy only records, indexes, pages, queries, and process definitions from the project.
- Do not reset the done flags before initiating the Copy process.
- Copy only English and Spanish translations.

```
PSIDE.EXE -HIDE -PJC PPLTOOLS -CT MICROST -CD PTDMO -CO PTDMO -CP PTDMO -TD
PTTST -TO PTDMO -TP PTDMO -QUIET -LF C:\TEMP\COPY.LOG -CL 150 -AF 0 -DDL 0 -OBJ
0,1,5,10,20 -RST 0 -LNG ENG, ESP
```

**Example: Copy To File Process**

These are some command line statements for the Copy to File process:

- Copy the PPLTOOLS project from the Oracle database, PTDMO, by using the PTDMO user ID, to the local directory c:\temp\export.

A directory named PPLTOOLS is created in c:\temp\export.

- Log progress and error messages to c:\temp\copy.log.

```
PSIDE.EXE -HIDE -PJTF PPLTOOLS -FP c:\temp\export -CT ORACLE -CD PTDMO -CO PTDMO -CP
PTDMO -QUIET -LF c:\temp\copy.log
```

**Example: Copy From File Process**

These are some command line statements for the Copy From File process.

- Copy the PPLTOOLS project to the DB2 UDB for Linux, Unix, and Windows database, PTDMO, by using the user ID PTDMO, from the local directory c:\temp\export.

A directory named PPLTOOLS must exist under c:\temp\export.

- Log progress and error messages to c:\temp\copy.log.

```
PSIDE.EXE -HIDE -PJFF PPLTOOLS -FP c:\temp\export -CT DB2UNIX -CD PTDMO -CO PTDMO -CP
PTDMO -QUIET LF c:\temp\copy.log
```

## Comparing Definitions for Upgrade Using the Command Line

Use the same command line syntax as identified previously to compare definitions.

See [Chapter 19, "Using Command Line Parameters," Understanding PeopleSoft Application Designer Command Line Parameters, page 497.](#)

Use the command line statement for an Upgrade Compare process to:

- Connect to the source database.
- Connect to the target database.
- Perform a project or database compare (based on parameters).
- Create compare reports in the specified output directory.

### Available Parameters

This table lists all of the available parameters for the upgrade compare statement:

<i>Parameter</i>	<i>Description</i>	<i>Error Handling</i>
-HIDE	Use to hide the PeopleSoft Application Designer interface. Always use this parameter when performing a command line Compare process.	Required for all project command line processes.
-CT	Source database type. Enter the type of source database to which you're connecting (for example, ORACLE, SYBASE, and so on). Choose from the Database Type Selection table.  See <a href="#">Appendix B, "Referencing Selections for Command Line Parameters," Database Type Selections, page 545.</a>	Required. If the process stops, error messages are written to the log file (if the log file name parameter is specified).
-CS	Source server name. Enter the name of the source database server for the database to which you're connecting.	Required for some database types. If you omit this parameter, the process stops and error messages are written to the log file (if the log file name parameter is specified).

<b><i>Parameter</i></b>	<b><i>Description</i></b>	<b><i>Error Handling</i></b>
-CD	Source database name.  Enter the name of the source database to which you are connecting, as you would enter it into the PeopleSoft Signon dialog box.	Required. If you omit this parameter, the process stops and error messages are written to the log file (if the log file name parameter is specified).
-CO	Source user ID.  Enter the PeopleSoft user ID to use to sign in to source database.	Required. If you omit this parameter, the user ID is taken from the registry. If it fails, the process stops and error messages are written to the log file (if the log file name parameter is specified).
-CP	Source user password.  Enter the password for the specified user ID for source database.	Required. If you omit this parameter, the PeopleSoft Signon dialog box appears and prompts the user for the password. If the password fails, the process stops and error messages are written to the log file (if the log file name parameter is specified).
-TS	Target server name.  Enter the name of the target database server for the database to which you're connecting.	Required for some database types. If you omit this parameter, the process stops and error messages are written to the log file (if the log file name parameter is specified).
-TD	Target database name.  Enter the name of the target database to which you are connecting, as you would enter it into the PeopleSoft Signon dialog box.	Required. If you omit this parameter, the process stops and error messages are written to the log file (if the log file name parameter is specified).
-TO	Target user ID.  Enter the PeopleSoft user ID to use to sign in to the target database.	Required. If you omit this parameter, the process stops and error messages are written to the log file (if the log file name parameter is specified).
-TP	Target user password.  Enter the password for the specified user ID for the target database.	Required. If you omit this parameter, the process stops and error messages are written to the log file (if the log file name parameter is specified).

<b><i>Parameter</i></b>	<b><i>Description</i></b>	<b><i>Error Handling</i></b>
-LF	Log file name. Enter the name of the file in which error messages are logged during the command line Upgrade Compare process.	Not required. If you omit this parameter, a file is created with the specified path and name, and all of the error messages are written to that file.
-PJM	Source project name. Enter the name of the project that is to be compared from the source database to the target database. This project should be available in the source database before starting the command line Upgrade Compare process to the target database.	Required. This is the main parameter that is used internally by the executable file to identify an upgrade compare. If you omit this parameter and if you supply all of the source database login parameters, this executable file starts the application.
-CL	Commit limit Enter the Commit limit (a number greater than 0) for the number of objects to be copied or compared before a commit is issued. Example: -CL 150	Not required. The default is 50.
-AF	Audit flags on records (0 or 1). Enter 1 to keep the target audit flags as is. Enter 0 to take them from the source database. Example: -AF 0	Not required. The default is 1.
-PPL	Portal registry structures permission list references. Enter 1 to keep the target portal registry permission list references as is. Enter 0 to take them from the source database. Example: -PPL 0	Not required. The default is 1.
-DDL	DDL on records and indexes ( 0 or 1). Enter 1 to keep the target database DDL flags as is. Enter 0 to take them from the source database. When keeping the target database DDL, no differences are shown on the compare report. Example: -DDL 0	Not required. The default is 1.

<b>Parameter</b>	<b>Description</b>	<b>Error Handling</b>
-CFD	Chartfield PageField Display Size (0 or 1). Enter 1 to keep the target database chartfield pagefield display size attributes. Enter 0 to take them from the source database. When keeping target database attributes, no differences are shown on the compare report for these attributes. Example -CFD 0	Not required. The default is 1.
-CFF	Chartfield Field Format (0 or 1). Enter 1 to keep the target database chartfield field format attributes. Enter 0 to take them from the source database. When keeping target attributes, no differences are shown on the compare report for these attributes. Example: -CFF 0	Not required. The default is 1.
-CMT	Compare type: project (1) or database (0). Example: -CMT 0	Not required. The default is 1.
-TGT	Target Orientation: PeopleSoft vanilla (0) or keep personalization (1).	Not required. The default is 0.
-CBY	Comparison By: To compare by release, enter REL, followed by the release number. To compare by date, enter DAT, followed by a date in the following format: YYYY-MM-DD-HH.MM.SS.sss. Release numbers are included in the Available Release Numbers list. Example: 2002-02-01-12.00.00.000	Not required. The default is REL.
-ROD	Report Output Directory. Enter the report output directory location. Example-ROD c:\temp\upgreports	Not required. If you omit this parameter, the system uses the Report Output Directory value from the Tools, Options General tab.

<b>Parameter</b>	<b>Description</b>	<b>Error Handling</b>
-OBJ	<p>Definition type to compare</p> <p>List the definition types to compare as numbers with commas as delimiters, or enter All. Example: -OBJ 0,1</p> <p>See <a href="#">Appendix B, "Referencing Selections for Command Line Parameters," Definition Type Selections, page 546.</a></p>	<p>Not required. If you omit this parameter, then all of the definitions are compared by default. For a project comparison, this refers to all definitions in the project. For a database comparison, this includes all definitions in the database. For example, If you choose records and indexes alone for copying, use the appropriate numbers for records and indexes from the Definition Type Selection table. For records, the number is 0, and for the indexes, the number is 1.</p>
-LNG	<p>Compare languages.</p> <p>List the language codes with commas as delimiters, or enter ALL to select all available languages. For new languages that are not listed in the Language Selection table, use the xlat short name from the LANGUAGE_CD field in the translate table. For example, if you choose English and Spanish for comparing, enter the appropriate codes for English and Spanish from the Language Selection table. For English, the code is ENG, and for Spanish, the code is ESP. Example: -LNG ENG,ESP</p> <p>See <a href="#">Appendix B, "Referencing Selections for Command Line Parameters," Language Selections, page 550.</a></p>	<p>Not required. If you omit this parameter, then the languages that are already set in the project are used as the default. The COMMON language code refers to nonlanguage attributes that are not translated. Translated attributes can be compared separately from nontranslated attributes. If you do not specify the COMMON code when using -LNG, you compare only the translated language attributes.</p> <p>Changing languages on the command line changes the values that are stored with the project definition. Subsequent Copy and Compare processes on the modified project use the new values unless they are explicitly changed again in the upgrade options.</p>

<b>Parameter</b>	<b>Description</b>	<b>Error Handling</b>
-CMXML	<p>Generate browser reports (0 or 1).</p> <p>This command enables you to enable the option of generating browser reports written to your report output directory. This is equivalent to setting the Generate Browser Reports options on the Report Options tab on the Upgrade Options dialog box.</p> <p>Enter 1 to enable this option, and enter 0 to disable this option.</p> <p>Example: -CMXML 1</p> <p>See <i>Enterprise PeopleTools 8.50 PeopleBook: PeopleSoft Application Designer Lifecycle Management Guide</i>, "Comparing Source and Target Definitions," Working with Browser Reports.</p>	Not required. The default is 0.
-CMR [0   1]	Use 0 to disable, 1 to enable, the generation of the Composite Compare Reports. Parameter is only valid when used in conjunction with the -PJM option. Ignored otherwise.	Not required. If not used, then composite report disabled. Must also use -CROD and -CRDB options when using -CMR.
-CROD [directory path]	Set path for the Old Release Compare reports when generating Composite Reports (-CMR 1). Only valid when used in conjunction with the -PJM and -CMR 1 options, ignored otherwise.	If not set when generating composite reports (-CMR 1) then its an error condition, and processing should stop, and an error generated to the log file. Directory path can be a local directory, a mapped network directory, or a UNC path specification.
-CRDB [database name]	Pass the value of the Old Release Source Database name when generating Composite Reports. This value must match the Source Database name value from the Old Release Report for each definition type. This is to ensure the original old release report is valid and contains the expected data. If the old release report source database name for the current definition type does not match the passed value, then an error message is generated to the log file, but process continues for other definition types.	Not required.

***Example: Running an Upgrade Compare Process***

Use this example as a guide for running an Upgrade Compare process:

- Run a database Compare process against the source Microsoft SQL Server database, PTDMO, and the target database, PTTST.
- Log process and error messages to c:\temp\compare.log.
- Set the commit limit to 150.
- Show the audit flag and DDL differences between databases.
- Compare records, indexes, pages, queries, and process definitions.
- Compare only English and Spanish translations.
- Set the database compare type.
- Set the target orientation to keep the customizations.
- Compare by release 8.40.00.000.
- Generate compare reports to the c:\temp\upgreports directory.

```
PSIDE.EXE -HIDE -CT MICROSOFT -CD PTDMO -CO PTDMO -CP PTDMO -PJM PPLTOOLS -TD  
PTTST -TO PTDMO -TP PTDMO -QUIET -LF C:\TEMP\COMPARE.LOG -CL 150 -AF 0 -DDL 0 -OBJ  
0,1,5,10,20 -LNG ENG, ESP -CMT 0 -TGT 1 -CBY REL Core 8.40.00.000 -ROD C:\TEMP\UPGREPORTS
```



## Chapter 20

# Using Change Control

This section provides overviews of Change Control features and implementation considerations and discusses how to:

- Set up Change Control.
- Use projects.
- Use Change Control.

---

## Understanding Change Control Features

The Change Control feature has three main functions to help you manage and track development. You can enable one or all of these functions to control how much access each user has to the Change Control commands.

This section discusses:

- Change Control locking
- Change Control history
- Change Control stamping

## Change Control Locking

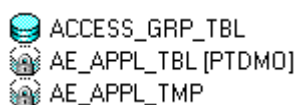
Change Control locking is keyed by PeopleSoft user IDs. When a definition is locked, it cannot be modified by anyone other than the user ID who locked it, and it can be unlocked only by that same user or by a Change Control administrator.

A locked definition is identified by:

- A small padlock icon above the definition in the project workspace (development view) of PeopleSoft Application Designer.
- The user ID of the person who locked the definition, which appears next to the name of the definition.

Your user ID is not displayed for definitions that *you* have locked.

For example, the following icons seen in the project workspace display two locked and one unlocked record definitions.



Locked and unlocked record icons

The top record is unlocked; the middle record is locked by another user ID (PTDMO), and the bottom record (with no ID displayed) is locked by you, the current user.

---

**Note.** Standard Change Control locking is supported only for definitions that you can modify with PeopleSoft Application Designer. Other PeopleSoft definitions can be added to a project for upgrading, but they cannot be locked by developers. If you are a Change Control administrator, you can lock all upgradable database definitions—both PeopleSoft Application Designer development definitions and other types. However, for non-PeopleSoft Application Designer definitions, this action prevents unauthorized upgrading only—not unauthorized development.

---

## Locking Projects

With Change Control, projects are treated like all other PeopleSoft Application Designer definitions—you must lock them before you can modify them. However, locking a project does *not* lock the definitions in the project, and modifying a definition in a project does not modify the project itself.

A project definition includes a name and a list of definitions. When you lock a project definition, other users cannot add or remove definitions from the project, and they cannot rename or delete the project. However, locking does *not* restrict access to the definitions that are named in the project definition. Likewise, modifying a definition has no effect on the definition of any project to which the definition might belong.

---

**Note.** PeopleSoft Application Designer provides an option to load the last open project on startup. If this option is enabled on your machine and Change Control locking is activated, you might receive an *open in read-only mode?* message at startup if you hadn't locked the project before, or if someone else has the project locked. In either case, you can open the project in read-only mode. Remember, this does not restrict your access to definitions in the project.

---

## Locking Compared to Version Control

Change Control locking is not the same as version control. With a version control system, you check out a copy of a definition and make your changes to the copy. After you check in the changed version, you can always undo your changes. This is not the case with change control locking.

Locking a definition prevents other users from modifying it. However, any changes that you save are written directly to the database, overlaying or replacing the existing definition. There is no way to restore a previous version of a definition.

## Locking and Upgrades

When preparing to upgrade a database, it's crucial that all development ceases in the source database. This assures that the Compare process is dealing with a static environment. It also assures that changes aren't made to any definitions between the time that you set the upgrade defaults and the time that you copy the definitions.

You can freeze all development by using the Change Control Administrator dialog box to lock *all* database definitions. When the upgrade is done, use the same dialog box to unlock all definitions. However, be aware that this action *permanently* removes all previous lock settings from all definitions. Developers have no way of resetting their locks except by manually relocking. When you lock definitions in this way, it is not reflected in the Locked Objects dialog box or in the project workspace. If a developer has unsaved changes when you lock all definitions, the developer cannot save those changes.

For all of these reasons, it's imperative that you inform developers that you plan to lock all definitions and that you give them time to save their changes, perhaps even to view the Locked Definitions dialog box and to print the screen.

Locking definitions with the Change Control Administrator dialog box doesn't actually mark every definition as locked. Instead, it adds a single row at the top of the locking table. The presence of this row indicates to the system that full database locking is in effect, and it stores the user ID of the administrator who enabled the locking.

Full database locking also plays a role in the target database during the upgrade copy process. During a copy, the system always checks the locking status of the target database definitions to see if they're locked and by whom. If they've been locked by a user ID other than the one performing the copy, those definitions aren't modified.

In major upgrades, checking the locking status of each definition before copying severely impacts performance. To prevent this, use the Change Control Administrator dialog box to lock all target database definitions before copying. During the copy process, if the entire target database is locked, the system verifies that the user ID performing the copy process is the same user ID that locked the database. If these conditions are true, the system assumes that it was locked for the purpose of the upgrade and that it can safely copy all definitions without checking each one individually.

---

**Note.** When you copy a project, the system doesn't check the locking status of the definitions in the source database. However, you should keep your definitions locked until the copy is complete.

---

## Change Control History

When Change Control history is enabled, you can enter comments about the modifications that you made to PeopleSoft Application Designer development definitions. History entries contain a common set of information, including who created the entry, when, and the type of action that is associated with the entry. For example, when a user locks a definition, a history entry is automatically created containing the user ID, the data and time, and an action value of Add. This entry can also contain a project name, incident ID, and comment.

### ***Automated History Prompting***

Although you can always insert history entries manually, there are many situations in which PeopleSoft Application Designer automatically inserts history entries and prompts you for comments. In addition, there are special circumstances when entries are added without a prompt. For example, an action history provides some idea of what has happened to a definition, even if no comments were entered. Possible action entries are:

<b>Action Entry</b>	<b>Description</b>
Lock and Unlock	Whenever a definition is locked or unlocked, you are automatically prompted for a comment, after which a Lock or Unlock action entry is added to the definition history.

<b>Action Entry</b>	<b>Description</b>
Rename	<p>When you rename a definition, you're prompted for a comment, after which a Rename action entry is added to the definition history. PeopleSoft provides a default comment of &lt;name1&gt; renamed to &lt;name2&gt;:</p> <ul style="list-style-type: none"> <li>• &lt;name1&gt; is the previous name of the definition.</li> <li>• &lt;name2&gt; is the new name that you gave it.</li> </ul> <p><b>Note.</b> If the definition is locked when you try to rename it, the system unlocks it before renaming and relocks it afterward. In this case, three history entries are added: one for unlocking, one for renaming, and one for relocking.</p>
Delete	<p>When you delete a PeopleSoft Application Designer definition, its history is retained. During the deletion, PeopleSoft Application Designer prompts you to add a final comment into the definition history, after which a Delete action entry is added to the definition history.</p> <p><b>Note.</b> If Change Control locking is enabled, you can only delete locked definitions. After a locked definition is deleted, it is automatically unlocked and an Unlock action history entry is added. You are not prompted for comments for this unlock event.</p>
Add	<p>When you create a new definition, PeopleSoft Application Designer creates a history entry with an Add action. You are not prompted for comments.</p>
Copy	<p>If Change Control history is enabled in the target database when you perform an upgrade copy, a Copy action entry is inserted into the histories for any added, replaced, or deleted definitions. You are not prompted for comments.</p>

### ***History and Upgrades***

Change Control history is not copied along with its associated definition during an upgrade. However, if history is enabled in the source database, then the history of each affected target definition is updated with a comment noting when the copy was performed and by whom.

This behind-the-scenes history updating occurs for all target definitions that are affected by the copy process—even non-PeopleSoft Application Designer definitions with histories that can't be updated or viewed—regardless of whether Change Control history is enabled in the source database.

## **Change Control Stamping**

Change Control stamping is always in effect, regardless of whether locking and history are also enabled. For every definition in the database, PeopleTools maintains a last updated stamp, which denotes the date and time of the last update and the user ID of the person who saved the definition. When PeopleSoft delivers a new database, all of the definitions are stamped with a PeopleSoft proprietary ID: PPLSOFT.

## Stamping and Upgrades

Change Control stamping provides critical information during an upgrade comparison. Because the system tracks the user ID of whoever last changed each definition, you can easily identify your adaptations. (Any definition that is stamped with a user ID that isn't the PeopleSoft proprietary ID is considered an adaptation.) Whether you made an adaptation before or after the last update is irrelevant. The adaptation is always identified as such.

During a comparison, definitions that you last modified are given a status of Custom Changed (if they've changed since the compare date), or Custom Unchanged (if they haven't changed since the compare date). Definitions that PeopleSoft last modified are given a status of either Changed or Unchanged.

---

## Understanding Implementation Considerations

When deciding how to implement Change Control, consider:

- The level of development control that you want to maintain.
- The amount of freedom that developers need.
- The security requirements of the site.

<i>How to Implement</i>	<i>Advantage</i>	<i>Disadvantage</i>
Individual Control	For maximum control of the development environment, use both locking and history, and assign each developer a unique user ID. Then, definitions can be modified by only one developer (user ID) at a time, and developers are always prompted for comments when they lock and unlock definitions.	Developers can share ownership of their definitions only by unlocking them after each change.
Group Control	For flexibility, use locking and history, but assign developers who work on the same project a common ID. Then, developers can share definitions with the other members of their group, but not with members of other groups. Administering security is also easier, as there are fewer user IDs to maintain.	There is decreased protection from simultaneous development on definitions. History is harder to track, unless developers always include their names in their comments.
History Only	Provides the least restricted Change Control environment. In this situation, all definitions can be shared among all developers.	Developers aren't automatically prompted for comments. They can all share the same ID. If you have a very small development team, this might be a good option.

## Setting Up Change Control

This section provides an overview of Change Control security and discusses how to:

- Appoint a Change Control administrator.
- Enable or disable Change Control.

## Understanding Change Control Security

Using the Security component, you can assign users one of three Change Control access levels, depending on how much authority you want them to have:

<b><i>Change Control Access Level</i></b>	<b><i>Description</i></b>
Restricted access	Restricts users from locking or unlocking definitions. When Change Control locking is enabled, users with restricted access can open PeopleSoft Application Designer development definitions in read-only mode only. Users are also unable to view or update definition histories.
Developer access	Allows users to lock any unlocked definitions and to unlock any definitions that they have locked. They can then manipulate definitions, as they are allowed in their security profile. Users can also view and enter definition history comments.
Supervisor access	Allows users to unlock any locked definitions, regardless of who locked them. They can also access the Change Control Administrator dialog box, in which you lock and unlock all definitions at one time and enable and disable Change Control locking and history.

If Change Control locking is disabled, these access levels have no security value. If history is disabled, users with developer and supervisor access can still view the History dialog box—users with restricted access cannot.

Remember that Change Control is based on user IDs. If developers all share the same user ID, Change Control offers no advantage in control because each developer can modify definitions that are locked by others.

## Appointing a Change Control Administrator

Appoint Change Control administrators by giving certain users supervisor-level access to Change Control. When users have this access level, they can enable and disable Change Control.

See *Enterprise PeopleTools 8.50 PeopleBook: Security Administration*, "Setting Up Permission Lists," Setting PeopleTools Permissions.

## Enabling or Disabling Change Control

To enable or disable Change Control:

1. In PeopleSoft Application Designer, select Tools, Change Control, Administrator.

The Change Control Administrator dialog box appears.

2. Specify the System Wide Options and access control:

**Use change control locking**

Select to require that PeopleSoft Application Designer definitions be locked before they can be modified.

**Use change control history**

Select to enable PeopleSoft Application Designer developers to insert comments about open definitions.

If both Use change control locking and Use change control history check boxes are selected, developers are prompted for comments when locking and unlocking definitions.

---

**Note.** Because these are system-wide settings, if you change them, all users must log off and on again for the changes to take effect.

---

**Lock all definitions**

Select or clear to lock or unlock all definitions in the database. Usually, you lock all definitions only before a major upgrade, because it permanently removes all individual developer locks.

---

## Using Projects

This section provides an overview of projects and discusses how to:

- Use multiple databases for development.
- Use distributed development environments.

## Understanding Projects

You can use all levels of Change Control with or without also using projects. If you decide not to use projects, you rely on the Locked Objects dialog box, rather than the project workspace, to identify locked definitions. The dialog box provides a better overall view of locking status because it shows all of the PeopleSoft Application Designer definitions in the database, not just those in the current project.

Use projects to track the definitions that are changed as part of a change or feature request. This set of definitions is commonly referred to as the change set. The Options dialog box has an option to insert a definition into a project when it is modified and saved. If you start with an empty project, this option provides an easy way of tracking the change set for this incident. When the change request is completed, the project contains everything that is associated with the change. It's also a good idea to use the Comments field in the Project Properties dialog box to list any external definitions like COBOL or Structured Query Report (SQR) modules that must be migrated with this change.

## Using Multiple Databases for Development

Managing change in a single database environment is straightforward, but very few PeopleSoft users operate in a single database environment. The classic development model uses three databases: development, test, and production. You apply all changes to the development database. After you unit-test the change, migrate the change set to the test database, where it goes through more rigorous testing. Usually, you would run one or more regression test suites to ensure that you resolve the issue that you intended to resolve with no unwelcome side effects. Finally, that change set is migrated into the production database. If you find a problem at any stage in the process, the incident is sent back to development and the process begins again.

This model assumes that the development database is the master database. You can use the Change Control locking feature to lock down the modules on which you and other developers are working. When the changes are completed in the development database, the Change Control administrator is notified and uses the upgrade copy facility to copy the change set into the test environment. As long as you use the technique that is described previously in this section, the project should contain the entire change set. The system tracks all of the documentation for the change in the development database. The only information that appears in the test and production databases is a history line indicating it was copied. Definitions move only in one direction in this model: from development to test, then from test to production.

---

**Note.** The only case in which you can copy a definition back to development from either test or production is if a problem must be re-created and another change has already been made to the affected definition. Do this with extreme care because upgrade copies are destructive and cannot be undone if you discover that you overlaid another developer's change. For this reason, you should only apply changes directly to test or production databases very rarely.

---

---

**Note.** This is just one Change Control model that you can use. This is provided to give you an idea of how you can implement Change Control in your environment. While you do not need to follow this model exactly, it is important that you implement a Change Control model that enables you to track changes to the system and prevent developers from overwriting each other's changes.

---

## Using Distributed Development Environments

It is good idea to use a master development database, even if each development team works on its own copy of the database. Developers should lock down the definitions on which they intend to work in the master development database, then copy those modules to their private databases. This ensures that no other developer makes a change to those definitions while they are checked out.

When the developer is ready to copy changes back to the master development database, check the Change Control history of the locked definitions in the master development database. Do this before using upgrade copy to migrate them back, just in case a Change Control administrator has overridden a lock and made a change while the definitions were checked out.

---

**Note.** Change Control administrators should always notify the developer who has a lock on a definition before they override to avoid unexpected surprises later.

---

---

## Using Change Control

This section discusses how to:

- Lock and unlock definitions.
- View locked definitions.
- Insert comments.
- Delete page definitions.
- View Change Control history.
- Report Change Control information.

### Locking and Unlocking Definitions

You can lock and unlock definitions manually. You can also have PeopleSoft Application Designer lock unlocked definitions for you each time that you open them.

You must have developer or supervisor access to Change Control to lock and unlock definitions. If you have supervisor access, you can also lock all definitions at once. This can be helpful when performing upgrades to ensure that definitions aren't modified in the middle of the process.

This section discusses how to:

- Lock or unlock an unopened definition in the current project.
- Unlock an unopened definition that is not in the current project.
- Lock or unlock an open definition.
- Lock definitions automatically when you open them.
- Lock or unlock all definitions at once.

#### ***Locking or Unlocking an Unopened Definition in the Current Project***

To lock or unlock an unopened definition in the current project:

1. In the project workspace, select the Development tab at the base of the workspace window to activate the development view.
2. Select the definitions that you want to lock or unlock.
3. Right-click any of the selected definitions and select Lock Definition or Unlock Definition.

#### ***Unlocking an Unopened Definition Not in the Current Project***

To unlock an unopened definition that is not in the current project:

1. Select Tools, Change Control, View Locked Definitions.

The Locked Definitions dialog box appears.

---

**Note.** The View Locked Definitions menu item is not available if you have restricted access to Change Control.

---

2. Select the definition type and user.

You can view all locked definitions of the specified type by selecting (*all*) from the User drop-down list box.

3. Select the definitions to unlock.
4. Right-click any selected definition and select Unlock Definition.

If Change Control history is enabled, you're prompted for comments.

### ***Locking or Unlocking an Open Definition***

To lock or unlock an open definition:

1. Activate the definition.
2. Select Tools, Change Control, and then select either Lock Definition or Unlock Definition.

### ***Locking Definitions Automatically When You Open Them***

To lock definitions automatically when you open them:

1. Select Tools, Options, Change Control.
2. Select Lock definition when it is opened.

Now, whenever you open a definition it is locked automatically, unless you have only restricted access to Change Control. In this case, you're notified that you have restricted access and asked whether you want to open the definition in read-only mode.

---

**Note.** Like all settings in the Application Designer Options dialog box, this setting controls the behavior on your workstation only. Also, definitions cannot be unlocked automatically. You must always unlock them manually.

---

### ***Locking or Unlocking All Definitions at Once***

To lock or unlock all definitions at once:

1. Select Tools, Change Control, Administrator.

The Change Control Administrator dialog box appears.

2. Select Lock all definitions to lock all definitions, or clear it to unlock all definitions.

Locking all definitions applies a database-wide lock that is tagged with your user ID.

---

**Warning!** Selecting this check box removes individual locks from all database definitions. You should proceed with this step only if you've informed all of your developers and given them an opportunity to save any unsaved work.

---

3. Click OK.

If you're locking all definitions, you're warned that this action permanently cancels existing locks. Click Yes.

## Viewing Locked Definitions

At times you may want to see a definition even if it is locked.

To view locked definitions:

1. Select Tools, Change Control, View Locked Definitions.

The Locked Definitions dialog box appears.

2. Select the user whose locked definitions you want to view.
3. Select the type to display.

You can view only one definition type at a time. You can also unlock definitions in this dialog box.

## Inserting Comments

When Change Control history is enabled, you can insert comments about an open definition at any time. To help ensure that you insert new comments with each modification, you can instruct PeopleSoft Application Designer to prompt you for a comment every time you save a definition, every time you lock or unlock a definition, or both.

This section discusses how to:

- Insert a comment for an unopened definition in the current project.
- Insert a comment for an open definition.
- Enable a prompt for comments when saving a definition.

### ***Inserting a Comment for an Unopened Definition in the Current Project***

To insert a comment for an unopened definition in the current project:

1. In the project workspace, select the Development tab at the base of the workspace window to activate the development view.

2. Select the definition for which you want to insert comments.

You cannot insert comments for more than one definition at a time.

3. Right-click the selected definition and select Insert Comment.

The Insert Comment dialog box appears.

4. Enter the necessary information:

<b>Project</b>	Displays the name of the current project by default, but you can delete or replace this value.
<b>Incident ID</b>	Denote the incident to which your development corresponds.
<b>Comments</b>	Include information about why and how you're modifying the definition.

If you click OK, the information is inserted into the definition history and the dialog box closes. If you click Apply, your comments are inserted, but the dialog box remains open. You can then enter comments for another history entry. When you click OK or Apply, these new comments are inserted as a new history entry; they do not replace the previous entry.

### ***Inserting a Comment for an Open Definition***

To insert a comment for an open definition:

1. Use the list in the Window menu to navigate to the definition.
2. Select Tools, Change Control, Insert Comment.
3. Enter the name of the project and incident ID.
4. Enter your comments.
5. Click OK or Apply.

### ***Enabling a Prompt for Comments When Saving a Definition***

To be prompted for comments when saving a definition:

1. Select Tools, Options, Change Control.

The Change Control page appears.

2. Select Prompt for comments when definition is saved.

Whenever you save a definition, the system prompts you to insert history comments.

---

**Note.** This setting affects the behavior on your workstation only. One possible drawback to using this setting is that a definition might be saved many times as part of a single change and you're prompted for comments at every save.

---

## Deleting Page Definitions

When deleting page definitions with Change Control enabled, the system presents this message:

Are you sure you want to delete <*definition name*> definition? You are presented with Yes and No options. If you click the Yes button, and the page is locked then the page will be deleted. If you click the Yes button and the page is not locked then this message box appears:

You must lock <*definition name*> before you may rename or delete it.

You must lock the page before you are allowed to delete it.

If you click the No button, then the system does not delete the definition.

---

**Note.** If the definition is a base language page, then the system also deletes all non-base language pages. If the definition is a non-base language page, the system deletes only the selected non-base language page.

---

If you select multiple pages to delete, but the base language page *is not included* in the selection, the system prompts you for lock comments for each selected non-base language page. If you select multiple pages to delete and the base language page *is included* in the selection, the system prompts the user for lock comments for only the base language page.

### See Also

*Enterprise PeopleTools 8.50 PeopleBook: Global Technology*, "Developing Global Applications," Designing Global-Ready Pages

## Viewing Change Control History

For every PeopleSoft Application Designer development definition in the database, you can view its Change Control history.

To view Change Control history:

1. Select Tools, Change Control, View History.

The History dialog box appears.

2. Select a definition type and name.

The Definition Name list contains only the names of PeopleSoft Application Designer definitions that have at least one history entry. Click the Refresh button to ensure that you're viewing the most recent listing of locked definitions.

The history table contains these columns:

<b>Date</b>	Displays when each entry was added.
<b>User</b>	Displays who added the entry.
<b>Action</b>	Displays why the comment was entered. Five of these Action types— <i>Lock</i> , <i>Informational</i> , <i>Unlock</i> , <i>Rename</i> , and <i>Delete</i> —represent actions that you perform in PeopleSoft Application Designer, and for which you're prompted for comments.
<b>Add</b>	Denotes an automatic history entry that PeopleTools inserts when a new definition is created. In this case—and whenever the system performs a behind-the-scenes lock or unlock—you aren't prompted for comments. The Comment column contains the text <i>System Generated</i> .
<hr/> <b>Note.</b> PeopleSoft Application Designer performs automatic locks and unlocks under certain situations. For example, when you rename a locked definition, that definition must be unlocked before the rename and relocked afterward. Likewise, when you delete a locked definition, the definition is automatically unlocked after the deletion. The system does not prompt you for comments during any of these unlock or relock actions, but corresponding history entries are added automatically. <hr/>	
<b>Copy</b>	This type of history entry is added automatically when a definition is copied into the current database. In this case, the Project value is the name of the copied project in the source database, and no comments are added.
<b>Project, Incident ID, and Comment</b>	Displays the project name, incident ID, and relevant comments.

3. Double-click a row in the grid to open a history entry, if necessary.

The History Details dialog box appears. This dialog box is a read-only version of the Insert Comments dialog box. You cannot update the information that is displayed in this dialog box.

Reporting Change Control Information

Currently, PeopleSoft doesn't deliver predefined reports for retrieving Change Control information. However, you can create your own reports by querying the Change Control tables.

The two tables that you can use for reporting are the Change Control History table (PSCHGCTLHIST) and the Change Control Locking table (PSCHGCTLLOCK). These tables have an almost identical column structure:

Num	Field Name	Type	Len	Format	H	Short Name	Long Name
1	OPRID	Char	8	Mixed		Operator	Operator Id
2	OBJECTTYPE	Nbr	5	Raw B		ObjectType	Object Type
3	OBJECTID1	Nbr	5	Raw B		Object Id 1	Object Id 1
4	OBJECTVALUE1	Char	30	Mixed		Object Value 1	Object Value 1
5	OBJECTID2	Nbr	5	Raw B		Object Id 2	Object Id 2
6	OBJECTVALUE2	Char	30	Mixed		Object Value 2	Object Value 2
7	OBJECTID3	Nbr	5	Raw B		Object Id 3	Object Id 3
8	OBJECTVALUE3	Char	30	Mixed		Object Value 3	Object Value 3
9	OBJECTID4	Nbr	5	Raw B		Object Id 4	Object Id 4
10	OBJECTVALUE4	Char	30	Mixed		Object Value 4	Object Value 4
11	DTTM_STAMP	DtTm	26	Scnds		Date/Time	Date/Time Stamp
12	CHGCTRL_ACTION	Char	1	Upper		Action	Change Control Action
13	PROJECTNAME	Char	30	Upper		Project	Project Name
14	INCIDENT_ID	Char	18	Mixed		Incident	Incident ID
15	DESCRLONG	Long	0			Descr	Description

### Structure of PSCHGCTLHIST

The main difference between these two tables is that PSCHGCTLHIST contains a CHGCTRL\_ACTION field, while PSCHGCTLLOCK does not.

Each PeopleSoft definition in these tables is uniquely identified by numeric codes (DEFINITIONID columns) and names (DEFINITIONVALUE columns). The different DEFINITIONID and DEFINITIONVALUE column pairs correspond to the various definition key types and values for each kind of definition. You can see these definition keys when you view the upgrade definition window. For example, translate values have four keys—Field Name, Field Value, Language Code, and Effective Date, as in this example:

Translate Values Key									
	Field Name	Language Code	Field Value	Effective Date	Source	Target	Action	Upgrade	Done
1	AE_ABEND_ACTION	ENG	A	1900-01-01	Unknown	Unknown	Copy	<input checked="" type="checkbox"/>	<input type="checkbox"/>
2	AE_ABEND_ACTION	ENG	B	1900-01-01	Unknown	Unknown	Copy	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3	AE_ABEND_ACTION	ENG	I	1900-01-01	Unknown	Unknown	Copy	<input checked="" type="checkbox"/>	<input type="checkbox"/>
4	AE_ABEND_ACTION	ENG	S	1900-01-01	Unknown	Unknown	Copy	<input checked="" type="checkbox"/>	<input type="checkbox"/>
5	AE_ACTIVE_STATUS	ENG	A	1900-01-01	Unknown	Unknown	Copy	<input checked="" type="checkbox"/>	<input type="checkbox"/>
6	AE_ACTIVE_STATUS	ENG	A	1997-02-03	Unknown	Unknown	Copy	<input checked="" type="checkbox"/>	<input type="checkbox"/>
7	AE_ACTIVE_STATUS	ENG	I	1900-01-01	Unknown	Unknown	Copy	<input checked="" type="checkbox"/>	<input type="checkbox"/>
8	AE_ACTIVE_STATUS	ENG	I	1997-02-03	Unknown	Unknown	Copy	<input checked="" type="checkbox"/>	<input type="checkbox"/>
9	AE_ADJUST_STATUS	ENG	A	1900-01-01	Unknown	Unknown	Copy	<input checked="" type="checkbox"/>	<input type="checkbox"/>
10	AE_ADJUST_STATUS	ENG	D	1900-01-01	Unknown	Unknown	Copy	<input checked="" type="checkbox"/>	<input type="checkbox"/>
11	AE_ADJUST_STATUS	ENG	M	1900-01-01	Unknown	Unknown	Copy	<input checked="" type="checkbox"/>	<input type="checkbox"/>
12	AE_ADJUST_STATUS	ENG	X	1900-01-01	Unknown	Unknown	Copy	<input checked="" type="checkbox"/>	<input type="checkbox"/>

### Viewing definition keys

In the Change Control tables, the row containing the first translate value in the preceding example has the following field values:

DEFINITIONVALUE1	DEFINITIONVALUE2	DEFINITIONVALUE3	DEFINITIONVALUE4
AE_ABEND_ACTION	A	ENG	1900-01-01

When reporting on a particular definition type, you want to retrieve definition values, but you must limit the query by using the definition IDs for the definition type. The following tables list all of the upgradable definition types, their corresponding definition ID codes, and the type of value that each ID represents (in parentheses).

### ***PeopleTools Definition Types***

This table lists PeopleTools definition types and their definition IDs:

<b><i>Definition Type</i></b>	<b><i>DEFINITIONID1</i></b>	<b><i>DEFINITIONID2</i></b>	<b><i>DEFINITIONID3</i></b>	<b><i>DEFINITIONID4</i></b>
Access Groups	17 (name)	0	0	0
Activities	18 (name)	0	0	0
Application Engine Programs	66 (name)	Not applicable (NA)	NA	NA
Application Engine Sections	66 (name)	77 (section)	NA	NA
Application Package	104 (name)	NA	NA	NA
Approval Rule Sets	85 (name)	21 (effective date)	NA	NA
Business Interlink	64 (name)	NA	NA	NA
Business Processes	7 (name)	NA	NA	NA
Colors	19 (name)	25 (user ID)	0	0
Components	10 (name)	39 (market)	NA	NA
Component Interfaces	74 (name)	NA	NA	NA
Cube Definitions	54 (name)	55 (description)	NA	NA
Cube Instance Definitions	56 (name)	57 (description)	NA	NA

<i>Definition Type</i>	<i>DEFINITIONID1</i>	<i>DEFINITIONID2</i>	<i>DEFINITIONID3</i>	<i>DEFINITIONID4</i>
Dimensions	51 (name)	52 (dimension type)	53 (description)	0
Fields	6 (name)	0	0	0
Field Formats	23 (family name)	0	0	0
File Layout Definitions	71 (name)	NA	NA	NA
HTML	90 (name)	95 (type)	NA	NA
Images	91 (name)	95 (type)	NA	NA
Indexes	1 (name)	24 (index ID)	0	0
Job Definitions	27 (name)	NA	NA	NA
Menus	3 (name)	0	0	0
Message Catalog Entries	48 (message set number)	48 (message number)	16 (language code)	50 (description)
Message Channels	61 (name)	NA	NA	NA
Message Definitions	60 (name)	NA	NA	NA
Mobile Page <b>Important!</b> PeopleSoft Mobile Agent is a deprecated product. These features exist for backward compatibility only.	111 (name)	NA	NA	NA
Message Nodes	62 (name)	NA	NA	NA
Pages	9 (name)	NA	0	0

<b>Definition Type</b>	<b>DEFINITIONID1</b>	<b>DEFINITIONID2</b>	<b>DEFINITIONID3</b>	<b>DEFINITIONID4</b>
PeopleCode	See PeopleCode Definition Types	NA	NA	NA
Problem Type	109 (name)	NA	NA	NA
Process Definitions	29 (process type)	28 (name)	0	0
Process Type Definitions	29 (name)	26 (operating system)	20 (database type)	NA
Queries	30 (name)	25 (user ID)	0	0
Records	1 (name)	2 (RecField name)	0	0
Recurrence Definitions	31 (name)	NA	NA	NA
Roles	32 (name)	0	0	0
Server Definitions	33 (name)	NA	NA	NA
SQL (Structured Query Language)	65 (name)	81 (SQL type)	NA	NA
Styles	35 (name)	0	0	0
Style Sheets	94 (name)	NA	NA	NA
Translate Values	6 (database field name)	22 (value)	21 (effective date)	NA
Trees	34 (setID)	68 (user key value)	36 (tree name)	21 (effective date)
Tree Structures	37 (name)	0	0	0

### ***PeopleCode Definition Types***

This table lists PeopleCode definition types and their definition IDs:

<i>Definition Type</i>	<i>DEFINITIONID1</i>	<i>DEFINITIONID2</i>	<i>DEFINITIONID3</i>	<i>DEFINITIONID4</i>
Application Engine	66 (PeopleSoft Application Engine program)	77 (section, market, database type, effective date)	78 (step)	12 (method)
Application Package	104 (application package)	NA	NA	NA
Component Interface	74 (business component)	12 (method)	NA	NA
Menu	3 (menu)	4 (bar)	5 (item)	12 (method)
Message	60 (message)	12 (method)	NA	NA
Page	9 (panel)	16 (language code)	12 (method)	NA
Page Field	9 (panel)	16 (language code)	67 (field)	12 (method)
Component	10 (panel group)	39 (market)	12 (method)	NA
Component Record	10 (panel group)	39 (market)	1 (record)	12 (method)
Component Record Field	10 (panel group)	39 (market)	1 (record)	2 (fieldname, method)
Record	1 (record)	2 (field)	12 (method)	NA
Subscription	60 (message)	87 (subscription)	12 (method)	NA

### ***Change Control Supported Definition Types***

These are the supported definition types for change control:

- Activity
- Application Engine Program
- Approval Rule Set
- Business Interlink
- Business Process

- Component
- Component Interface
- Field
- File Layout
- HTML
- Image
- Menu
- Message
- Message Channel
- Mobile Page
- Problem Type
- Page
- Project
- Record
- SQL
- Style Sheet

When reporting on Change Control history, consider one other field: CHGCTRL\_ACTION. This field stores the one-letter code for the various actions that Change Control history tracks:

- A (Add)
- C (Copy)
- D (Delete)
- I (Informational)
- L (Lock)
- R (Rename)
- U (Unlock)

Here's an example of a SQL query to report on all deleted definitions:

```
select oprid, definitionvalue1, definitionvalue2,  
definitionvalue3, definitionvalue4, dtm-_stamp, projectname,  
incident_id, descrlong  
from pschgctlhist  
  where chgctrl_action = 'D'  
  order by oprid, definitionvalue1
```

---

**Note.** Full history tracking is supported only for PeopleSoft Application Designer definitions—business processes, business process maps, fields, menus, panels, panel groups, projects, and records. Other definition types have history entries only when the CHGCTL\_ACTION field value is *C*, and only if they've been upgraded.

---



## Appendix A

# Understanding Functional Indexes

This appendix discusses functional indexes for:

- Microsoft SQL Server 2000
- DB2/UDB for Linux, UNIX, and Windows
- Informix

---

## Functional Indexes for Microsoft SQL Server 2000

The following example shows how the key fields are concatenated:

```

USE FS840U70
go
SET IMPLICIT_TRANSACTIONS ON
go
IF EXISTS (SELECT 'X' FROM SYSOBJECTS WHERE TYPE = 'U' AND NAME =
'PS_QE_ALTER_NEW') DROP TABLE PS_QE_ALTER_NEW
go
CREATE TABLE PS_QE_ALTER_NEW (QE_ALTER_FLD1 NCHAR(1) NOT NULL,
    QE_ALTER_FLD2 NCHAR(1) NOT NULL,
    QE_ALTER_FLD3 NCHAR(1) NOT NULL,
    QE_ALTER_FLD4 NCHAR(10) NOT NULL,
    QE_ALTER_FLD5 NCHAR(20) NOT NULL,
    QE_ALTER_FLD6 NCHAR(11) NOT NULL,
    QE_ALTER_FLD7 NCHAR(1) NOT NULL,
    QE_ALTER_FLD8 NCHAR(1) NOT NULL,
    QEPC_AGE SMALLINT NOT NULL,
    QEPC_FILE_NUM DECIMAL(6, 2) NOT NULL,
    QE_BEGIN_DT PSDATE NULL,
    QE_BIRTHDATE PSDATE NULL,
    QE_31DIGFLD9 DECIMAL(31, 8) NOT NULL,
    QE_ANNUAL_PCT DECIMAL(3, 1) NOT NULL,
    QE_MS_DTTM1 PSDATETIME NULL,
    QE_MS_DTTM2 PSDATETIME NULL,
    QE_ALTER_TIMEA PTIME NULL,
    QE_ALTER_TIMEB PTIME NULL,
    MSSCONCATCOL AS QE_ALTER_FLD1 + QE_ALTER_FLD2 + QE_ALTER_FLD3 +
QE_ALTER_FLD4 + QE_ALTER_FLD5 + QE_ALTER_FLD6 + QE_ALTER_FLD7 +
QE_ALTER_FLD8 + CONVERT(NCHAR,QEPC_AGE) + CONVERT(NCHAR
,QEPC_FILE_NUM) + CONVERT(NCHAR(16),QE_BEGIN_DT, 109) +
CONVERT(NCHAR(16),QE_BIRTHDATE, 109) + CONVERT(NCHAR,QE_31DIGFLD9) +
CONVERT(NCHAR,QE_ANNUAL_PCT) + CONVERT(NCHAR(16),QE_MS_DTTM1, 109) +
CONVERT(NCHAR(16),QE_MS_DTTM2, 109) + CONVERT(NCHAR(16)
,QE_ALTER_TIMEA, 109))
go
COMMIT
go
SET ARITHABORT ON
go
CREATE UNIQUE CLUSTERED INDEX PS_QE_ALTER_NEW ON PS_QE_ALTER_NEW
(MSSCONCATCOL)
go
CREATE NONCLUSTERED INDEX PSWQE_ALTER_NEW ON PS_QE_ALTER_NEW
(QE_ALTER_FLD1,
    QE_ALTER_FLD2,
    QE_ALTER_FLD3,
    QE_ALTER_FLD4,
    QE_ALTER_FLD5,
    QE_ALTER_FLD6,
    QE_ALTER_FLD7,
    QE_ALTER_FLD8,
    QEPC_AGE,
    QEPC_FILE_NUM,
    QE_BEGIN_DT,
    QE_BIRTHDATE,
    QE_31DIGFLD9,
    QE_ANNUAL_PCT,
    QE_MS_DTTM1,
    QE_MS_DTTM2)
go
COMMIT
Go

```

## DB2/UDB for Linux, UNIX, and Windows

The following example shows how the key fields are concatenated:

```
CREATE TABLE PS_QE_ALTER_NEW (QE_ALTER_FLD1 VARGRAPHIC(1) NOT NULL,
    QE_ALTER_FLD2 VARGRAPHIC(1) NOT NULL,
    QE_ALTER_FLD3 VARGRAPHIC(1) NOT NULL,
    QE_ALTER_FLD4 VARGRAPHIC(10) NOT NULL,
    QE_ALTER_FLD5 VARGRAPHIC(20) NOT NULL,
    QE_ALTER_FLD6 VARGRAPHIC(11) NOT NULL,
    QE_ALTER_FLD7 VARGRAPHIC(1) NOT NULL,
    QE_ALTER_FLD8 VARGRAPHIC(1) NOT NULL,
    QEPC_AGE VARGRAPHIC(8) NOT NULL,
    QEPC_FILE_NUM DECIMAL(6, 2) NOT NULL,
    QE_BEGIN_DT DATE,
    QE_BIRTHDATE DATE,
    QE_31DIGFLD9 DECIMAL(31, 8) NOT NULL,
    QE_ANNUAL_PCT DECIMAL(3, 1) NOT NULL,
    QE_MS_DTTM1 TIMESTAMP,
    QE_MS_DTTM2 TIMESTAMP,
    QE_ALTER_TIMEA TIME,
    QE_ALTER_TIMEB TIME,
    DBXCONCATCOL GENERATED ALWAYS AS (QE_ALTER_FLD1 || QE_ALTER_FLD2 ||
QE_ALTER_FLD3 || QE_ALTER_FLD4 || QE_ALTER_FLD5 || QE_ALTER_FLD6 ||
QE_ALTER_FLD7 || QE_ALTER_FLD8 || QEPC_AGE ||
VARGRAPHIC(CHAR(QEPC_FILE_NUM)) ||
VALUE(VARGRAPHIC(CHAR(QE_BEGIN_DT)), '')) ||
VALUE(VARGRAPHIC(CHAR(QE_BIRTHDATE)), '')) ||
VARGRAPHIC(CHAR(QE_31DIGFLD9)) || VARGRAPHIC(CHAR(QE_ANNUAL_PCT)) ||
VALUE(VARGRAPHIC(CHAR(QE_MS_DTTM1)), '')) ||
VALUE(VARGRAPHIC(CHAR(QE_MS_DTTM2)), '')) ||
VALUE(VARGRAPHIC(CHAR(QE_ALTER_TIMEA)), ''))) IN PTAPP INDEX IN
PTAPPIDX NOT LOGGED INITIALLY;
COMMIT;
CREATE UNIQUE INDEX PS_QE_ALTER_NEW ON PS_QE_ALTER_NEW (DBXCONCATCOL);
CREATE INDEX PSWQE_ALTER_NEW ON PS_QE_ALTER_NEW (QE_ALTER_FLD1,
    QE_ALTER_FLD2,
    QE_ALTER_FLD3,
    QE_ALTER_FLD4,
    QE_ALTER_FLD5,
    QE_ALTER_FLD6,
    QE_ALTER_FLD7,
    QE_ALTER_FLD8,
    QEPC_AGE,
    QEPC_FILE_NUM,
    QE_BEGIN_DT,
    QE_BIRTHDATE,
    QE_31DIGFLD9,
    QE_ANNUAL_PCT,
    QE_MS_DTTM1,
    QE_MS_DTTM2);
COMMIT;
```

## Functional Indexes for Informix

The following example shows how the key fields are concatenated:

```

BEGIN WORK;
CREATE TABLE PS_QE_ALTER_NEW (QE_ALTER_FLD1 CHAR(1) NOT NULL,
    QE_ALTER_FLD2 CHAR(1) NOT NULL,
    QE_ALTER_FLD3 CHAR(1) NOT NULL,
    QE_ALTER_FLD4 CHAR(10) NOT NULL,
    QE_ALTER_FLD5 CHAR(20) NOT NULL,
    QE_ALTER_FLD6 CHAR(11) NOT NULL,
    QE_ALTER_FLD7 CHAR(1) NOT NULL,
    QE_ALTER_FLD8 CHAR(1) NOT NULL,
    QEPC_AGE SMALLINT NOT NULL,
    QEPC_FILE_NUM DECIMAL(6, 2) NOT NULL,
    QE_BEGIN_DT DATE,
    QE_BIRTHDATE DATE,
    QE_31DIGFLD9 DECIMAL(31, 8) NOT NULL,
    QE_ANNUAL_PCT DECIMAL(3, 1) NOT NULL,
    QE_MS_DTTM1 DATETIME YEAR TO FRACTION(3),
    QE_MS_DTTM2 DATETIME YEAR TO FRACTION(3),
    QE_ALTER_TIMEA DATETIME HOUR TO FRACTION(3),
    QE_ALTER_TIMEB DATETIME HOUR TO FRACTION(3)) IN PTAPP LOCK MODE ROW;
REVOKE ALL ON PS_QE_ALTER_NEW FROM PUBLIC;
COMMIT;
BEGIN WORK;
CREATE FUNCTION PS_QE_ALTER_NEW_SP1(
    QE_ALTER_FLD1 CHAR(1),
    QE_ALTER_FLD2 CHAR(1),
    QE_ALTER_FLD3 CHAR(1),
    QE_ALTER_FLD4 CHAR(10),
    QE_ALTER_FLD5 CHAR(20),
    QE_ALTER_FLD6 CHAR(11),
    QE_ALTER_FLD7 CHAR(1),
    QE_ALTER_FLD8 CHAR(1),
    QEPC_AGE SMALLINT,
    QEPC_FILE_NUM DECIMAL,
    QE_BEGIN_DT DATE,
    QE_BIRTHDATE DATE,
    QE_31DIGFLD9 DECIMAL,
    QE_ANNUAL_PCT DECIMAL,
    QE_MS_DTTM1 DATETIME YEAR TO FRACTION(3),
    QE_MS_DTTM2 DATETIME YEAR TO FRACTION(3))
    RETURNING VARCHAR(166) WITH (NOT VARIANT);
RETURN QE_ALTER_FLD1||
    QE_ALTER_FLD2||
    QE_ALTER_FLD3||
    QE_ALTER_FLD4||
    QE_ALTER_FLD5||
    QE_ALTER_FLD6||
    QE_ALTER_FLD7||
    QE_ALTER_FLD8||
    QEPC_AGE||
    QEPC_FILE_NUM||
    QE_BEGIN_DT||
    QE_BIRTHDATE||
    QE_31DIGFLD9||
    QE_ANNUAL_PCT||
    QE_MS_DTTM1||
    QE_MS_DTTM2;
END FUNCTION;
CREATE FUNCTION PS_QE_ALTER_NEW_SP2(
    QE_ALTER_TIMEA DATETIME HOUR TO FRACTION(3))
    RETURNING VARCHAR(15) WITH (NOT VARIANT);
RETURN QE_ALTER_TIMEA;
END FUNCTION;
CREATE UNIQUE INDEX PS_QE_ALTER_NEW ON PS_QE_ALTER_NEW
    (PS_QE_ALTER_NEW_SP1(

```

```
QE_ALTER_FLD1,
QE_ALTER_FLD2,
QE_ALTER_FLD3,
QE_ALTER_FLD4,
QE_ALTER_FLD5,
QE_ALTER_FLD6,
QE_ALTER_FLD7,
QE_ALTER_FLD8,
QEPC_AGE,
QEPC_FILE_NUM,
QE_BEGIN_DT,
QE_BIRTHDATE,
QE_31DIGFLD9,
QE_ANNUAL_PCT,
QE_MS_DTTM1,
QE_MS_DTTM2),
PS_QE_ALTER_NEW_SP2(
QE_ALTER_TIMEA)) IN FS840IDX;
CREATE INDEX PSWQE_ALTER_NEW ON PS_QE_ALTER_NEW (QE_ALTER_FLD1,
QE_ALTER_FLD2,
QE_ALTER_FLD3,
QE_ALTER_FLD4,
QE_ALTER_FLD5,
QE_ALTER_FLD6,
QE_ALTER_FLD7,
QE_ALTER_FLD8,
QEPC_AGE,
QEPC_FILE_NUM,
QE_BEGIN_DT,
QE_BIRTHDATE,
QE_31DIGFLD9,
QE_ANNUAL_PCT,
QE_MS_DTTM1,
QE_MS_DTTM2) IN FS840IDX;
COMMIT;
```



## Appendix B

# Referencing Selections for Command Line Parameters

This appendix lists various values that are useful when specifying command line parameters. Specifically, these tables are referenced:

- Database type selections.
- Definition type selections.
- Language selections.
- Available release numbers.

---

## Database Type Selections

Use the following database types:

<b><i>Database</i></b>	<b><i>Database Description</i></b>
DB2ODBC	DB2 UDB for OS/390 and z/OS
DB2UNIX	DB2 UDB for Linux, Unix, and Windows
INFORMIX	Informix
MICROSFT	Microsoft SQL Server
ORACLE	Oracle
SYBASE	Sybase

## Definition Type Selections

Use the following numbers to describe the corresponding definition:

<i>Number</i>	<i>Definition Description</i>
0	Record
1	Index
2	Field
3	Format definition
4	Translate
5	Page
6	Menu
7	Components
8	Record PeopleCode
9	Menu PeopleCode
10	Query
11	Tree structure
12	Tree*
13	Access group*
14	Color
15	Style

<b><i>Number</i></b>	<b><i>Definition Description</i></b>
16	Not used
17	Business process
18	Activity
19	Role*
20	Process definition
21	Process server
22	Process type
23	Process job
24	Process recurrence
25	Message catalog entries*
26	Dimension*
27	Cube definitions*
28	Cube instance definitions*
29	Business interlink
30	SQL definition
31	File layout definition
32	Component interfaces
33	Application Engine program

<b><i>Number</i></b>	<b><i>Definition Description</i></b>
34	Application Engine section
35	Message node
36	Message channel
37	Message definition
38	Approval rule set
39	Message PeopleCode
40	Subscription PeopleCode
41	Not used
42	Component interface PeopleCode
43	Application engine PeopleCode
44	Page PeopleCode
45	Page field PeopleCode
46	Component PeopleCode
47	Component record PeopleCode
48	Component record field PeopleCode
49	Image
50	Style sheet
51	HTML

<i>Number</i>	<i>Definition Description</i>
52	Not used
53	Permission list
54	Portal registry definitions
55	Portal registry structures
56	URL definitions
57	Application Packages
58	Application Package PeopleCode
59	Portal Registry User homepage
60	Problem type definition
61	Archive templates (deprecated in PeopleTools release 8.44 and above).
62	XSLT
63	Portal Registry User Favorite
64	Mobile page  <b>Important!</b> PeopleSoft Mobile Agent is a deprecated product. These features exist for backward compatibility only.
65	Relationships
66	Component Interface Property PeopleCode**
67	Optimization Models***
68	File References***

<b>Number</b>	<b>Definition Description</b>
69	File Reference Type Code***
70	Archive object definitions***
71	Archive Templates (Type 2)***
72	Diagnostic Plug-Ins***

\* These definition types do not have Copy To File or Copy From File process support.

\*\* This definition type is supported only in PeopleTools 8.42 and later.

\*\*\* These definition types are supported only in PeopleTools 8.44 and later.

---

## Language Selections

Use the following codes to describe the supported languages:

<b>Code</b>	<b>Language Description</b>
CFR	Canadian French
DAN	Danish
DUT	Dutch
ENG	English
ESP	Spanish
FRA	French
GER	German
GRK	Greek
ITA	Italian

<b>Code</b>	<b><i>Language Description</i></b>
JPN	Japanese
KOR	Korean
MAY	Bhasa Malay
POL	Polish
POR	Portuguese
SVE	Swedish
THA	Thai
ZHS	Simplified Chinese
ZHT	Traditional Chinese
COMMON	Nonlanguage attributes

---

## Available Release Numbers

The available release numbers include:

- Core 5.10.00.000
- Core 6.00.00.000
- Core 6.10.00.000
- Core 7.00.00.000
- Core 7.50.00.000
- Core 8.00.00.000
- Core 8.10.00.000
- Core 8.40.00.000



## Appendix C

# Using Default Style Classes

This appendix provides an overview of default style classes.

---

## Default Style Classes

This table lists the many of the default style classes that are shipped with your system, the sub style to which each belongs, and a brief description of the style class:

<b><i>Style Class Name</i></b>	<b><i>Sub Style</i></b>	<b><i>Description</i></b>
EDGE	PTSTYLEDEF	*Default PeopleSoft Pure Internet Architecture navigation.
EOPP_SCCHILDINDENT	PTSTYLEDEF	Navigation collection child folder indentation.
EOPP_SCPAGEBORDER	PTSTYLEDEF	Navigation collection page border.
EOPP_SCSECTIONCONTENTDARK	PTSTYLEDEF	Navigation collection section content dark style.
EOPP_SCSECTIONFOLDERDARK	PTSTYLEDEF	Navigation collection section folder dark style.
EOPP_SCSECTIONCONTENT	PTSTYLEDEF	Navigation collection section content.
EOPP_SCSECTIONFOLDER	PTSTYLEDEF	Navigation collection section folder.
EOPP_SCPAGETITLESECTION	PTSTYLEDEF	Navigation collection page title.

<b>Style Class Name</b>	<b>Sub Style</b>	<b>Description</b>
EOPP_SCBREADCRUMBSECTION	PTSTYLEDEF	Navigation collection breadcrumbs.
EOPP_SCADDITIONALTEXT	PTSTYLEDEF	Navigation collection additional text.
EOPP_SCPAGEDESCRSECTION	PTSTYLEDEF	Navigation collection page description.
EOPP_SCCHILDCONTENTLINK	PTSTYLEDEF	Navigation collection child content link.
EOPP_SCCHILDFOlderLINK	PTSTYLEDEF	Navigation collection child folder link.
EOPP_SCCHILDINDENT	PTSTYLEDEF	Navigation Collection child folder indentation.
EOPP_SCMORELINK	PTSTYLEDEF	Navigation collection more link.
EOPP_SCPAGEBORDER	PTSTYLEDEF	Navigation Collection page border.
EOPP_SCSECTIONCONTENT	PTSTYLEDEF	Navigation Collection section content.
EOPP_SCSECTIONCONTENTDARK	PTSTYLEDEF	Navigation Collection section content dark style.
EOPP_SCSECTIONFOLDERDARK	PTSTYLEDEF	Navigation Collection section folder dark style.
EOPP_SCSECTIONCONTENTLINK	PTSTYLEDEF	Navigation collection section content link.
EOPP_SCSECTIONFOLDERLINK	PTSTYLEDEF	Navigation collection section folder link.
HEADERLINK	PTSTYLEDEF	Used by Branding area links on a page.

<b>Style Class Name</b>	<b>Sub Style</b>	<b>Description</b>
HEADERLINKACTIVE	PTSTYLEDEF	Used by Branding area links on a page that are active.
HILEFT1	PTSTYLEDEF	*Default PeopleSoft Pure Internet Architecture navigation
HILEFT2	PTSTYLEDEF	*Default PeopleSoft Pure Internet Architecture navigation.
KNOCKOUT	PTSTYLEDEF	*Portal administration left navigation.
LEVEL1	PTSTYLEDEF	*Default PeopleSoft Pure Internet Architecture navigation.
LEVEL2	PTSTYLEDEF	*Default PeopleSoft Pure Internet Architecture navigation.
LEVEL3	PTSTYLEDEF	*Default PeopleSoft Pure Internet Architecture navigation.
MID	PTSTYLEDEF	Links at the first level of the PeopleSoft Pure Internet Architecture navigation.
MUTE	PTSTYLEDEF	Descriptive text in the PeopleSoft Pure Internet Architecture navigation.
PAADDITIONALINSTRUCTIONS	PSALTERNATE	Additional instructions on the page (slightly smaller font).
PABACKGROUNDINVISIBLE	PSALTERNATE	Body background of hidden group boxes and grids.
PABOLDBLUETEXT	PSALTERNATE	Strong emphasis for text on a page.

<b>Style Class Name</b>	<b>Sub Style</b>	<b>Description</b>
PABOLDTEXT	PSALTERNATE	Field value emphasis.
PACALENDARDAYNUM	PSALTERNATE	Text for numeric days on calendars.
PACALENDARDAYNUMDISABLED	PSALTERNATE	Text for numeric days on calendars (disabled state).
PACFGRIDFOOT	PSALTERNATE	Footer borders for ChartField grids.
PACFGRIDLABEL	PSALTERNATE	Header label background for ChartField grids.
PAERRORTTEXT	PSALTERNATE	Inline error message text font type, color, and size.
PAEXAMPLE	PSALTERNATE	Smaller font size to demonstrate an example.
PAEXPANDCOLLAPSE	PSALTERNATE	Expandable and collapsible area borders.
PAFRAMEBACKGROUND	PSALTERNATE	Same as the PAGROUPDIVIDER style class, except that it allows text on the dividing bar for self-service pages.
PAFRAMELEVEL1	PSALTERNATE	Level 1 frames.
PAFRAMELEVEL2	PSALTERNATE	Level 2 frames.
PAFRAMELEVEL3	PSALTERNATE	Level 3 frames.
PAGRIDHEADER	PSALTERNATE	Grid column headers for self-service grids without navigation controls.
PAGROUPBOX	PSALTERNATE	Group boxes.

<b>Style Class Name</b>	<b>Sub Style</b>	<b>Description</b>
PAGROUPBOXLABELINVISIBLE	PSALTERNATE	Invisible group box label.
PAGROUPBOXLABELLEVEL1	PSALTERNATE	Group box labels at level 1.
PAGROUPBOXLABELLEVEL2	PSALTERNATE	Group box labels at level 2.
PAGROUPBOXLABELLEVEL3	PSALTERNATE	Group box labels at level 3.
PAGROUPBOXLEVEL1	PSALTERNATE	Group box body at level 1.
PAGROUPBOXLEVEL2	PSALTERNATE	Group box body at level 2.
PAGROUPBOXLEVEL3	PSALTERNATE	Group box body at level 3.
PAGROUPDIVIDER	PSALTERNATE	Group box label style on self-service pages.
PAGROUPDIVIDERBODY	PSALTERNATE	Group box body style on developer-defined advanced search pages.
PAHEADERTOOLBARBODY	PSALTERNATE	Frame background for the customer relationship management (CRM) toolbar.
PAHEADERTOOLBARFRAME	PSALTERNATE	Frame borders for the CRM toolbar.
PAHORIZONTALRULELEVEL1	PSALTERNATE	Horizontal rule at level 1.
PAHORIZONTALRULELEVEL2	PSALTERNATE	Horizontal rule at level 2.
PAHORIZONTALRULELEVEL3	PSALTERNATE	Horizontal rule at level 3.
PAHYPERLINKDESCR	PSALTERNATE	Text description for links on application homepages.
PAHYPERLINKERRSTAT	PSALTERNATE	Links conveying an error status (use with image).

<b>Style Class Name</b>	<b>Sub Style</b>	<b>Description</b>
PAHYPERLINKLVL1ODD	PSALTERNATE	Link style for grid odd rows that use the PeopleCode style property.
PAHYPERLINKSUCCESSSTAT	PSALTERNATE	Links conveying a successful status (use with image).
PAHYPERLINKWARNSTAT	PSALTERNATE	Links conveying a warning status (use with image).
PALEGENDBACKGROUND	PSALTERNATE	Group box body color for legends on pages.
PALEVEL0PRIMARY	PSALTERNATE	Level 0, high-level key data on self-service pages.
PALEVEL0SECONDARY	PSALTERNATE	Additional fields to identify the object.
PAPAGEINSTRUCTIONS	PSALTERNATE	Text instructions on the top of the page.
PAPAGETITLE	PSALTERNATE	Page title.
PASEARCHBOXBODY	PSALTERNATE	Group boxes around search criteria.
PASTEPBYSTEPTITLE	PSALTERNATE	Step-by-step transaction.
PASUBTABLINKACTIVE	PSALTERNATE	Active Sub tab style.
PASUBTABLINKINACTIVE	PSALTERNATE	Inactive Sub Tab style.
PATEXTBACKGROUND	PSALTERNATE	Text background.
PATRANSACTIONTITLE	PSALTERNATE	Transaction name above the current page title.
PSACTIVETAB	PTSTYLEDEF	Active tab labels and background colors.

<b>Style Class Name</b>	<b>Sub Style</b>	<b>Description</b>
PSBOTTOMLINE	PTSTYLEDEF	Tab bottom line style to specify height and width. Height can be set to 0 if tab bottom line is not required.
PSCHARTAXISTITLE	PTSTYLEDEF	Chart axis labels.
PSCHARTDEFAULT	PTSTYLEDEF	Default chart text.
PSCHARTTITLE	PTSTYLEDEF	Chart title text.
PSCHECKBOX	PTSTYLEDEF	Check box label.
PSDISABLED	PTSTYLEDEF	Disabled field value font type, size, and color.
PSDROPDOWNLABEL	PTSTYLEDEF	Drop-down list box label.
PSDROPDOWNLIST	PTSTYLEDEF	Drop-down list box text.
PSEDITBOX	PTSTYLEDEF	Edit box label.
PSEDITBOX_DISABLED	PTSTYLEDEF	Disabled edit box style (white background with a thin, gray border).
PSEDITBOX_DISPONLY	PTSTYLEDEF	Display-only edit boxes.
PSEDITBOXLABEL	PTSTYLEDEF	Edit box text.
PSERROR	PTSTYLEDEF	Error field background color and text.
PSFRAME	PTSTYLEDEF	Frame borders.
PSGRIDCOLUMNHDR	PTSTYLEDEF	Grid column header texts.
PSGRIDCOUNTER	PTSTYLEDEF	Color for the row counter text displayed in the grid header.

<b>Style Class Name</b>	<b>Sub Style</b>	<b>Description</b>
PSGRIDEVENROWSORTED	PTSTYLEDEF	Sorted even row style in a grid.
PSGRIDODDROWSORTED	PTSTYLEDEF	Sorted odd row style in a grid.
PSGROUPBOX	PTSTYLEDEF	Group box borders and body style.
PSGROUPBOXLABEL	PTSTYLEDEF	Group box labels.
PSHEADERHYPERLINK	PTSTYLEDEF	Link in a scroll area or grid header.
PSHEADERHYPERLINKD	PTSTYLEDEF	Disabled link in a scroll area or grid header.
PSHEADERTOOLBARFRAME		Meant for PeopleTools usage for buttons.
PSHORIZONTALRULE	PTSTYLEDEF	Horizontal rule.
PSHYPERLINK	PTSTYLEDEF	Link.
PSHYPERLINKACTIVE	PTSTYLEDEF	Selected link.
PSHYPERLINKDISABLED	PTSTYLEDEF	Disabled link.
PSHYPERLINKHOVER	PTSTYLEDEF	Link when the mouse hovers over it.
PSHYPERLINKVISITED	PTSTYLEDEF	Previously visited link.
PSHYPERLINKYELLOWBKGRD	PTSTYLEDEF	Link with a yellow background.
PSIMAGE	PTSTYLEDEF	Image border style, width, and colors.
PSINACTIVETAB	PTSTYLEDEF	Inactive tab labels and background colors.

<b>Style Class Name</b>	<b>Sub Style</b>	<b>Description</b>
PSINACTIVETABHOVER	PTSTYLEDEF	Inactive tab labels and background colors when the mouse hovers over them.
PSLEVEL1GRID	PTSTYLEDEF	Level 1 grid background colors, border settings, and fonts.
PSLEVEL1GRIDACTIVETAB	PTSTYLEDEF	Active tab labels and background colors on level 1 grids.
PSLEVEL1GRIDCOLUMNHDR	PTSTYLEDEF	Grid column labels on level 1 grids.
PSLEVEL1GRIDCOLUMNHDRTXT	PTSTYLEDEF	Level 1 grid column header text style
PSLEVEL1GRIDEVENROW	PTSTYLEDEF	Even rows on level 1 grids.
PSLEVEL1GRIDINACTIVETAB	PTSTYLEDEF	Inactive tab labels and backgrounds on level 1 grids.
PSLEVEL1GRIDINACTIVETABHOVER	PTSTYLEDEF	Inactive tab labels and backgrounds (when mouse hovers over them) on level 1 grids.
PSLEVEL1GRIDLABEL	PTSTYLEDEF	Level 1 grid labels.
PSLEVEL1GRIDNAVIGATIONBAR	PTSTYLEDEF	Navigation bars on level 1 grids.
PSLEVEL1GRIDODDROW	PTSTYLEDEF	Odd rows on level 1 grids.
PSLEVEL1GRIDROW	PTSTYLEDEF	Level 1 grid background settings, border settings, and fonts.
PSLEVEL1SCROLLAREABODY	PTSTYLEDEF	Level 1 scroll area body.
PSLEVEL1SCROLLAREAFOOTER	PTSTYLEDEF	Level 1 scroll area footer.

<b>Style Class Name</b>	<b>Sub Style</b>	<b>Description</b>
PSLEVEL1SCROLLAREAHEADER	PTSTYLEDEF	Level 1 scroll area header.
PSLEVEL2GRID	PTSTYLEDEF	Level 2 grid background colors, border settings, and fonts.
PSLEVEL2GRIDACTIVETAB	PTSTYLEDEF	Active tab labels and background colors on level 2 grids.
PSLEVEL2GRIDCOLUMNHDR	PTSTYLEDEF	Grid column labels on level 2 grids.
PSLEVEL2GRIDCOLUMNHDRTXT	PTSTYLEDEF	Level 2 grid column header text style.
PSLEVEL2GRIDEVENROW	PTSTYLEDEF	Even rows on level 2 grids.
PSLEVEL2GRIDINACTIVETAB	PTSTYLEDEF	Inactive tab labels and backgrounds on level 2 grids.
PSLEVEL2GRIDINACTIVETABHOVER	PTSTYLEDEF	Inactive tab labels and backgrounds (when mouse hovers over it) on level 2 grids.
PSLEVEL2GRIDLABEL	PTSTYLEDEF	Labels on level 2 grids.
PSLEVEL2GRIDNAVIGATIONBAR	PTSTYLEDEF	Navigation bars on level 2 grids.
PSLEVEL2GRIDODDROW	PTSTYLEDEF	Odd rows on level 2 grids.
PSLEVEL2GRIDROW	PTSTYLEDEF	Level 2 grid background settings, border settings, and fonts.
PSLEVEL2SCROLLAREABODY	PTSTYLEDEF	Level 2 scroll area body.
PSLEVEL2SCROLLAREAFOOTER	PTSTYLEDEF	Level 2 scroll area footer.
PSLEVEL2SCROLLAREAHEADER	PTSTYLEDEF	Level 2 scroll area header.

<b>Style Class Name</b>	<b>Sub Style</b>	<b>Description</b>
PSLEVEL3GRID	PTSTYLEDEF	Level 3 grid background settings, border settings, and fonts.
PSLEVEL3GRIDACTIVETAB	PTSTYLEDEF	Active tab labels and background colors on level 3 grids.
PSLEVEL3GRIDCOLUMNHDR	PTSTYLEDEF	Grid column labels on level 3 grids.
PSLEVEL3GRIDCOLUMNHDRTXT	PTSTYLEDEF	Level 3 grid column header text style.
PSLEVEL3GRIDEVENROW	PTSTYLEDEF	Even rows on level 3 grids.
PSLEVEL3GRIDINACTIVETAB	PTSTYLEDEF	Inactive tab labels and backgrounds on level 3 grids.
PSLEVEL3GRIDINACTIVETABHOVER	PTSTYLEDEF	Inactive tab labels and backgrounds (when the mouse hovers over them) on level 3 grids .
PSLEVEL3GRIDLABEL	PTSTYLEDEF	Labels on level 3 grids.
PSLEVEL3GRIDNAVIGATIONBAR	PTSTYLEDEF	Navigation bars on level 3 grids.
PSLEVEL3GRIDODDROW	PTSTYLEDEF	Odd rows on level 3 grids.
PSLEVEL3GRIDROW	PTSTYLEDEF	Level 3 grid background settings, border settings, and fonts.
PSLEVEL3SCROLLAREABODY	PTSTYLEDEF	Level 3 scroll area body.
PSLEVEL3SCROLLAREAFOOTER	PTSTYLEDEF	Level 3 scroll area footer.
PSLEVEL3SCROLLAREAHEADER	PTSTYLEDEF	Level 3 scroll area header.

<b>Style Class Name</b>	<b>Sub Style</b>	<b>Description</b>
PSLONGEDITBOX	PTSTYLEDEF	Long edit box data.
PSLONGEDITLABEL	PTSTYLEDEF	Long edit box labels.
PSMULTILANG	PTSTYLEDEF	Multilingual field background.
PSNAVPARENTLINK	PTSTYLEDEF	Left navigation link color.
PSPAGE	PTSTYLEDEF	Page background color and margins.
PSPAGECONTAINER	PTSTYLEDEF	Background color for the page container.
PSPAGECUST	PTSTYLEDEF	Shaded background for page customization image.
PSPROCESSING	PTSTYLEDEF	Processing message background color.
PSPSMALLTEXT	PTSTYLEDEF	Footer text font type, size, weight, and color.
PSPTABTABLE	PTSTYLEDEF	Style for Portal tab label.
PSPUSHBUTTON	PTSTYLEDEF	Push button labels and background.
PSPUSHBUTTON*	PTSTYLEDEF	Style class for push buttons in non-SWAN mode. For example, ,PSPUSHBUTTONRETURN, when clicked returns to the previous page.
PSPUSHBUTTONDISABLED	PTSTYLEDEF	Disabled push button labels and background color.
PSRADIOBUTTON	PTSTYLEDEF	Radio buttons.

<b>Style Class Name</b>	<b>Sub Style</b>	<b>Description</b>
PSSRIGHTLEFTSCROLL		Style specification for Show following Tabs and Show previous Tabs button.
PSSRCHACTION		Font style for Search action.
PSSRCHCHECKBOX	PTSTYLEDEF	Search check box.
PSSRCHDROPDOWNLABEL	PTSTYLEDEF	Search drop-down labels.
PSSRCHEDITBOXLABEL	PTSTYLEDEF	Search edit box labels.
PSSRCHINSTRUCTIONS	PTSTYLEDEF	Search instructions.
PSSRCHPAGE	PTSTYLEDEF	Search page font type, background color, and margins.
PSSRCHRESULTSEVENROW	PTSTYLEDEF	Even rows on search page result grids.
PSSRCHRESULTSFOOTER	PTSTYLEDEF	Footer on search page result grids.
PSSRCHRESULTSHDR	PTSTYLEDEF	Header on search page result grids.
PSSRCHRESULTSHDRTXT	PTSTYLEDEF	Header text style for search results.
PSSRCHRESULTSHYPERLINK	PTSTYLEDEF	Navigation links above the search results.
PSSRCHRESULTSHYPERLINKD	PTSTYLEDEF	Disabled navigation links above the search results.
PSSRCHRESULTSODDROW	PTSTYLEDEF	Odd rows on search page result grids.
PSSRCHRESULTSTITLE	PTSTYLEDEF	Instructional text on search results page.

<b>Style Class Name</b>	<b>Sub Style</b>	<b>Description</b>
PSSRCHSUBTITLE	PTSTYLEDEF	Search results section title.
PSSRCHTIPS	PTSTYLEDEF	Search tips.
PSSRCHTITLE	PTSTYLEDEF	First-level title on a search page.
PSSSTATICIMAGE	PTSTYLEDEF	Static image border styles, width, and colors.
PSTEXT	PTSTYLEDEF	Static text controls.
PSTLEFTMENU	PTSTYLEDEF	No longer used.
PSTREENODESELECTED	PTSTYLEDEF	Selected tree node.
PSTRELLINKSBODYNARROW	PAALTERNATE	Related links body for narrow templates.
PSTRELLINKSBODYWIDE	PAALTERNATE	Related links body for wide templates.
PSTRELLINKSHDRNARROW	PAALTERNATE	Related links header label for narrow templates.
PSTRELLINKSHDRWIDE	PAALTERNATE	Related links header label for wide templates.
PTBREADCRUMB	PTSTYLEDEF	Breadcrumbs.
PTBREADCRUMBCUR	PTSTYLEDEF	Current location on a breadcrumb trail.
PTBREADCRUMBMARGIN	PTSTYLEDEF	Breadcrumb margins.
PTGRIDROWMOUSEOVER	PTSTYLEDEF	Grid row style on mouse over.
PTGRIDROWSELECT	PTSTYLEDEF	Grid row style for a selected row.

<b>Style Class Name</b>	<b>Sub Style</b>	<b>Description</b>
PT_MCFADDITIONALINSTRUCTIONS	PTSTYLEDEF	Any additional instructions/error messages with small font.
PT_MCFCHATME	PTSTYLEDEF	Self chat message class.
PT_MCFCHATSYSMSG	PTSTYLEDEF	System Chat message.
PT_MCFCHATTHEM	PTSTYLEDEF	Chat message from others.
PT_MCFCHATUS	PTSTYLEDEF	Chat message from another Agent.
PT_MCFCHECKBOX	PTSTYLEDEF	MCF check box.
PT_MCFCHECKBOXLABEL	PTSTYLEDEF	MCF check box label.
PT_MCFDROPDOWNLIST	PTSTYLEDEF	MCF drop-down list box.
PT_MCFEDITBOX	PTSTYLEDEF	MCF edit box.
PT_MCFEVTNAME	PTSTYLEDEF	REN event name in tracer log.
PT_MCFEVTVALUE	PTSTYLEDEF	REN event value in tracer log.
PT_MCFGRAYTEXT	PTSTYLEDEF	Non-editable gray text.
PT_MCFGROUPBOX	PTSTYLEDEF	MCF group box style.
PT_MCFHEADER	PTSTYLEDEF	MCF header style.
PT_MCFLABEL	PTSTYLEDEF	MCF label style.
PT_MCFPAGETITLE	PTSTYLEDEF	MCF page title style.
PT_MCFPUSHBUTTON	PTSTYLEDEF	MCF push button style.

<b>Style Class Name</b>	<b>Sub Style</b>	<b>Description</b>
PT_MCFROW	PTSTYLEDEF	Row style in an MCF table.
PT_MCFSPVSRTABLE	PTSTYLEDEF	Table style for MCF monitoring supervisor.
PT_MCFTABLE	PTSTYLEDEF	Table style.
PT_MCFTABLEHEADER	PTSTYLEDEF	Transparent table header style.
PT_MCFTABLEHEADER1	PTSTYLEDEF	Table header style.
PT_MCFTRACEREVENT	PTSTYLEDEF	REN event in tracer window style
PT_MCFTRACERREQ	PTSTYLEDEF	REN request in tracer window style.
PTCXADMINBAR	PTSTYLEDEF	Context manager "Administer Pagelets" bar style.
PTCXADMINLINK	PTSTYLEDEF	Context Manager "Administer Pagelets" link style.
PTCXMPAGELETBODYBAK	PTSTYLEDEF	Context Manager template Pagelet body background.
PTCXMPAGELETHEADERBAK	PTSTYLEDEF	Context Manager template pagelet header background
PTCXMPGLTHDRLINK	PTSTYLEDEF	Context Manager template pagelet header link.
PTCXMPGLTHDRLINKHOVER	PTSTYLEDEF	Context Manager template Pagelet header link hover.
PTGRIDROWMOUSEOVER	PTSTYLEDEF	Grid row style on mouse over.
PTGRIDROWSELECT	PTSTYLEDEF	Grid row style for a selected row.

<b>Style Class Name</b>	<b>Sub Style</b>	<b>Description</b>
PTMENULINK	PTSTYLEDEF	Link on level 2 of the PeopleSoft Pure Internet Architecture navigation.
PTNAVBACKGROUND	PTSTYLEDEF	Background style in navigation menu.
PTNAVLINK	PTSTYLEDEF	Content reference link style.
PTNAVLINKHOVER	PTSTYLEDEF	Content reference link style when the mouse hovers over it.
PTNAVSEARCHDIVIDER	PTSTYLEDEF	Search divider element style (in navigation).
PTNAVSEARCHLABEL	PTSTYLEDEF	Search label style (in navigation).
PTNAVSELCHILDBK	PTSTYLEDEF	Selected content reference background color (in navigation).
PTNAVSELCHILDEDGEDK	PTSTYLEDEF	Background dark color on the top edge of the selected content reference (in navigation)
PTNAVSELCHILDEDGELT	PTSTYLEDEF	Background color on the top edge of the selected content reference (in navigation).
PTNAVSELCHILDLINK	PTSTYLEDEF	Selected content reference link font type and color (in navigation).
PTNAVSELFOLDERBK	PTSTYLEDEF	Selected folder background color (in navigation).
PTNAVSELPARENTLINK	PTSTYLEDEF	Selected parent link color (in navigation).

<b>Style Class Name</b>	<b>Sub Style</b>	<b>Description</b>
PTNAVSELPARENTBK	PTSTYLEDEF	Selected parent folder background color (in navigation).
PTNAVSELPARENTEDGE	PTSTYLEDEF	Background color on the edge of the selected parent folder (in navigation).
PTPAGELET	PTSTYLEDEF	Homepage pagelet border thickness and color.
PTPAGELETBODY	PTSTYLEDEF	Homepage pagelet font type and size, border thickness, border color, and padding.
PTPAGELETHEADER	PTSTYLEDEF	Homepage pagelet header font type, size, and color; background color, border thickness; border color; and padding.
PTPOPUP_HEADER	PTSTYLEDEF	Header style for pop up page element.
PTPOPUP_TITLE	PTSTYLEDEF	Title style that appears in the header of the popup page element.
PTQUERYSTYLESUB	PTSTYLEDEF	Query style definition.
PTSELECTBODY	PTSTYLEDEF	Portal layout page selection box body.
PTSELECTHEADER	PTSTYLEDEF	Portal layout page selection box header.
PTSRCHRESULTSHDRTXT	PTSTYLEDEF	Text style in Search results header.
PTSPELLCHECKERROR	PTSTYLEDEF	Invalid word for the spell check (highlighted in black with white text).

<b>Style Class Name</b>	<b>Sub Style</b>	<b>Description</b>
PTTRANSPARENT	PTSTYLEDEF	Makes text labels display as transparent so that they are not visible in standard layout mode but are embedded in the HTML so that they can be read by a screen reader.
RESET	PTSTYLEDEF	*Default PeopleSoft Pure Internet Architecture navigation.
SMALL	PTSTYLEDEF	Link on the portal homepage personalization page and portal structure breadcrumbs.
STRONG	PSSTYLEDEF	Emphasized label text size and weight.
STRONGWRAP	PTSTLYEDEF_SWAN	Emphasized text size and weight that will break to the next line in longer sentences.

\* Default PeopleSoft Pure Internet Architecture styles are no longer in use.

---

**Note.** Style classes with the suffix WBO, have border properties. For example, the style PSGROUPBOXWBO enables you to define border properties such as color, width and style under the Border tab of the style property.

Style classes with the suffix NBO, have no border properties. For example, the style PSGROUPBOXNBO enables you to define a border property having *None* as its style.

---



## Appendix D

# Modifying Classic Style Sheet Tabs

This section provides an overview of classic tab definitions and discusses how to create classic tab images.

---

## Understanding Classic Tab Definitions

You can change the background colors of folder tabs in a component and tabs in a grid. To change the background colors of tabs, it is helpful to have a basic understanding of how tab definitions are built and stored in the database.

PeopleSoft stores tab definitions as images. Each tab, or image definition, has three parts that are stored as rows in the database:

- Row 1 contains code representing the top border.
- Row 2 contains the tab itself.
- Row 3 contains the bottom border.

Because tabs look different depending on where they are located relative to other tabs and whether they are active, they must be defined in *sets* for a combination of active and inactive tab colors.



Active General tab with inactive tabs

The tab image names reflect the role of the image in the tab row and the colors that it uses. Colors are encoded using the 6-character, hexadecimal RGB value of the color.

When the system displays tabs on a page, it verifies the existence of all of the images that are required for the pair of colors that is specified by the active and inactive tab styles. If these images are found, the tabs are drawn using these colors. If the system cannot find the images, it uses white (FFFFFF) for the active tab and blue (6666FF) for the inactive tab. If the images for these colors are not in the database, then the tabs do not appear at all.

### ***Classic Tab Image Naming Scheme***

This section describes in detail how classic tab images are named. However, to change tab colors, change only the RGB values.

All tab images are named in this format:

`PT_TABNABBRRGGBB`

- *PT\_TAB*: System-defined image definition name prefix.
- *N*: HTML table row.

Values are 1, 2, or 3. Create images only for 2 or 3.

- *A*: Location of the tab parts in a tab.

Values are:

- *L*: Left end.
- *B*: Between.
- *R*: Right end.
- *M*: Middle of a tab.
- *BB*: Relative position to other tabs.

Values are:

- *AI*: Piece between active and inactive.
- *IA*: Piece between inactive and active.
- *II*: Piece between two inactive tabs.
- *AX*: Piece of an active tab.
- *IX*: Piece of an inactive tab.
- *RRGGBB* is either omitted (when the tab image does not use any color), or it is one or two 6-character hexadecimal codes for inactive and active tab colors.

---

## Modifying Classic Style Sheet Tabs

This section provides an overview of tab definitions and discusses how to create tab images.

### Creating Tab Images

Before you can assign a new background color to the active or inactive tabs, you must create the necessary tab images with that color.

To produce tabs in a different color combination:

1. Open the image definitions for the tabs that you want to change.

Because you want to change only the color of tabs, open only those image definitions that already contain an RGB color code.

If you are changing *both* the active and inactive tab color, access these image definitions:

- PT\_TAB2LAXFFFFFF
- PT\_TAB2RAXFFFFFF
- PT\_TAB2LIXB8B090
- PT\_TAB2RIXB8B090
- PT\_TAB2BAIB8B090FFFFFF
- PT\_TAB2BIAB8B090FFFFFF
- PT\_TAB2BIIB8B090
- PT\_TAB3LAXFFFFFF
- PT\_TAB3RAXFFFFFF
- PT\_TAB3MAXFFFFFF
- PT\_TAB3LIXB8B090
- PT\_TAB3RIXB8B090
- PT\_TAB3MIXB8B090
- PT\_TAB3BAIB8B090FFFFFF
- PT\_TAB3BIAB8B090FFFFFF
- PT\_TAB3BIIB8B090

If you are changing only the inactive color, which is beige, you need only the files that contain the beige color code, B8B090. Similarly, if you are changing only the active tab color, which is white, you need only the files that include FFFFFFFF. However, for the changes to apply, you must change all those files with that color code. If one file is missed or there is a typographical error in the file name, the default style still applies.

2. To export the file to a directory, select File, Export Image.
3. In the Save Image dialog box, select the directory in which you want to place the image.
4. Use any image editor to alter the image colors.

---

**Note.** Do not change the size or shape of the images, and do *not* replace any of the transparent pixels with solid pixels.

---

5. In PeopleSoft Application Designer, add the new image definitions by selecting File, New, Image.

Save the new image files under the appropriate name. The new images must have new names reflecting the HTML colors that they represent. For example, replace FFFFFFFF with the new active tab color, and replace B8B090 with the new inactive tab color. Be careful when naming these images. If the images with the correct colors for the tab style are not in the database, the system automatically applies the default images.

6. Open the appropriate style sheet.

If you are changing the tab folder colors, use PSSTYLEDEF. If you are changing grid tabs, use PSSTYLEDEF or the style sheet that is associated with the page definition in which the grid is located.

---

**Note.** The grid tab styles can be set for each grid individually using the grid properties. However, for page tabs, you can set the styles using only the default styles in the application style sheet.

---

You must access PSACTIVETAB, PSINACTIVETAB, and any other tab classes that your application uses (for example, PSLEVEL1GRIDACTIVETAB).

7. For folder tabs, open the PSACTIVETAB class by double-clicking the class name.
8. On the Background tab, select the color of the background of the active tab (matching the color of the tab image that you created).  
  
Use the drop-down list box to find the color or browse through the color choices.
9. Do the same for the PSINACTIVETAB class (or any others that you need) for folder tabs by double-clicking the class name and selecting the Background tab.
10. For grid tabs, access the style class that is associated with the grid tab.
11. Change the colors as indicated in the previous steps.
12. Save the style sheet.

## Examples

This is an example of folder tabs with a green background. In addition, the color of the text on the tabs was changed to black.



### Green tabs

In the example, the active tab color stayed the same (FFFFFFF). Only the inactive tab color changed. Therefore, only these files were created and added to the list of images:

- PT\_TAB2LIX80FF80
- PT\_TAB2RIX80FF80
- PT\_TAB2BAI80FF80FFFFFF
- PT\_TAB2BIA80FF80FFFFFF

- PT\_TAB2BII80FF80
- PT\_TAB3LIX80FF80
- PT\_TAB3RIX80FF80
- PT\_TAB3MIX80FF80
- PT\_TAB3BAI80FF80FFFFFF
- PT\_TAB3BIA80FF80FFFFFF
- PT\_TAB3BII80FF80



## Appendix E

# Performing Bulk Operations

This appendix provides an overview of bulk operations and discusses how to perform bulk:

- Insertions
- Modifications
- Deletions

---

## Understanding Bulk Operations

PeopleTools enables administrators to perform bulk operations on fields that might be in many different records or pages.

---

**Note.** For security reasons, only those users logged in to PeopleSoft Application Designer under the "PeopleSoft Administrator" role have access to the Bulk Operations menu items. For all other users these menu options are disabled. To gain access, navigate to PeopleTools, Security, User Profiles, User Profiles, Roles tab.

---

Bulk operations are useful for financial ChartFields. *ChartField* is an accounting term that describes a small group of character fields, typically 5 to 20, that are present in many records and pages. ChartFields can be between 1 and 30 characters long, with an average length of 10. They often identify organization-wide attributes, such as DEPTID (department identification).

---

**Important!** Many PeopleSoft financial applications have created a standardized way to modify ChartFields by using standard pages and their corresponding PeopleCode application programming interfaces (APIs). The financials applications do more than just invoke the bulk operations to modify ChartFields. For this reason, users of these applications should follow the applications guidelines rather than using these bulk operations menu items in PeopleSoft Application Designer. Consult your application to ensure that you are using the appropriate mechanism to update your ChartFields, before using PeopleSoft Application Designer's Bulk Operations tool.

---

---

## Performing Bulk Insertions

To perform bulk insertions by using PeopleSoft Application Designer, select Tools, Bulk Operations, Insert.

This section lists common elements and discusses how to insert:

- Fields into records.

- Record fields onto pages.
- Fields into records and onto pages.

---

**Note.** If you try to insert a field that is already present in a record or on a page, no processing takes place.

---

## Common Elements Used in This Section

<b>Model Field</b>	Specify a field with record attributes that you want to copy to all inserted fields.
<b>Clone PeopleCode of model field</b>	Select if you want PeopleCode in the record fields to be copied and references modified to the inserted field.

## Inserting Fields into Records

You can insert a source field into selected records only if the model field exists in those records. The source field is assigned the same record field properties as the model field in each record and is inserted directly after the model field.

If the model field has a prompt table, a prompt table is created for the source field by using the name of the source field with *TBL* appended to it.

If the record is either a structured query language (SQL) view or a dynamic view type, the associated SQL is modified by expanding the SELECT clause to include the new field.

If the record is a subrecord, the parent records of the SQL view or dynamic view type that contain this subrecord are updated. If the SQL contains the model field in the WHERE clause or if the SQL is complex, the associated record is inserted into a project called BLK\_<SOURCEFIELDNAME> and displays a status message. Examine the contents of that project after the operation completes to verify successful completion.

If the model field has PeopleCode associated with it in the record or in a component, this PeopleCode is copied to the new field with all of the references to the model field changed to refer to the new field.

---

**Note.** Because performing this operation changes records that are used to build application tables, you must rebuild (alter) the database before these changes can be used.

---

## Inserting Record Fields onto Pages

This operation inserts the specified record field onto each page that is listed if the model field exists on the page.

If the model field is in a grid, the system inserts the new field into the grid next to the model field and assigns it the same page field properties.

If the model field is not in a grid, the system inserts the new field to the right of the model field (in the first open space) and assigns it the same page field properties. If the system detects a questionable field position, it inserts the page into a project called BLK\_<SOURCEFIELDNAME> and displays a status message. Examine that project after the operation completes to verify successful completion.

The page field name property is not cloned if it exists on the model field. Instead, the name of the new field is used, because the page field name should be a unique identifier for page elements.

---

**Note.** You do not need to rebuild the database after performing this operation. The changes take affect when you open the page.

---

**Record Name** Enter the record name containing the field to insert onto the pages.

**Get Pages from Project** Specify the project name containing the pages to modify.

**Get Records from Project** Specify the project containing the records to modify.

## Inserting Fields into Both Records and onto Pages

This operation is a combination of inserting fields into records and inserting record fields onto pages. The system examines each record to see if a model field exists. If so, it inserts the source field after the model field, including all field properties, PeopleCode, and SQL additions to itself and parent records. Then, for each record, the system checks all pages to see if the new record field should be inserted.

If the system detects a questionable field position, it inserts the page into a project called BLK\_<SOURCEFIELDNAME> and displays a status message. Examine that project after the operation completes.

**Get Records and Pages from Project** Specify the project containing the records and pages to modify.

---

## Performing Bulk Modifications

To perform bulk modifications by using PeopleSoft Application Designer, select Tools, Bulk Operations, Modify.

This section discusses how to:

- Modify record fields.
- Modify page fields.
- Reorder or resize grid fields.
- Change page field attributes.

## Modifying Record Fields

Use this operation to modify record fields by applying record field properties to a field in a set of records. All of the properties that are associated with the page are set for the record field in the listed records. Therefore, after this operation completes, the source field in these records contains identical record field properties.

---

**Note.** In most cases, you must rebuild the database for these attributes to take effect.

---

- Field will be modified on records from Project**
- Specify the project name containing the record fields to modify.
- Records**
- Specify the records containing the fields with attributes that you want to modify.
- Change Attributes**
- Click to open the menu to select specific attributes. Enter the appropriate attributes for the modified fields.

Modifying Page Fields

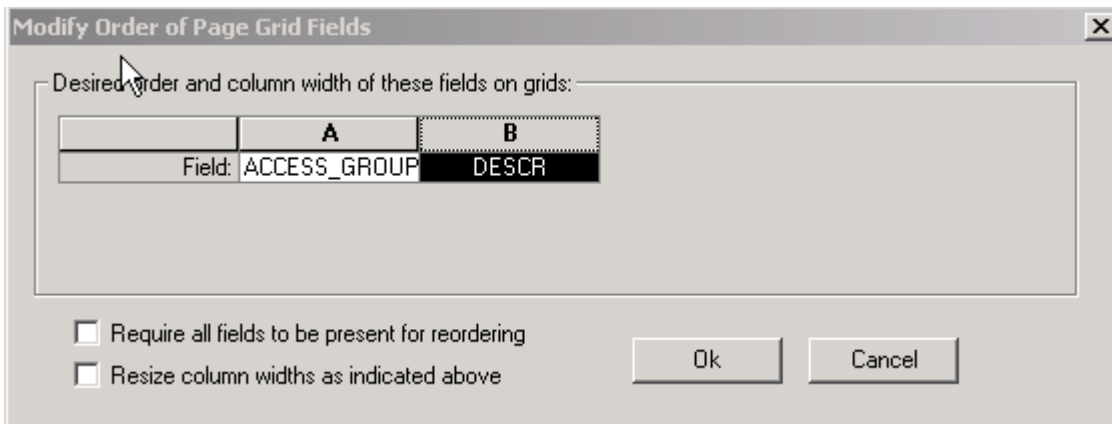
Use this operation to modify page fields in a variety of ways. You can specify a source field list from a list of fields, records, pages, or projects. You can only modify page field attributes on a single field at a time; therefore, if you do not specify a single field, the Change Attributes button is unavailable.

- Field,From Record, From Page, andFrom Project**
- Specify the source that is used to populate the Fields to be modified list.
- Get Pages from Project**
- Specify the project that is used to populate the Pages to be modified list.
- Reorder/Resize Grid Fields**
- Click to access the Modify Order of Page Grid Fields dialog box and specify the order and width of the Fields to be modified fields on all listed pages.
- Change Attributes**
- Click to access the Modify Page Field Attribute dialog box and specify the attributes of the field on all listed pages.

Reordering or Resizing Grid Fields

You can rearrange columns as needed, and the order is applied on target pages. The reordering algorithm groups the fields together at the first instance of any of the fields in a target page grid and forces the remaining fields into the specified order.

Pages that have these fields outside of grids are not affected by these changes. The resize operation applies only to grid column widths.



Modify Order of Page Grid Fields dialog box

**Require all fields to be present for reordering** Select if you do not want to modify pages that do not have all of the fields in the grid.

**Resize column widths as indicated above** Select to resize the columns to the column widths that are defined.

---

**Note.** These changes take effect immediately; no database build is required.

---

## Changing Page Field Attributes

Like record field attributes, you can modify page field attributes for the selected field (on the selected set of pages) by using the Modify Page Field Attribute dialog box. All of the attributes on the screen are applied to the page field on all pages.

---

## Performing Bulk Deletions

To perform bulk deletions by using PeopleSoft Application Designer, select Tools, Bulk Operations, Delete.

Use the delete operation to:

- Remove the field from a list of pages, regardless of where the field exists on the page or whether it is in a grid.
- Remove the field from the list of records.

This removes associated PeopleCode and modifies associated SQL either in the record or, if the record is a subrecord, in parent records.

If the field is in the SELECT clause of the SQL, the removal is straightforward. However, if the field is also used in a WHERE clause, or if the field is the only item in the SELECT clause, the record is not modified. Instead, it is inserted into a project called BLK\_<SOURCEFIELDNAME> as noted in messages in the status screen.

---

**Note.** A project with a targeted field in SQL statements should be examined by hand to delete the field as required from the associated SQL.

---

Deleting fields from records and pages does *not* remove the field definition itself and it does not remove the field from other applications, such as Crystal Reports.

---

**Note.** Because performing this operation changes records, you must subsequently rebuild the database, because each application data table is based on the record definition.

---

## Appendix F

# Running PeopleTools Cross-Reference Reports

This chapter provides an overview of cross-reference reports and discusses how to run a sample Crystal report.

---

## Understanding Cross-Reference Reports

By using PeopleTools, you can create new applications by defining menus and pages that you use to enter data and database tables. PeopleTools includes a variety of cross-reference reports (similar to blueprints) that guide you through the definitions (such as menus, pages, and records) that are delivered with the system and those that you create or adapt with PeopleTools.

PeopleTools cross-reference reports are predefined Crystal reports, not unlike the standard reports that are delivered with the PeopleSoft application. Like other standard reports, you can identify cross-reference reports by the three-character prefix *XRF*. The reports provide several views of the application, ranging from high-level lists of the windows, menus, and pages to the detailed database attributes of fields.

You should already be familiar with the types of definitions that are delivered with the PeopleSoft application, as well as with PeopleTools and relational databases.

The cross-reference reports include:

<i>Report</i>	<i>ID</i>	<i>Description</i>
Fields and Pages	XRFFLPN	Lists all fields in alphabetical order. It includes the names of all record and page definitions in which the fields occur and lists the field long names.
Fields and Records	XRFRCL	Lists all fields in alphabetical order by the associated record definition name. It details the field long name, type, length, and formatting.

<b>Report</b>	<b>ID</b>	<b>Description</b>
Records and Fields	XRFFLRC	Lists all fields in alphabetical order. It includes the field long name, type, length, and formatting, as well as the names of all record definitions that contain the field.
Applications and Fields	XRFAPFL	Lists all menus, such as general tables, in alphabetical order, including the fields in each menu. For each field, it lists the field name, type, length, and format, as well as all record and page definitions that contain the field (in the window).
Fields Referenced by PeopleCode Programs	XRFFLPC	Lists all PeopleCode programs in alphabetical order by the associated record definition and field. It includes the types of fields that are referenced in the PeopleCode program.
Field Listing	XRFIELDS	Lists all fields in alphabetical order. It includes field type, length, format, long name, and short name.
Menu Listing	XRFMENU	Lists all menus in alphabetical order, including all page definitions in each menu. It includes the associated search record definition name and detail page definition name.
Page Listing	XRFPAGE	Lists all page definitions in alphabetical order.
PeopleCode Programs and Field References	XRFPCFL	Lists record definitions that contain fields with PeopleCode program attributes. It includes the field name, as well as the associated record definitions and fields that are referenced in the PeopleCode program.

<b>Report</b>	<b>ID</b>	<b>Description</b>
Pages with PeopleCode	XRFPNPC	Lists all pages that contain fields with PeopleCode attributes. For each page, it includes the name of the record definitions that contain the field, as well as the field name and type
Records and Pages	XRFRCPN	Lists all record definitions in alphabetical order. It includes the menu and page definitions that are associated with each record definition.
Window Listing	XRFWIN	Lists all application menu windows in alphabetical order.

**Note.** In addition to standard cross-reference reports, if you are familiar with the PeopleSoft database, you can generate additional ad hoc reports to extract the exact combination of information that you need.

## Running a Crystal Report

Run cross-reference reports on the Process Scheduler Request page.

The following example explains how to generate the Field Cross Reference (XRFIELDSD) report.

### Process Scheduler Request

<b>User ID:</b>	QEDMO	<b>Run Control ID:</b>	CrystalClient
<b>Server Name:</b>	<input type="text" value="PSNT"/>	<b>Run Date:</b>	<input type="text" value="12/03/2001"/>
<b>Recurrence:</b>	<input type="text"/>	<b>Run Time:</b>	<input type="text" value="11:31:33AM"/>
<b>Time Zone:</b>	<input type="text"/>	<input type="button" value="Reset to Current Date/Time"/>	

Select	Description	Process Name	Process Type	*Type	*Format	
<input type="checkbox"/>	<a href="#">COBOL Multi-process Job</a>	3CBL	PSJob	(None)	(None)	
<input type="checkbox"/>	<a href="#">Crystal Multi-process Job</a>	3CRYSTAL	PSJob	(None)	(None)	
<input type="checkbox"/>	<a href="#">SQR Multi-process Job</a>	3SQR	PSJob	(None)	(None)	
<input type="checkbox"/>	Simple AE test program	AEMINTEST	Application Engine	Web	TXT	
<input type="checkbox"/>	<a href="#">All Process Types</a>	ALLTYPES	PSJob	(None)	(None)	

Process Scheduler Request page

To create a Field Cross Reference report:

1. Select PeopleTools, Process Scheduler , System Process Requests.

The Sample Processes search page appears.

2. Click Add a New Value.
3. Enter *CrystalClient* for the run control ID.
4. Click Add.
5. Click Run.

The Process Scheduler Request page appears.

6. Select the *Field Cross Reference*, *XRFIELDS*, and *Crystal* by clicking the Select check box next to each of these values in the Process List grid.
7. Click OK to generate a formatted display of this Crystal report in HTML.

## Appendix G

# Working With Records Used in the Mobile Synchronization Framework

This appendix provides an overview of records used in the Mobile Synchronization Framework and discusses how to:

- Configure records for the Mobile Synchronization Framework.
- Example: Build records for the Mobile Synchronization Framework.

---

**Important!** PeopleSoft Mobile Agent is a deprecated product. These features exist for backward compatibility only.

---

---

## Understanding Mobile Synchronization Framework Records

Because of the nature of the architecture of the Mobile Synchronization Framework, there are additional settings that need to be considered when building applications that you intend to run on mobile or disconnected computers. These settings address the issue of the application database running on the main network being updated while mobile users have their computers in a disconnected state.

With the Mobile Synchronization Framework, the mobile user has a copy of the application tables running on their computers. If a system administrator, for example, adds a column to a table used in the mobile application, the copy of this same table running on the disconnected computer is no longer current. When the mobile users connect to the main network and attempt to synchronize the data on their machines with the data in the main application database, the synchronization operation will fail if the new column is a required (or 'NOT NULL') field, because the application database expects a value to be inserted within the new column.

To ensure that the synchronization operation completes successfully, the system allows default values to be inserted in the new column.

---

**Note.** Default values can only be specified in fields of the following types: Character, Number, and Signed Number.

---

---

## Configuring Records for the Mobile Synchronization Framework

This section discusses how to:

- Set record properties.

- Set record field properties.

## Setting Record Properties

On the Use tab of the Record Properties dialog there is a group box named Sync type (MSF) devoted entirely to records used in mobile applications. The options in the Sync type (MSF) group box enable you to identify records as being used for mobile applications by selecting either of the options.

**Server —> User (Down Sync)** Select this option if the data existing on the mobile computer needs to be updated with the data existing on the main application database server. Selecting this option does not affect any behavior of the build process, such as inserting default values for required fields.

**User —> Server (Up Sync)** Select this option if the data existing on the main application database server needs to be updated by the data existing on the mobile computer.

If this option is selected, and a Character, Number, or Signed Number column is set as required, during the build process the system automatically inserts the default values for these fields.

Also, if this option is selected, the system does not allow you to set the following field types as required fields:

- Long Character
- Date
- Time
- Datetime

---

**Note.** You can select both options, depending on the application requirements.

---

### See Also

[Chapter 6, "Creating Record Definitions," Setting Record Properties, page 123](#)

## Working with Required Fields and Default Values

This section discusses how to:

- Create a required field.
- Specify default values for required fields.

### *Creating a Required Field*

As with any record definition, you specify that a field is required on the Edits tab of the Record Field Properties. Selecting the Required check box specifies that a particular field is 'NOT NULL'.

This is also the case with records used in mobile applications, except for records that have the User —> Server (Up Sync) option set. If set, only Character, Number, and Signed Number fields can be set to required. For all other field types, the system disables the Required check box.

### ***Specifying Default Values for Required Fields***

If the User —> Server (Up Sync) option is set in the record properties, *and* the Required option is set in the record field properties, during the build process the system inserts the following default values.

<b>Field Type</b>	<b>Default Value</b>
Character	'<space>'
Number	0 (zero)
Signed Number	0 (zero)

However, you can specify customer default values on the Use tab of the Record Field Properties dialog box. As with all other fields you use the Default Value group box to specify custom options.

---

## **Example: Building Records for the Mobile Synchronization Framework**

This example illustrates the SQL generated for the build scripts for records that meet the following criteria: used for mobile applications that

- Used in mobile applications.
- Have the User —> Server (Up Sync) record property set.
- Contain Long, Numbered, or Signed Number fields with the record field property of Required set.

The following SQL examples create a table with four NOT NULL fields that exist in the mobile record PS\_MSF\_DEMO. Two are Character fields (CUST\_ID and AA\_STATUS), one is a Number field (STD\_HOURS), and one is a Signed Number field (QTY\_ORDERED).

This sample SQL contains no custom defaults that are specified on the Edits tab on the Record Field Properties dialog box.

```
CREATE TABLE PS_MSF_DEMO_1 (CUST_ID NCHAR(15) DEFAULT N' ' NOT NULL,
  STD_HOURS DECIMAL(6, 2) DEFAULT 0 NOT NULL,
  QTY_ORDERED DECIMAL(14, 4) DEFAULT 0 NOT NULL,
  ACTION_DT PSDATE NULL,
  PM_VALID_TO_TIME PSTIME NULL,
  APPROVEDTM PSDATETIME NULL,
  AA_STATUS NCHAR(1) DEFAULT N' ' NOT NULL,
  QE_TEST_DESCR NTEXT NULL)
```

This sample SQL contains custom defaults that are specified on the Edits tab on the Record Field Properties dialog box. For STD\_HOURS the default is 8, for QTY\_ORDERED the default is 1, and for AA\_STATUS the default is A.

```
CREATE TABLE PS_MSF_DEMO_1 (CUST_ID NCHAR(15) DEFAULT N' ' NOT NULL,  
    STD_HOURS DECIMAL(6, 2) DEFAULT 8 NOT NULL,  
    QTY_ORDERED DECIMAL(14, 4) DEFAULT 1 NOT NULL,  
    ACTION_DT PSDATE NULL,  
    PM_VALID_TO_TIME PSTIME NULL,  
    APPROVEDTTM PSDATETIME NULL,  
    AA_STATUS NCHAR(1) DEFAULT N'A' NOT NULL,  
    QE_TEST_DESCR NTEXT NULL)
```

## Appendix H

# Creating Custom Plug-in Files for the Rich Text Editor

Plug-ins enable you to add custom buttons to the rich text editor toolbar.

This appendix discusses how to:

- Create custom buttons.
- Use images with plug-ins.
- Add multilingual support.
- Add the plug-in to the configuration file.
- Specify the Plugin File Id value.

---

## Creating Custom Buttons

To create custom buttons:

1. Create a new HTML definition and save it with the prefix *PT\_RTE\_PLUGIN\_*, for example, *PT\_RTE\_PLUGIN\_MYPLUGIN*.

2. Use JavaScript to code the functionality of the button in the HTML definition. This is an sample code for a button that will create a new page:

```
CKEDITOR.plugins.add( 'TestPlugin',
{
  init : function( editor )
  {

    editor.addCommand( 'Oracle_New_Page',
    {
      modes : { wysiwyg:1, source:1 },

      exec : function( editor )
      {
        editor.setData( editor.config.newpage_html );
        editor.focus();
      }
    });

    var iconpath= CKEDITOR.getUrl('skins/' + editor.config.skin + '/icons.png' );
    editor.ui.addButton( 'TestPlugin',
    {
      label : ''Oracle_New_Page'',
      command : ''Oracle_New_Page'',
      icon: iconpath,
      iconOffset : 38
    });
  }
});

CKEDITOR.config.newpage_html = '';
```

---

## Using Images with Plug-Ins

To use images with plug-ins:

1. Add the images to the database by creating new image definitions in Application Designer.
2. Create an iScript method to download the images you need for the plug-in. This is an example:

```
Function IScript_ GetImages
  &ImageURL = %Response.GetImageURL( Image.XYZ );
  %Response.Write(&ImageURL);
End-Function;
```

3. Send an AJAX request from the plugin to the iScript method.

The response is the image URL.

4. Use the image URL to display the image for the plug-in.

---

## Adding Multilingual Support

To add multilingual support to the button:

1. Add any Message Catalog entries necessary for multilingual support.

2. Create an iScript method to get the language-specific information from the Message Catalog. This is an example:

```
Function iScript_ GetData
  %Response.Write (MsgGetText(1234, 1, "Message not found.));
End-Function;
```

3. Send an AJAX request from the plugin to the iScript method.

The response is the language-specific information.

4. Use the data as the new language information for the button.

---

## Adding Plug-Ins to the Configuration File

To add plug-ins to the configuration file:

1. Add the new toolbar button to the toolbar set in the configuration file. This is an example of the code where the new button is *Oracle\_New\_Page*:

```
CKEDITOR.config.toolbar =
[
  ['Preview','Print','-','Cut','Copy','Paste','-','Undo','Redo','->
  ','Find','Replace', 'Oracle_New_Page']
];
```

2. Add the plug-in that hosts the new button. This is an example of the code:

```
CKEDITOR.plugins.addExternal( 'Oracle_New_Page', CKEDITOR.config.PluginPath );
CKEDITOR.config.plugins += ', 'Oracle_New_Page';

CKEDITOR.config.toolbar =
[
  ['Preview','Print','-','Cut','Copy','Paste','-','Undo','Redo','->
  ','Find','Replace', 'Oracle_New_Page']
];
```

---

## Specifying the Plugin File ID Value

To specify the plugin File Id value:

1. In Application Designer, access the properties for the rich text-enabled long edit box.
2. Click the Options tab.
3. Select the name of the plug-in file from the Plugin File Id drop-down list box.
4. Save the page definition.



# Index

## Numerics/Symbols

%EditTable function 195  
%Market system variable 370

## A

accessibility  
    deactivating image labels 201  
    designing accessible pages 197  
    labelling buttons/links 201  
    labelling grids/scroll areas 201  
    labels, using *See Also* labels  
    ordering page tabs 201  
    understanding 197  
    using colors 200, 202  
    using deferred mode 201  
    using instructional text 202, 205  
    using status icons 200  
access keys, setting 368  
access profiles  
    locking/unlocking definitions 522  
    understanding 42  
    using deferred mode 382  
activities, workflow 26  
Add Translate Table Value dialog box 91  
alignment  
    controlling via style classes 417  
    setting button label 252  
    setting in style sheets 420  
    setting page control 302, 314, 340  
    setting page label 319  
    setting static text 220  
alternate search keys  
    setting for records 132  
    understanding 130  
    using in indexes 168  
Alter process  
    confirming table alterations 183  
    creating/running alter scripts 176  
    creating temporary tables 183  
    determining tables to alter after field changes 179  
    forcing without table changes 182  
    handling data conversions 174  
    performing Alter in Place operations 182  
    renaming table fields 79  
    running 179  
    setting alter options 173, 180  
    setting column options 181  
    setting custom options 181  
    understanding 173  
    when to run 179  
analytic model 26  
application classes 26  
application data tables  
    creating/understanding 95  
    synchronizing with records 156  
    using the Alter process 179

Application Designer  
    configuring 38  
    hiding the interface 13, 153  
    implementing 3  
    setting undo/redo layers 22  
    setting Word Wrap for editors 22  
    starting 11, 498  
    understanding 3, 9  
    window  
        *See Also* Application Designer window  
Application Designer window  
    definition workspace definition workspace  
    menus 14  
    navigating 14  
    output window *See Also* output window  
    project workspace project workspace  
    using the components 14  
    using the toolbar 19  
application development  
    eight steps 10  
    planning records, control tables, and tablesets 95  
Application Engine programs  
    import *See Also* import program  
    understanding 26  
    using temporary tables 119  
application packages  
    accessing associated PeopleCode programs 25  
    Application Packages Editor 25  
    understanding 26  
Applications and Fields report 586  
approval rule sets 26  
auditing  
    creating audit records 126  
    naming audit records 118, 124  
    setting field audit options 133  
    using record-level audits 124

## B

backgrounds  
    changing colors for classic tabs 573  
    changing colors for tabs 574  
    colors, using 202  
    pages, overriding style sheets for 353  
    pages, selecting backgrounds for 239  
    referencing image URLs 425  
    selecting for edit boxes 314  
    selecting for grids 280  
    style classes, setting attributes for 423, 424  
    style classes, using 417  
    style classes, using default 553  
Body Area Row Action Button Properties dialog box 281  
borders  
    displaying for grids 280  
    hiding for frames 213  
    setting for group boxes 214  
    setting for page reports 355

- setting for record definition reports 143
    - style classes, setting attributes for 417, 426
    - style classes, using default 553
  - browsers
    - adjusting time fields for time zones 70, 99
    - changing default 345
    - configuring options 46
    - defining tableset controls 110
    - displaying fonts 419
    - editing/viewing DDL 188
    - HTML tables and performance 213, 342
    - indicating link status to users 430
    - opening for component search pages 377
    - ordering tabs 337
    - overriding date field formats 69
    - setting edit box colors 314
    - setting the base URI 22
    - sizing page controls 309, 314
    - spacing page controls 340
    - supporting buttons 253
    - supporting calendar prompts 297
    - supporting images 451, 454
    - supporting style class attributes 417
    - using mouse-over text 198, 199
    - using WBMP images 450
    - View in Browser function
      - See Also* View in Browser function
    - viewing HTML for pages 347
    - viewing pages 343
    - wrapping long words 324
  - buffers
    - adding/deleting rows 196, 286
    - controlling scrollable data 264
    - displaying the buffer size for file layouts 479
    - setting undo/redo layers 22
    - understanding component buffers 362
  - Build dialog box 149, 151
  - Build menu 17, 149
  - Build process
    - Alter process *See Also* Alter process
    - building script files 150
    - Build Settings dialog box
      - See Also* Build Settings dialog box
    - configuring 152
    - creating triggers 183
    - errors, navigating to 162
    - errors for project builds 153
    - granting build authority 148
    - indexes, creating 155, 167, 169
    - indexes, setting create options for 159
    - logs, setting options for build 159
    - logs (sample), output build 160
    - monitoring 150
    - running 151
    - selecting the scope 151
    - setting options 152
    - setting parameters 17
    - setting script options 162
    - table builds, confirming 167
    - tables, building SQL 11
    - tables, creating 155, 156
    - tables, setting create options for 157
    - understanding 48, 147
    - using the Build menu 149
    - viewing messages 21
    - views, creating 184
    - views, setting create options for 158
    - views, setting the build order for dynamic 145
  - Build Settings dialog box
    - opening 152
    - setting alter options 180
    - setting create options 157
    - setting logging options 159
    - setting script options 162
    - understanding 149
  - bulk operations
    - deletions 583
    - insertions 579
    - modifications 581
    - performing in PeopleSoft financial applications 579
    - understanding 579
  - Business Interlink 26
  - business processes 27
  - buttons
    - activating via the Enter key 245
    - adding for grids 249
    - adding for scroll areas 249
    - adding for scroll bars 249
    - adding instant messaging 248
    - adding scroll action buttons 264
    - associating with PeopleCode commands 248
    - calendar prompt *See Also* calendar prompts
    - enabling for display-only access 245
    - hover text, setting 256
    - HTML entries for 346
    - inserting 244, 302
    - jumping to page anchors 248
    - labels, applying 201
    - labels, setting images for 253
    - labels, setting text for 251
    - prompt/lookup *See Also* prompt buttons
    - running processes 248
    - secondary pages, accessing 249
    - secondary pages, associating primary pages with 236
    - setting as external/internal links 247
    - setting destination types 245
    - toolbar *See Also* toolbars
    - understanding 211, 244
    - using default style classes 553
- ## C
- calendar prompts
    - button image for 459
    - understanding 297
  - CDATA 224
  - Change Control
    - deleting page definitions 529
  - Change Control Change Control locking
    - understanding 517
  - Change Control feature
    - appointing an administrator 522
    - enabling/disabling 522
    - history *See Also* Change Control history
    - implementation considerations 521
    - locking *See Also* Change Control locking
    - reporting information 530
    - setting up 18, 522
    - stamping *See Also* Change Control stamping
    - supported definition types 535
    - understanding 517
    - understanding security 522

- using distributed development environments 524
  - using multiple databases for development 524
  - using projects 523
- Change Control history
  - enabling 523
  - implementation considerations 521
  - inserting/prompting automatically 519
  - inserting definition comments 527
  - opening entries 530
  - reporting 536
  - understanding 519
  - upgrading applications 520
  - using the PSCHGCTLHIST table 530
  - viewing 529
- Change Control locking
  - definitions, automatically locking 526
  - definitions, deleting 520
  - definitions, identifying locked 517
  - definitions, locking 525
  - definitions, locking all 523, 526
  - definitions, viewing locked 527
  - enabling 523
  - entering comments 519
  - implementation considerations 521
  - projects 518
  - understanding upgrades 518
  - understanding version control 518
  - using the PSCHGCTLLOCK table 530
- Change Control stamping
  - understanding 520
- change projects
  - file type codes, understanding 42
  - understanding 57
- Change Record Indexes dialog box 170
- change sets 523
- Change Space dialog box 187
- channels 27
- character fields
  - attributes, setting 65
  - attributes for the Alter process, setting 173
  - creating 63
  - Japanese 66
  - long *See Also* long character fields
  - understanding 63, 65
- characters
  - applying uppercase automatically 88
  - character fields *See Also* character fields
  - end of file (EOF) 493
  - interpreting backslashes as escape characters 480
  - long character fields
    - See Also* long character fields
  - naming definitions 31
  - naming records 117
  - replacing blank edit box spaces with fill characters 316
  - using lexical, literal and meta 80, 87
  - using Smart Punctuation 87
- ChartFields
  - setting styles 556
  - understanding 579
- charts
  - ChartFields *See Also* ChartFields
  - drawing chart placeholders 208
  - inserting 303
  - using 212
- check boxes
  - associating with records 220
  - enable when page is display only 324
  - HTML entries for 346
  - inserting 302
  - labels, enabling/disabling colons in 320
  - labels, using default 220
  - setting properties 220
  - understanding 209
  - using default style classes 553
  - using in deferred mode 195
- child tables 128
- Class dialog box
  - setting backgrounds 423
  - setting border attributes 426
  - setting fonts 418
  - setting margin attributes 427
  - setting miscellaneous attributes 429
  - setting spacing and alignment 420
- colors
  - background *See Also* backgrounds
  - changing colors for classic tabs 573, 574
  - changing colors for tabs 443, 574
  - modifying system colors 40
  - optimizing accessibility 200
  - selecting for edit boxes 314
  - setting for borders 426
  - setting for fonts 420
  - setting for grids 286
  - setting for page controls 314
  - style classes, setting attributes for 423
  - style classes, using 417
  - style classes, using default 553
  - using for accessibility 202
- columns, grid
  - aligning labels 319
  - deleting 270
  - enabling View All 288
  - freezing 273
  - inserting/manipulating 269
  - moving 270
  - ordering 267
  - setting heading properties 281
  - setting options in the Alter process 181
  - setting properties 271
  - showing when cells hidden 288
  - sizing 117, 271
  - sorting 117, 285
- command line
  - building projects 153
  - parameters
    - See Also* command line parameters
  - starting Application Designer 12
- command line parameters
  - building projects 500
  - comparing definitions 509
  - copying definitions 502
  - project builds 153
  - referencing selections 545
  - starting Application Designer 498
  - starting Data Mover 498
  - syntax 12, 497
  - understanding 497
- comments
  - enabling the prompt for definitions 528
  - including/excluding alter comments 163
  - inserting for definitions 527
- ComponentChanged function 245

- component definition window 363
  - component interfaces
    - deleting 33
    - understanding 27
  - Component Properties dialog box
    - opening 371
    - setting general properties 371
    - setting Internet properties 374
    - setting use properties 371
  - components
    - adding to a portal registry 399
    - adding to permission lists 399
    - assigning to portals 398
    - ComponentChanged function 245
    - Component Properties dialog box
      - See Also* Component Properties dialog box
    - customizing the display 378
    - defining 363, 364
    - defining pagebars 377
    - definitions, creating 11
    - definitions, opening 365
    - definitions, opening record 364
    - deleting 33
    - displaying/updating record rows 373
    - displaying toolbar actions 375
    - enabling the Expert Entry feature 377
    - enabling the Expert Entry feature via
      - PeopleTools Security 383
    - including the Refresh button 382
    - interfaces *See Also* component interfaces
    - links, displaying 376
    - links, selecting components for internal 250
    - menu navigation, including components for 373
    - menus, adding components to 389, 399
    - menus, creating custom 386
    - menus, finding components that reference 370
    - opening attached PeopleCode 364
    - ordering within folders 402
    - page attributes, setting 367
    - page definitions, opening 363
    - page items/attributes, viewing 363
    - page level information, viewing 364
    - pages, adding 365
    - pages, copying/moving between components 367
    - pages, enabling deferred mode for 368
    - pages, hiding 368
    - pages, naming 368
    - pages, navigating 362, 369, 376
    - pages, reordering 366
    - pages based on work records, adding 380
    - processing modes, setting 376
    - processing modes based on users, setting 382
    - registering 11, 398
    - registering in multiple portals 404
    - search page history, enabling check boxes 375
    - search page options, setting 375
    - search pages, using instructional text for 375
    - search records, overriding 381
    - search records, setting 372
    - setting access keys 368
    - setting market-specific 370
    - setting properties
      - See Also* Component Properties dialog box
    - setting the three-tier execution location 373
    - spell check, enabling mandatory 373
    - spell check errors, understanding 44
    - tabs, displaying 376
    - tabs, labelling 368
    - understanding 27, 361
    - understanding buffers 362
    - validating 21
    - viewing record/role information 364
    - viewing update history 371
  - Consolidate Images dialog box 455
  - content references
    - assigning names 398
    - assigning to portals 398
    - creating for components 391, 399
    - creating for pages 391
    - labelling 402
    - understanding 397
    - using default style classes 553
  - control tables
    - prompt tables *See Also* prompt tables
    - record groups record groups
    - sharing values 100, 101
    - tablesets *See Also* tablesets
    - understanding 100
  - Convert Images dialog box 454
  - Convert Scroll to Grid function 265
  - Create Table process 155
  - cross-reference reports *See* Crystal reports
  - Crystal reports
    - running 587
    - understanding 585
  - CSV files
    - converting tabs to spaces 480
    - field padding 486
    - importing fields 487
    - inserting segments/fields 495
    - interpreting backslashes as escape characters 480
    - interpreting backslash quotes (\") 481
    - interpreting double quotes (\" ") 481
    - parsing file fields 480
    - parsing support 471
    - previewing input data 488
    - selecting field qualifiers 486
    - trimming field spaces 486
    - troubleshooting input data 490
    - understanding 468, 470
  - currencies
    - displaying symbols 315
    - multicurrency fields 215, 220, 323
  - cursors
    - selecting page control groups 308
    - style classes, setting attributes for 430
    - style classes, using 417
    - using the page control toolbar 307
  - custom indexes
    - adding 171
    - understanding 168
- ## D
- database
    - validating for WSRP 17
  - databases
    - Change Control, using 524
    - Change Control stamping 520
    - command line parameters 497
    - comparing via command line parameters 509

- copying between databases via command line parameters 502
  - indexes *See Also* indexes
  - normalized relational 96
  - saving script files 164
  - selecting types 545
  - synchronizing definitions 48
  - tables for PeopleTools applications 95
  - triggers *See Also* triggers
  - understanding forms 97
  - using master development databases 524
  - viewing space allocation 18
- Data Definition Language (DDL) *See* DDL
- data interchange EDI
- Data Mover 498
- date fields
  - creating 63
  - enabling four-digit century entries 315
  - field length in file layouts 477
  - setting attributes 69
  - understanding 63, 69
- dates
  - date fields *See Also* date fields
  - DateTime fields DateTime fields
  - effective effective dates
  - selecting formats/separators for file fields 484
  - understanding reasonable 140
  - using the ReadRowset method 477
- DateTime fields
  - creating 63
  - displaying time zones 316
  - field length in file layouts 477
  - setting attributes 70
  - understanding 63, 70
- DB2 UDB
  - altering tables – impact on views 178
  - functional indexes 169, 541
  - maximum image field size 71
  - performing Alter in Place operations 182
  - saving script files 164
- DDL
  - administering 186
  - editing 172
  - understanding 148
  - understanding the Build process 147
  - using record DDL 186
  - viewing parameters for creating tables, indexes, and tablespaces 188
- DDL Model Defaults page 188
- Debug menu 17
- decimal precision
  - inserting decimals automatically for page controls 315, 317
  - meta symbols 81
  - selecting for file fields 485
  - setting for field definitions 114
  - setting for numeric fields 69
  - using CSV files 471
  - using FIXED files 477, 479
  - using XML files 473
- deferred mode
  - adding multiple rows 195
  - enabling for page controls 354
  - enabling for pages 354, 368
  - limiting refreshes 201
  - setting fields with edit tables 194
  - setting for components 376
  - setting for grids/scroll areas 285
  - triggering server processing 192
  - understanding 192
  - understanding performance issues 342
  - using check boxes and radio buttons 195
  - using drop-down list boxes 195
  - using error/warning messages 196
  - using on prompt tables 195
  - using PeopleCode 193
  - using related displays 195
  - using the Expert Entry feature 382
- deferred processing mode *See* deferred mode
- definition properties dialog box 29
- definitions
  - access profiles *See Also* access profiles
  - Change Control, definitions supported by 535
  - Change Control history, viewing 529
  - Change Control implementation considerations 521
  - Change Control stamping
    - See Also* Change Control stamping
  - closing 32
  - comments, enabling the prompt for 528
  - comments, entering 520
  - comments, inserting 527
  - component *See Also* components
  - constructing SQL 24
  - creating 30
  - definition workspace
    - See Also* definition workspace
  - deleting 33
  - drag-and-dropping 37
  - field *See Also* fields
  - field formats 40
  - file type code *See Also* file type codes
  - locked, viewing 527
  - locking 517, 525
  - locking all 523, 526
  - locking automatically 526
  - naming 31, 32
  - opening 15, 28
  - page *See Also* pages
  - PeopleCode, parent definitions for 22
  - PeopleCode, viewing 23
  - PeopleCode, viewing characteristics 16
  - PeopleCode definition types/IDs 534
  - PeopleTools definition types/IDs 532
  - performing bulk operations 18
  - projects, accessing definitions for 50
  - projects, inserting into 16, 53, 58
  - projects, modifying definitions for 52
  - projects, removing from 55
  - projects, using 49
  - properties, editing 15, 29
  - properties, viewing 29
  - record *See Also* records
  - references, finding
    - Find Definition References process
  - references, understanding 35
  - references, viewing 16
  - related definitions, finding
    - See Also* Find Definition References process
  - related definitions, understanding 35
  - reports for definitions, viewing 15
  - saving 31, 58
  - selecting types 546

- setting default owner IDs 22
- styles *See Also* styles
- synchronizing database definitions 48
- system colors *See Also* colors
- understanding the Build process 147
- understanding types 25
- upgrade, displaying definitions available for 51
- using definition keys 530
- viewing 15, 29
- definition workspace
  - cascading/closing/grouping/tiling definitions 19
  - opening definitions 28
  - understanding 20
  - using the multiple document interface (MDI) 36
- Delete Definition dialog box 33, 79
- development database 524
- Development view
  - identifying locked definitions 517
  - understanding 50
  - using 50
  - viewing upgrade-only definitions 55
- device independent bitmap (DIB) images 454
- DIB images 454
- DoModal function 211
- drag-and-drop
  - adding file fields to file layouts 476
  - adding file records to file layouts 475
  - copying/moving controls within/between pages 310
  - dragging field definitions from records to pages 305
  - dragging fields from the project workspace to a page 306
  - dragging pages into components 366
  - dragging record definitions onto pages 306
  - inserting columns into grids 269
  - inserting fields into records 122
  - moving fields between records 140
  - moving fields within records 140
  - moving grid columns 270
  - moving HTML areas 242
  - moving page control labels 310
  - ordering page controls via the order list 338
  - reordering component pages 366
  - using 37
- drop-down list boxes
  - default base styles 325
  - default style classes 553
  - HTML entries for 346
  - inserting 302
  - positioning labels 319
  - setting properties 221
  - understanding 210
  - using 296
  - using deferred mode 195
  - using display-only 325
  - using the Expert Entry feature 383
  - wrapping long words 324
- duplicate order keys
  - ordering in records 130
  - setting for records 131
  - using in records 129
- dynamic views
  - creating select statements 145
  - defining records as 120
  - naming records as 118

- setting the build order 145
- using cross-platform meta-SQL 185

## E

- EDI
  - applying file layouts to 468
  - EDI Manager *See Also* EDI Manager
  - performing 487
  - using segments 475
- EDI Manager
  - CSV file properties 470
  - FIXED file properties 469
- edit boxes
  - as mouse over popup fields 223
  - default base styles 325
  - default style classes 553
  - enabling 1000 separators 315
  - enabling spell check 316
  - HTML entries for 346
  - inserting 302
  - labels, applying 223
  - labels, enabling/disabling colons in 320
  - labels, positioning 319
  - long *See Also* long edit boxes
  - replacing blank spaces with fill characters 316
  - sizing 314
  - understanding 210
  - using display-only 224, 325
  - wrapping long words 324
- Edit Box Properties dialog box
  - setting general properties 327
  - setting label properties 317
  - setting record properties 312
  - setting use properties 321
- Edit menu 15, 300
- editors *See Also* SQL Editor
- Edits Display mode 115
- effective dates
  - level-based controls 294
  - translate table values, adding 92
  - translate table values, understanding 91
  - understanding 98
  - using drop-down list boxes 195
  - using in components 373
- electronic data interchange (EDI) *See* EDI
- end of file (EOF) characters 493
- Enter key, activating links/buttons via 245
- EOF characters 493
- errors
  - displaying in deferred mode 196
  - handling for pages/page-controls 354
  - handling for project builds 153
  - logging during alter processes 153
  - setting build log options 159
  - setting error message styles 556
  - spell check results 43
  - troubleshooting file layout input data 490
  - viewing for PeopleCode 21
- events
  - naming records with event fields 118
  - triggering server processing 192
  - viewing for field definitions 24, 116
- Excel, Microsoft *See* MS Excel
- Expert Entry feature
  - enabling 377, 382

- enabling via PeopleTools Security 383
- understanding 382
- understanding limitations 383
- using the Refresh button 382

## F

- Field Display mode 114

- field labels 310

- Field Listing report 586

- fields

- applying uppercase automatically 88
  - character *See Also* character fields
  - containing PeopleCode 116
  - date *See Also* date fields
  - datetime DateTime fields
  - definitions, changing 77
  - definitions, creating 10, 62
  - definitions, deleting 79
  - definitions, dragging 305
  - definitions, opening 75
  - definitions, printing 80
  - definitions, renaming 78
  - definitions, setting attributes for 63
  - definitions, understanding 61
  - definitions, viewing 76, 114
  - file fields *See Also* file fields
  - format families, changing 83
  - formats, adding display formats to stored 86
  - formats, adding stored 86
  - formats, deleting stored 86
  - formats, editing 85
  - formats, testing 88
  - formats, understanding 40
  - formats, updating display 86
  - formats and notation, understanding 80
  - image *See Also* image fields
  - ImageReference ImageReference fields
  - key key fields
  - label IDs, setting 133
  - labels, adding 73
  - labels, applying 199
  - labels, deleting 74
  - labels, selecting record field 74
  - labels, using multiple 73
  - long character *See Also* long character fields
  - numeric numeric fields
  - page controls, setting properties for 311
  - page field attributes, bulk changes for 583
  - page fields, bulk modification of 582
  - pages, bulk insertion into 580, 581
  - pages, deleting from 141
  - pages, dragging fields from the project workspace to 306
  - record field properties, understanding 127
  - record fields, bulk modification of 581
  - record field use properties, setting 130
  - records, adding SetIDs to 107
  - records, bulk insertion into 580, 581
  - records, deleting from 141
  - records, inserting into 122
  - records, moving between 140
  - records, moving within 140
  - records, understanding fields in 61, 121
  - records, using audit fields in 126
  - records, using search fields in 132

- refreshing 192
  - related display *See Also* related display
  - reordering 116
  - report, Applications and Fields 586
  - report, Field Listing 586
  - report, Fields and Pages 585
  - report, Fields and Records 141, 585
  - report, Fields Referenced by PeopleCode Programs 141, 586
  - report, PeopleCode Programs and Field References 586
  - report, Records and Fields 586
  - set control *See Also* set control fields
  - setting as required 138
  - setting audit options 133
  - setting default values 134
  - setting edit properties 137
  - setting related date fields 134
  - setting the maximum display length 85
  - setting time zone fields 134
  - setting type ahead 134
  - table data, converting 174
  - tables, enabling prompt 139
  - tables, enabling translate 139
  - tables to alter after field changes, determining 179
  - testing for reasonable dates 140
  - time *See Also* time fields
  - types, changing 77
  - types, understanding 63
  - understanding 27
  - understanding derived/work 298
  - understanding model fields 580
  - understanding spell check errors 45
  - using smart punctuation 87
  - viewing editing options 115
  - viewing key-related characteristics and default values 114

- Fields and Pages report 585

- Fields and Records report 141, 585

- Fields Referenced by PeopleCode Programs report 141, 586

- file

- length 40

- file fields

- adding to file layouts 474, 476, 495
  - converting text to uppercase 486
  - entering default values 485
  - entering descriptions 485
  - entering the decimal precision 485
  - including/excluding 487
  - interpreting backslashes as escape characters 480
  - naming 476, 485
  - padding 486
  - parsing 480
  - selecting data types 485
  - selecting date formats/separators 484
  - selecting delimiters 483
  - selecting inheritance 485
  - selecting qualifiers 483, 486
  - selecting XML tags 485
  - setting lengths 486
  - setting properties 483
  - spaces, converting tabs to 480
  - spaces, trimming 486
  - stripping characters 486
  - using read/write methods 477

- File Layout Definition Properties dialog box 478
- File Layout Field Properties dialog box 483
- file layouts
  - applying to data interchanges 468
  - creating definitions 473
  - creating definitions (sample) 492
  - exporting data 491
  - field properties, setting 483
  - fields, adding 474, 476, 495
  - fields, including/excluding 487
  - format, CSV *See Also* CSV files
  - format, FIXED FIXED files
  - format, XML XML files
  - formats, applying 477
  - formats, flat file 468
  - formats, viewing 479
  - import programs, generating/running 491
  - import programs, understanding 487
  - input data, previewing 488
  - input data, troubleshooting 490
  - naming 476
  - producing a sample 492
  - records, adding 474, 475
  - records, root-level 488
  - reordering nodes 476
  - segments *See Also* segments
  - setting properties 478, 494
  - understanding 27, 467
  - using read/write methods 477
- File Layout Segment Properties dialog box 481
- File Layout window 473
- File menu 14, 300
- file records
  - adding to file layouts 474, 475
  - assigning file record IDs 483
  - assigning identification sequence numbers 482
  - entering descriptions for 483
  - naming 476, 482
  - segment terminator 483
  - selecting delimiters 483
  - selecting qualifiers 483
  - selecting XML tags 483
  - setting maximum combined field lengths 483
  - setting properties 481
  - understanding hierarchy 488
  - using read/write methods 477
- file type codes
  - understanding 42
- filtering
  - setting upgrade view options 16
- Find Definition References process
  - changing field definitions 77
  - deleting definitions 33
  - deleting fields 141
  - deleting page controls 310
  - deleting pages 354
  - finding menus that reference components 370
  - running 15, 34
  - viewing referenced definitions 21
- FIXED files
  - importing fields 487
  - input data, previewing 488
  - input data, troubleshooting 490
  - inserting segments/fields 495
  - outputting numbers to 477
  - properties 469
  - trimming field spaces 486
  - understanding 468, 469
- flat files
  - understanding file layouts 467
  - understanding formats 468
- folders
  - content references 402
  - ordering components within 402
  - Project 50
  - tabs for *See Also* tabs
- fonts
  - setting attributes for specific languages 420
  - setting color, style, variant and weight 420
  - setting families 419
  - setting for grids 286
  - setting for page controls 314
  - setting in style sheets 418
  - sizing 419
  - sizing page controls 314
  - understanding generic/installed 417
  - using default style classes 553
  - using style class attributes 417
- footers
  - printing in reports 355
  - setting borders for ChartFields 556
  - setting footer areas for grids 281
  - setting for scroll areas 261
- Format Families dialog box 83
- Formats dialog box 85
- forms 97
- frames
  - grouping page controls 208
  - inserting 302
  - understanding performance issues 342
  - using 212
  - using default style classes 553
- free form sub style sheets
  - changing font and color settings 435
  - understanding 410
- functional indexes
  - DB2 UDB 541
  - Informix 541
  - MSSQL Server 539
  - platform support 169
- functions
  - %EditTable 195
  - Alter *See Also* Alter process
  - ComponentChanged 245
  - Convert Scroll to Grid 265
  - DoModal 211
  - Generate HTML
    - See Also* Generate HTML function
  - GetHTMLText 464
  - iScripts *See Also* iScripts
  - naming records containing 118
  - ScrollSelect 284
  - SetDBFieldNotUsed 114
  - View in Browser
    - See Also* View in Browser function

## G

- Generate HTML function
  - understanding 301
  - using 346
- GetHTMLText function 464
- GIF images
  - converting 454

- importing 452
- understanding 450
- Graphics Interchange Format (GIF)
  - See* GIF images
- grid
  - summary properties 282
- Grid Properties dialog box
  - setting body area properties 280
  - setting general properties 275
  - setting use properties 283
- grids
  - button images for 457
  - buttons, providing 249
  - collapsing to header bars 288
  - column heading properties, setting 281
  - columns, deleting 270
  - columns, freezing 273
  - columns, inserting/manipulating 269
  - columns, moving 270
  - columns, resizing 271
  - columns properties, setting 271
  - converting from scroll bars 265
  - customizing at runtime 267
  - default settings 266
  - displaying/hiding 302
  - dragging record definitions into 306
  - enabling/disabling personalizations 274
  - HTML tag 346
  - inserting 303
  - inserting/resizing 269
  - inserting page controls 266
  - inserting subpages 235
  - label properties, setting 275
  - labels, aligning column 319
  - labels, applying column 199
  - labels, applying title 201
  - nesting in scroll areas 293
  - ordering 338
  - ordering in bulk 582
  - processing in deferred mode 285
  - resizing in bulk 582
  - rows, controlling styles for 286
  - rows, inserting multiple 285
  - row selection indicators, setting 286
  - selecting pop-up menus 286
  - setting body area properties 280
  - setting display options 287
  - setting footer area properties 281
  - setting general properties 275
  - setting header area properties 277
  - setting scroll bar options 287
  - setting the View All feature 279
  - setting use properties 283
  - tabs, creating grids with 271
  - tabs, setting styles for 576
  - tab separators, inserting 302
  - understanding 211, 265
  - understanding performance issues 342
  - using default style classes 553
  - using multiple 273
  - using page layout grids 303
  - using the Page Control Inspector 310
- group boxes
  - associating with multicurrency processing 215
  - changing labels 214
  - collapsing 215
  - grouping page controls 208

- hiding fields 215
- inserting 213, 302
- moving 213
- setting properties 214, 215
- understanding performance issues 342
- using default style classes 553

## H

- Header Navigation Bar Properties dialog box 277
- headers

- collapsing grids to header bars 288
- printing for reports 355
- setting backgrounds for ChartFields 556
- setting for scroll areas 260
- setting header areas for grids 277

- Help menu 19

- history

- Change Control
  - See Also* Change Control history
- displaying/updating rows for component records 373
- effective dates for data rows 99
- enabling history check boxes on search pages 375

- homepages

- registering pagelets 402
- setting templates for homepage pagelets 402
- using default style classes 553

- horizontal rules

- drawing 208, 216
- inserting 302
- ordering tabs 217
- setting properties 216
- using default style classes 553

- hot keys

- disabling 375
- invoking spell check 316
- refreshing pages 192, 382
- saving pages 373
- viewing related displays 195

- hover text

- programmatically changing 256

- HTML

- converting page controls 346
- definitions, creating 463
- definitions, naming 464
- definitions, opening 464
- definitions, referencing 464
- definitions, understanding 463
- HTML areas *See Also* HTML areas
- using the Generate HTML function
  - Generate HTML function
- viewing for pages 347

- HTML Area Properties dialog box 242

- HTML areas

- changing labels 243
- generating trees in 240
- HTML definitions, referencing 464
- HTML definitions, understanding 463
- inserting 242, 302
- populating 241, 242
- understanding 211, 240

**I**

- image catalog
  - general images 458
  - look-up page images 458
  - portal images 461
  - query images 459
  - scroll/grid action buttons 457
  - search page images 458
  - tab images 458
  - trees images 460
  - understanding 450
  - using 457
- image fields
  - creating 63
  - setting attributes 70
  - understanding 63, 70, 449
- ImageReference fields
  - creating 63
  - setting attributes 71
  - understanding 63, 71, 449
- images
  - associating with records 218
  - background images, referencing URLs for 425
  - background images in style sheets, setting 423
  - catalog *See Also* image catalog
  - changing display sizes 454
  - classic tabs, naming images for 573
  - consolidating 455
  - creating alternate types 452
  - definitions, creating 451
  - definitions, opening 453
  - definitions, understanding 449, 450
  - definitions, updating 453
  - drawing image frames 208
  - formats, converting 454
  - formats, selecting 71, 450, 451, 452
  - formats, setting storage 22, 454
  - GIF *See Also* GIF images
  - image fields image fields
  - ImageReference fields ImageReference fields
  - importing 452
  - inserting 302
  - JPG *See Also* JPG images
  - labels, button/link 253
  - labels, deactivating 201
  - labels, functional image 198
  - labels properties, setting 219
  - mirroring 452
  - naming 457
  - printing a snapshot of your page design 355
  - setting properties 218, 219, 451
  - showing status via icons 200
  - static *See Also* static images
  - tabs, creating images for 574
  - understanding 449
  - using controls 449
  - using default style classes 553
  - using in workflow maps 452
  - WBMP *See Also* WBMP images
- import process *See* import program
- import program
  - generating/running 491
  - understanding 487
- indexes
  - custom, adding 171
  - custom, understanding 168

- editing DDL 172
- functional *See Also* functional indexes
- ordering keys 170
- search 169
- setting Alter process options 182
- specifying creation options 159
- supported platforms 169
- understanding 167
- understanding DDL 148
- using record DDL 186
- viewing DDL 188
- when to create (based on record changes) 169

**Informix**

- altering tables – impact on views 178
- converting table data 174
- functional indexes 169, 541
- maximum image field size 71
- performing Alter in Place operations 182
- saving script files 164

**inheritance**

- file fields 485

**Insert into Project dialog box 53****Insert menu 16, 301, 307, 365****Install process 3****instant messaging 248****interactive mode**

- setting for components 376
- triggering server processing 192
- understanding 191
- using the Expert Entry feature 382

**interfaces**

- component *See Also* component interfaces
- hiding the Application Designer 13, 153
- multiple document interface (MDI) 36

**International Organization for Standardization (ISO) 8601 standards 477****Internet**

- Internet scripts *See Also* iScripts
- setting component properties 374

**iScripts**

- naming records 118
- registering 398
- understanding 397

**ISO 8601 standards 477****J****Japanese characters 66****joins 185****Joint Photographic Experts Group (JPEG)**

- See* JPEG images

**JPG images**

- converting other formats to 454
- understanding 450

**JSR168STYLEDEF\_SWAN sub-style sheet 438****JSR168STYLEDEF sub-style sheet 437****K****key fields**

- descending keys 132
- effective dates and level-based controls 294
- identifying 131, 364
- ordering 130, 336

- setting search keys 132
- understanding 114, 130
- using in child/parent tables 128
- using multiple keys in records 128
- using records without keys 129
- key lists
  - understanding 294
  - using data from current page for searches 250
- keys
  - accessing pages 294
  - alternate search
    - See Also* alternate search keys
  - duplicate order duplicate order keys
  - fields key fields
  - indexes indexes
  - maximum number 169
  - normalized databases 97
  - ordering 130, 170
  - search records *See Also* search records
  - setting access keys 368
  - sharing tables 103
  - using definition keys 530
  - using drop-down list boxes 195
  - using for views 295
  - viewing key-related field characteristics 114

## L

- labels
  - buttons/links, applying image labels to 253
  - buttons/links, applying text labels to 251
  - buttons/links, labelling 201
  - check boxes, labelling 221
  - choosing text 204
  - component tabs, labelling 368
  - content references, labelling 402
  - drop-down lists, labelling 222
  - edit boxes, labelling 223
  - field label IDs, setting 133
  - fields, deleting labels for 74
  - fields, labelling 73
  - fields, labelling data-entry 199
  - functional images, labelling 198
  - grid columns, labelling 199
  - grids, labelling 201, 275
  - group boxes, changing for 214
  - horizontal rules, labelling 216
  - HTML areas, changing for 243
  - images, deactivating for 201
  - images, labelling 219
  - menus, labelling 387
  - page controls, aligning for 319
  - page controls, displaying text for 318, 319
  - page controls, inserting line feeds in labels for 319
  - page controls, labelling 317, 319, 320
  - page controls, labelling invisible 323
  - page controls, labelling related display 320
  - page controls, moving for 319
  - page controls, moving labels for 310
  - page controls, selecting label IDs for 319
  - page controls, using blanks for 318
  - pages, labelling 198
  - pages, positioning on 302
  - radio buttons, labelling 222, 223
  - record fields, labelling 74
  - scroll areas, labelling 201, 260
  - scroll bars, labelling 262
  - static text, labelling 220
  - subpages, labelling 235
  - understanding multiple 73
- languages
  - codes for 550
  - setting font attributes 420
  - setting related language records 124
  - support for spell check 316
  - translations, database 18
- Layout Grid dialog box 303
- Layout menu 301
- lexical characters 80, 87
- library, PeopleSoft Online 19
- line feeds (for page labels) 319
- links
  - activating via the Enter key 245
  - adding prompts 248
  - associating with PeopleCode commands 248
  - destination pages, selecting 250
  - destination pages, selecting action modes for 250
  - destinations, selecting components/menus for 250
  - destination types, setting 245
  - displaying for components 376
  - enabling for display-only access 245
  - external, setting as 247
  - external, setting URLs for 249
  - hover text, setting 256
  - HTML entries for 346
  - indicating link status to users 430
  - inserting 244, 302
  - internal, selecting nodes for 249
  - internal, selecting portals for 249
  - internal, setting as 247
  - labels, applying 201
  - labels, setting image 253
  - labels, setting text 251
  - page anchors, jumping to 248
  - page data for searches, using current 250
  - secondary pages, accessing 249
  - secondary pages, associating primary pages with 236
  - understanding 211, 244
  - using default style classes 553
  - using for toolbar actions 249
  - using pseudo classes 430
- list-item markers, setting styles for 429
- literal characters 80, 87
- logging
  - Build process 150
  - sample output for Build logs 160
  - setting build options 159, 160
  - setting levels 160
  - spell check results 43
- long character fields
  - converting table data 174
  - creating 63
  - positioning via builds 155
  - setting attributes 67
  - understanding 63, 67
- long edit boxes 223
  - as rich text 210
  - as rich text fields 224
  - configuring as rich text 328

- enabling 1000 separators 315
- enabling spell check 316
- expanding at runtime 323
- HTML entries for 346
- inserting 302
- labels, applying 223
- labels, enabling/disabling colons in 320
- labels, positioning 319
- replacing blank spaces with fill characters 316
- sizing 314
- understanding 210
- using default base styles 325
- using default style classes 553
- using display-only 224, 325
- wrapping long words 324
- Long Edit Box Properties dialog box
  - See* Edit Box Properties dialog box
- lookup buttons prompt buttons
- look-up pages 458

## M

- Maintain Record DDL dialog box 186
- margins
  - setting for page reports 356
  - setting for record definition reports 143
  - style classes, setting attributes for 417, 427
  - style classes, using default 553
- MDI 36
- menu import process 391
- Menu Listing report 380, 390, 586
- Menu Properties dialog box 388
- menus
  - Application Designer window 14
  - Build menu 17, 149
  - component menus, creating custom 386
  - components, adding 389, 399
  - components, finding menus that reference 370
  - components in navigation, including 373
  - Debug menu 17
  - defining pages 300
  - defining transfer 394
  - definitions, copying 389
  - definitions, printing 390
  - definitions, renaming 389
  - Edit menu 15
  - entering descriptions for 388
  - File menu 14
  - Help menu 19
  - importing custom menu groups into portals 391
  - Insert menu 16, 307
  - item properties, setting 387
  - items, deleting 390
  - item types, selecting 387
  - Menu Listing report 586
  - modifying 388
  - naming 387
  - pop-up *See Also* pop-up
  - selecting for link destinations 250
  - setting properties 388
  - setting up security 391
  - Tools menu 17
  - understanding 27, 385
  - using dynamic 37
  - viewing modification history 388
  - View menu 16, 39
  - Window menu 18, 36
- Merge Projects dialog box 55
- message catalog
  - accessing 381
  - setting button/link labels 252, 253
  - setting image labels 219
  - understanding spell check errors 43
  - using static text 209
- messaging
  - instant messaging 248
  - message catalog *See Also* message catalog
  - publishing messages in PeopleCode 194
  - setting error message styles 556
  - suppressing system message boxes 13
  - understanding message channels 27
  - understanding message definitions 27
- meta characters 80, 87
- meta-SQL 185
- methods
  - Alter *See Also* Alter process
  - ReadRowset 473, 477
  - WriteRecord 477, 483, 491
  - WriteRowset 477, 483, 491
- Microsoft Excel *See* MS Excel
- Microsoft SQL Server MSSQL Server
- mobile applications
  - setting system IDs 59
  - using record auto-update options 125
- mobile pages
  - registering 404
  - understanding 27
- modes
  - deferred *See Also* deferred mode
  - Edits Display 115
  - Field Display 114
  - interactive *See Also* interactive mode
  - PeopleCode Debugger PeopleCode Debugger
  - PeopleCode Display 116
  - processing modes *See Also* processing modes
  - selecting action modes for link destination pages 250
  - selecting action modes for search pages 375
  - selecting action modes for transfer menus 395
  - setting quiet mode 13
  - Use Display 114
  - View in Browser
    - See Also* View in Browser function
- Modify Order of Page Grid Fields dialog box 582
- MS Excel
  - downloading grid data 279
  - parsing file fields 480
  - parsing support for CSV files 472
- MSSQL Server
  - altering tables – impact on views 178
  - functional indexes 169, 539
  - maximum image field size 71
  - performing Alter in Place operations 182
  - saving script files 164
- multiple document interface (MDI) 36
- My Reports
  - SWAN style sheet 439

## N

- navigation

- components/pages 362, 369, 376
- including components in menu navigation 373
- setting header areas for grids 277
- using command line parameters 497
- using default style classes 553
- using the pagebar 377
- nodes
  - reordering file layout nodes 476
  - selecting for internal links 249
  - setting for the base URI 46
  - Tree Manager buttons 460
- normal forms (databases) 96
- normalized tables 96
- notifications, enabling user 376
- numeric fields
  - creating 63
  - enabling 1000 separators 315
  - inserting decimals automatically 315, 317
  - setting attributes 68, 173
  - setting to zero 315
  - understanding 63, 68

## O

- Open Definition dialog box
  - deleting fields 79
  - opening definitions 28
  - renaming fields 78
- optimization
  - selecting delete records 124
  - understanding problem types 28
- Options dialog box
  - configuring browser options 46
  - setting PeopleTools options 21
  - setting project options 57
- Oracle
  - altering tables – impact on views 178
  - converting table data 174
  - maximum character field size 68
  - maximum image field size 71
  - performing Alter in Place operations 182
  - running alter scripts on tables with logs 177
  - saving script files 164
  - using PL/SQLF 173, 177
- output window
  - activating/deactivating 16
  - navigating to build errors 162
  - understanding 14, 20
  - viewing build log output 160
- owner IDs
  - setting for components 371
  - setting for content references 402
  - setting for definitions 29
  - setting for fields 76
  - setting for menus 388
  - setting for pages 350
  - setting for projects 53
  - setting for records 123
  - setting the definitions default 22
- PAALTERNATE sub-style sheet 437
- pagebars, defining 377
- Page Control Inspector 309
- page controls
  - accessing PeopleCode 342
  - adding static text 209
  - aligning 302, 314, 340
  - applying anchor tags 328
  - associating with multicurrency processing 323
  - buttons *See Also* buttons
  - charts charts
  - check boxes check boxes
  - converting to HTML 346
  - copying/moving within/between pages 310
  - deferred mode, enabling 354
  - deleting 310
  - display control and related fields, positioning 333
  - display control and related fields, using 332
  - display controls, ordering 338
  - displaying currency symbols 315
  - drop-down list boxes
    - See Also* drop-down list boxes
  - edit boxes edit boxes
  - enable when page is display only 324
  - enabling spell check 316
  - finding in page definitions 339
  - frames *See Also* frames
  - function and data processing 210
  - grids *See Also* grids
  - group boxes group boxes
  - grouping 208
  - hiding password entries 315
  - horizontal rules *See Also* horizontal rules
  - HTML areas HTML areas
  - images images
  - inserting 302
  - inserting decimals automatically 315, 317
  - invisible, setting as 322
  - invisible controls, creating 335
  - invisible controls, modifying 323
  - invisible controls, setting labels for 323
  - labels, applying 317
  - labels, applying related display 320
  - labels, moving 310
  - labels, setting invisible control 323
  - level-based, understanding 290
  - level-based, understanding nested 291
  - level-based controls and effective dates 294
  - links *See Also* links
  - long edit boxes long edit boxes
  - manipulating 307
  - naming (system only) 327
  - order, using the default 339
  - ordering 301, 335
  - ordering display controls 338
  - ordering logically 336
  - ordering via the order list 338
  - ordering visually 336
  - order rules, applying 338
  - order testing 337
  - overlapping 340
  - populating automatically 315
  - printing a list of 355
  - properties, setting general 327
  - properties, setting use 321
  - radio buttons *See Also* radio buttons

## P

- PAALTERNATE\_SWAN sub-style sheet 438

- scroll areas scroll areas
- scroll bars scroll bars
- secondary pages secondary pages
- selecting 308
- setting control relationships between fields 323
- setting record/field properties 311
- setting styles 413
- setting to display only 322
- setting to display-only 325
- sizing 308, 314
- spacing 335, 340
- static images *See Also* static images
- static text text
- style classes, setting 314
- style classes, using 411
- style classes, using default 553
- subpages *See Also* subpages
- trees trees
- understanding 207
- understanding performance issues 341
- using aesthetic 208
- using data-entry 209
- using the Page Control Inspector 309
- using the page control toolbar 302, 307
- wrapping long words 324
- page definition report 356
- pagelets
  - moving/removing 461
  - registering homepage 402
  - setting templates for homepage pagelets 402
  - sizing pages 353
- Page Listing report 586
- Page Properties dialog box
  - changing general information 350
  - changing use information 351
  - defining secondary pages 237
- pages
  - accessing PeopleCode 342
  - accessing via keys 294
  - browsers, verifying page design in 344
  - browsers, viewing in 343
  - buttons *See Also* buttons
  - charts charts
  - check boxes check boxes
  - components components
  - creating 305
  - customizing end user pages 299
  - deferred mode, designing transactions for 193
  - deferred mode, enabling 354
  - deferred mode, using 201
  - definition menus, using 300
  - definition reports, reading 356
  - definitions, cloning 304
  - definitions, creating 11
  - definitions, identifying online 347
  - definitions, printing 355
  - definitions, understanding 289
  - deleting 354
  - designing accessible 197
  - designing inquiry pages 340
  - displaying multiple data occurrences 295
  - dragging record definitions onto pages 306
  - drop-down list boxes
    - See Also* drop-down list boxes
  - edit boxes edit boxes
  - field attributes, bulk modifications of 583
  - field definitions, dragging 305
  - fields, bulk insertion of 581
  - fields, bulk modification of 582
  - fields, deleting 141
  - fields, dragging 306
  - fields (record), bulk insertion of 580
  - frames *See Also* frames
  - grids grids
  - group boxes group boxes
  - homepages homepages
  - HTML, generating 346
  - HTML, viewing 347
  - HTML areas *See Also* HTML areas
  - images images
  - labels labels
  - levels and runtime processing 294
  - links *See Also* links
  - long edit boxes long edit boxes
  - look-up pages search pages
  - maximizing performance 341
  - mobile *See Also* mobile pages
  - navigating in components 362
  - optimizing performance 191
  - page controls *See Also* page controls
  - page definition toolbar 301
  - producing 349
  - properties, setting general 350
  - properties, setting use 351
  - radio buttons *See Also* radio buttons
  - refreshing 192, 201
  - renaming 354
  - report, Fields and Pages 585
  - report, Page Listing 586
  - report, Pages with PeopleCode 587
  - report, Records and Pages 587
  - reports, reading definition 356
  - saving 354
  - scroll areas *See Also* scroll areas
  - scroll bars scroll bars
  - search search pages
  - secondary secondary pages
  - setting styles 353
  - setting style sheets 441
  - sizing 352
  - sizing for hidden fields 354
  - subpages *See Also* subpages
  - tabs tabs
  - understanding 27
  - understanding layout 310
  - understanding spell check errors 45
  - using default style classes 553
  - using derived/work fields 298
  - using hidden 299
  - using instructional text 202, 205
  - using prompts 296
  - using related displays 195
  - using sensitive data 299
  - using static text 219
- Pages with PeopleCode report 587
- parent tables 128
- passwords
  - command line parameters 13, 154
  - hiding user entries 315
- PeopleCode
  - Debugger *See Also* PeopleCode Debugger
  - definitions, accessing PeopleCode in page 342
  - definitions, viewing for 16
  - definitions, viewing related 23
  - definition types/IDs 534
  - deleting records 142

- Display mode 116
- editing in subrecords 123
- Editor *See Also* PeopleCode Editor
- opening PeopleCode for components 364
- publishing messages 194
- report, Fields Referenced by PeopleCode Programs 586
- report, Pages with PeopleCode 587
- report, PeopleCode Programs and Field References 586
- searching for text strings 16
- selecting PeopleCode menu items 387
- understanding 22
- using deferred mode 193
- validating 18
- viewing program types 116
- PeopleCode Debugger
  - running with undocked windows 39
  - starting 17
  - viewing errors 21
- PeopleCode Display mode 116
- PeopleCode Editor
  - understanding 24
  - viewing program types 116
- PeopleCode events *See* events
- PeopleCode functions functions
- PeopleCode Programs and Field References report 586
- PeopleSoft Application Designer
  - See* Application Designer
- PeopleSoft Application Engine programs
  - Application Engine programs
- PeopleSoft Data Mover Data Mover
- PeopleSoft EDI Manager EDI Manager
- PeopleSoft financial applications 579
- PeopleSoft Online Library 19
- PeopleSoft Process Scheduler
  - See* Process Scheduler
- PeopleSoft Pure Internet Architecture
  - converting page controls to HTML 346
  - supporting button/link features 244
  - supporting style class attributes 417
  - understanding applications 9
- PeopleSoft splash screen 13
- PeopleSoft Tree Manager 460
- PeopleTools
  - administering DDL 186
  - definition types/IDs 532
  - PeopleCode *See Also* PeopleCode
  - setting options 21
  - tables *See Also* PeopleTools tables
  - using the default style sheet 437, 438
  - viewing release, database, server and user information 19
- PeopleTools tables
  - synchronizing with system catalog tables 156
  - understanding/viewing 95
- performance issues
  - designing pages and controls 341
  - optimizing page performance 191
  - ordering index keys 170
  - using prompts/drop-down lists for grids 266
- permission lists
  - adding components 399
  - understanding 397
- personalizations, grid 274
- Personalize Column and Sort Order page 267
- PL/SQL 173, 177
- pop-up menus
  - associating with page fields 393
  - button images for 392, 459
  - creating 392
  - defining transfer menu items 394
  - selecting for grids 286
  - understanding 391
  - using 37
- popup pages
  - creating 227
  - creating context sensitive help 229
  - identifying 224
  - inserting controls for 228
  - sizing 228
  - understanding 225
  - using edit boxes as 326
  - using edit boxes to control 228
  - using links to access 256
  - using message catalog with 229
  - using push buttons to access 256
- Portal Registry
  - adding components 399
  - spell check errors 44
  - understanding 397
- portals
  - assigning components 398
  - button images for 461
  - importing custom menu groups 391
  - Portal Registry *See Also* Portal Registry
  - registering components in multiple 404
  - selecting for internal links 249
  - using default style classes 553
  - viewing pages in 352
- PPLTOOLS project 96
- printing
  - button image for 459
  - list of page controls and parameters 355
  - menu definitions 390
  - page definitions 355
  - page design snapshot 355
  - page report headers, footers and borders 355
  - projects 15
  - record definitions 142
  - setting page margins 356
- Print Setup dialog box 355
- procedural language/Structured Query Language (PL/SQL) 173, 177
- processes
  - Alter *See Also* Alter process
  - batch Application Engine programs
  - Build Build process
  - business 27
  - Create Table 155
  - Find Definition References
    - See Also* Find Definition References
  - process
    - import import program
  - Install 3
  - menu import 391
  - processing groups 373
  - processing modes 191
  - Process Scheduler
    - See Also* Process Scheduler
  - Upgrade Compare Upgrade Compare process
  - Upgrade Copy Upgrade Copy process
- processing groups 373
- processing modes
  - setting based on users 382

- setting for components 376
    - understanding 191
  - Process Scheduler
    - running Crystal reports 587
    - running processes 248
  - Process Scheduler Request page 587
  - production database 524
  - profiles
    - access *See Also* access profiles
    - user 395
  - programs
    - Application Engine
      - See Also* Application Engine programs
    - Fields Referenced by PeopleCode Programs
      - report 141, 586
    - PeopleCode Programs and Field References
      - report 586
  - Project Properties dialog box 52
  - projects
    - building 500
    - building records 17
    - building record subsets 151
    - building via the Build menu 151
    - building via the command line 153
    - change *See Also* change projects
    - comparing via command line parameters 509
    - creating 52
    - definitions, accessing 50
    - definitions, automatically locking 526
    - definitions, inserting 16, 53, 58
    - definitions, inserting comments into 527
    - definitions, locking 525, 526
    - definitions, locking all 526
    - definitions, modifying 52
    - definitions, removing 55
    - handling errors for builds 153
    - locking 518
    - locking all definitions 526
    - locking definitions 525, 526
    - locking definitions automatically 526
    - merging 15, 55
    - PPLTOOLS 96
    - printing 15
    - project workspace *See Also* project workspace
    - properties, editing 15
    - properties, setting 52
    - reloading upon startup 58
    - saving 15, 52
    - setting options 21, 57
    - setting system IDs 59
    - understanding 20, 28, 49, 523
    - Upgrade Compare process
      - See Also* Upgrade Compare process
    - Upgrade Copy process Upgrade Copy process
    - upgrading application projects 18
    - using Change Control 523
    - validating 17, 21, 22, 58
    - viewing 50
    - viewing messages 21
  - project workspace
    - activating/deactivating 16
    - displaying/hiding 301
    - understanding 20, 50
    - Upgrade view *See Also* Upgrade view
    - viewing definitions 29
  - prompt buttons
    - adding 248
    - enabling 315
    - understanding 296
  - prompt dialogs
    - allowing Search events 132
    - enabling PeopleCode Search events 132
  - prompt tables
    - button image for 459
    - defining records as 139
    - naming records as 117
    - setting for fields 115, 139
    - translate tables *See Also* translate tables
    - using deferred mode 195
    - using effective dates 100
  - PSACE\_SWAN sub-style sheet 438
  - PSACE sub-style sheet 437
  - PSALTERNATE\_SWAN sub-style sheet 438
  - PSALTERNATE sub-style sheet 437
  - PSAUDIT table 126
  - PSCHGCTLHIST table 530
  - PSCHGCTLLOCK table 530
  - PSIDE.EXE *See* command line
  - PSSTYLEDEF\_SWAN style sheet 438
  - PSSTYLEDEF style sheet 437
  - PTSTYLEDEF\_SWAN sub-style sheet 438
  - PTSTYLEDEF sub-style sheet 437
  - Push Button/Hyperlink Properties dialog box
    - change hover text 256
    - labelling buttons/links 250
    - setting destination types 245
  - push button/hyperlinks
    - setting mouse over popup properties 256
  - push buttons
    - modifying SWAN 446
- ## Q
- queries
    - button images for 459
    - specifying query security records 124
    - views *See Also* query views
  - query views
    - defining records as 120
    - naming records as 118
  - quiet mode 13
- ## R
- radio buttons
    - enable when page is display only 324
    - HTML entries for 346
    - inserting 303
    - linking with records 223
    - ordering 338
    - setting label properties 223
    - setting use properties 223
    - understanding 210, 222
    - using default style classes 553
    - using in deferred mode 195
  - ReadRowset method 473, 477
  - reasonable dates 140
  - record definition report 143
  - Record Field Properties dialog box
    - selecting field labels 74
    - setting field edit properties 137
    - setting field use properties 130

- viewing editing options 115
- viewing key-related characteristics and default values 114
- record field table (RFT) *See* RFT
- record groups
  - defining 109
  - understanding 104
- Record Properties dialog box 123
- records
  - adding descriptions 123
  - adding SetIDs 107
  - altering 17
  - Alter process *See Also* Alter process
  - associating with check boxes 220
  - associating with images 218
  - audit options, setting 133
  - audit records, creating 126
  - audit records, setting user-defined 124
  - audits, using record-level 124
  - building 17
  - building subsets 151
  - creating 118
  - definition, building the active 151
  - definitions, creating 10
  - definitions, deleting 142
  - definitions, dragging 306
  - definitions, planning 98
  - definitions, printing 142
  - definitions, renaming 141
  - displaying/updating rows for component records 373
  - dynamic views *See Also* dynamic views
  - effective dates, understanding 98
  - effective dates, using 373
  - field (record) edit properties, setting 137
  - field definitions, dragging 305
  - field definitions, viewing basic 114
  - field label IDs, setting 133
  - field labels, selecting 74
  - field properties, understanding 127
  - fields, bulk insertion of 580, 581
  - fields, bulk modifications of 581
  - fields, defining set control 107
  - fields, deleting 141
  - fields, dragging 306
  - fields, identifying key 131
  - fields, inserting 122
  - fields, moving 140
  - fields, reordering 116
  - fields, setting related date 134
  - fields, setting time zone 134
  - fields, setting type ahead 134
  - fields, understanding 61, 121
  - fields (record), bulk insertion of 580
  - field types, changing 77
  - field use properties, setting 130
  - field values, setting default 134
  - file records *See Also* file records
  - groups record groups
  - indexes, creating 168
  - indexes, when to create 169
  - key fields, identifying 131
  - key-related characteristics and default values, viewing 114
  - keys, ordering 130
  - keys, using duplicate order 129
  - keys, using multiple 128
  - keys, using no 129
  - keys on parent/child tables, using 128
  - linking with radio buttons 223
  - naming 117, 123
  - navigating to build errors 162
  - opening 121
  - properties, setting 123
  - properties, setting drop-down list record 221
  - properties, setting field use 130
  - properties, setting record field edit 137
  - properties, understanding field 127
  - properties for page controls, setting 311
  - query views *See Also* query views
  - relationship to scroll areas/bars 258
  - report, Fields and Records 141, 585
  - report, record definition 143
  - report, Records and Fields 586
  - report, Records and Pages 587
  - saving 117
  - search *See Also* search records
  - securing access 124
  - selecting delete records for optimization 124
  - setting parent 124
  - setting related language 124
  - setting tablespaces 187
  - specifying relationship settings 124
  - SQL tables *See Also* SQL tables
  - SQL views SQL views
  - subrecords subrecords
  - synchronizing with application data tables 156
  - temporary tables *See Also* temporary tables
  - understanding 28, 113
  - understanding DDL 148
  - understanding performance issues 341
  - using auto-update options 125
  - using subpages 233
  - using third normal form 98
  - viewing 113
  - viewing editing options 115
  - viewing modification history 123
  - viewing PeopleCode program types 116
  - viewing records associated with set controls 111
  - work *See Also* work records
  - worklist worklist records
- Records and Fields report 586
- Records and Pages report 587
- refreshing
  - pages 192
  - Refresh buttons in components, including 382
  - related displays 195
  - using deferred mode 201
- Registration Wizard
  - accessing 398
  - registering components 11, 398, 404
  - registering iScripts 398
  - registering mobile pages 404
  - setting menu item properties 387
  - understanding 397
- related display fields
  - labelling 320
  - ordering display controls 338
  - understanding 195
  - using 333
- related edit fields, creating 334
- release numbers, available 551
- Rename Definition dialog box 32, 78

## reports

- Applications and Fields 586
- Change Control information 530
- creating compare reports via command line
  - parameters 509
- cross-reference *See Also* Crystal reports
- Field Listing 586
- Fields and Pages 585
- Fields and Records 141, 585
- Fields Referenced by PeopleCode Programs
  - 141, 586
- generating ad hoc 587
- Menu Listing 586
- page definition 356
- Page Listing 586
- Pages with PeopleCode 587
- PeopleCode Programs and Field References
  - 586
- PeopleTools Menu Listing 380, 390
- printing 15
- record definition 143
- Records and Fields 586
- Records and Pages 587
- reporting Change Control history 536
- upgrade reports, printing 15
- viewing 15
- viewing a print preview 15
- viewing for definitions 15
- Window Listing 587

## RFT

- setting button/link labels 199, 247, 252, 253
- setting field names 64, 318
- setting image labels 219

## rich text editor

- configuring 210
- configuring long edit boxes 328

## rich text fields 224

## rows

- adding multiple rows in deferred mode 195
- grids, controlling row styles for 286
- grids, disabling row insertion/deletion for 285
- grids, inserting multiple rows into 285
- grids, setting fixed row height for 288
- grids, setting odd/even row style for 287
- grids, setting row selection indicators for 286
- scroll bars, deleting rows from 264
- scroll bars, disabling row insertion/deletion for
  - 264
- scroll bars, identifying rows in 264
- scroll bars, inserting rows into 264
- scroll bars, navigating in 264

## S

## scripts

- building script files 150
- build options, setting 162
- builds, selecting file name options for 166
- builds, selecting file options for 165
- builds, selecting file overwrite options for 165
- iScripts *See Also* iScripts
- modifying invisible page controls via
  - JavaScript 323
- reviewing/updating 152
- running 152
- saving script files 164

## using comments 163

## Scroll Area Properties dialog box 261

## scroll areas

- button images for 457
- buttons, providing 249
- inserting 303
- labelling 201
- label properties, setting 260
- labels, displaying 320
- manipulating fields 259
- nesting grids 293
- ordering 338
- processing in deferred mode 285
- processing pages 258
- properties, setting body area 260
- properties, setting display 287
- properties, setting footer area 261
- properties, setting general 258
- properties, setting header area 260
- properties, setting label 260
- properties, setting use 261
- record relationships 258
- understanding 211, 257, 258
- understanding performance issues 342
- using default style classes 553

## Scroll Bar Properties dialog box 262

## scroll bars

- button images for 457
- buttons, adding scroll action 264
- buttons, providing 249
- controlling scrollable data 264
- converting to grids 265
- inserting 303
- ordering 338
- processing pages 258
- record relationships 258
- setting attributes 263
- setting field use options 264
- setting label properties 262
- setting use properties 262
- understanding 211, 257
- understanding performance issues 342

## ScrollSelect function 284

## searches

- alternate search keys
  - See Also* alternate search keys
- definitions and their references
  - Find Definition References process
- PeopleCode programs 16
- projects 21
- records *See Also* search records
- running for components 372
- search pages *See Also* search pages
- SQL programs 16

## search indexes 169

## search pages

- adding from/through logic 135
- button images for 458
- disabling advanced search features 132
- enabling history check boxes 375
- enforcing required property and table edits 132
- opening new browser windows 377
- setting component options 375
- using instructional text 375

## search records

- adding from/through search logic 135
- designing inquiry pages 340
- key lists *See Also* key lists

- overriding 381
- selecting search fields 132
- setting for components 372
- understanding 379
- secondary pages
  - accessing via buttons/links 249
  - associating with primary pages 236
  - defining 237
  - inserting 303
  - inserting controls 239
  - selecting style sheets 239
  - sizing 238, 302
  - understanding 211, 236
- security
  - Change Control 521, 522
  - enabling the Expert Entry feature 382, 383
  - granting build authority 148
  - modifying invisible page controls via JavaScript 323
  - performing bulk operations 579
  - setting up for menus 391
  - specifying query security records 124
  - using sensitive data on pages 299
- segments
  - inserting into file layouts 475, 495
  - setting properties 481
  - using 475
- servers
  - optimizing transmissions 191
  - triggering processing 192
  - using deferred mode 192, 201, 376
  - using interactive mode 376
  - using the Refresh button 382
- set control fields
  - assigning 106
  - defining 107
  - defining for trees 110
  - enabling 139
  - modifying 108
  - overriding 140
  - understanding 105
  - using associated records 115
  - viewing associated records 111
- SetDBFieldNotUsed function 114
- setIDs
  - adding to records 107
  - assigning default 108
  - creating 109
- Set System Default Colors dialog box 40
- sizing sets 148
- sliding door tabs 443
- smart punctuation 87
- spell check
  - button image for 459
  - enabling for Application Designer 22, 46
  - enabling for component fields 376
  - enabling for components 373
  - enabling for edit boxes 316
  - launching portal pages 46
  - understanding 42
  - understanding results 43
- splash screen, PeopleSoft 13
- splitter bars 38
- SQL
  - constructing definitions 24
  - DDL *See Also* DDL
  - Editor SQL Editor
  - running as part of the Build process 152
  - searching for text strings in programs 16
  - tables *See Also* SQL tables
  - understanding definitions 28
  - views *See Also* SQL views
- SQL Alter
  - creating indexes 168
  - renaming database table fields 79, 142
  - synchronizing database/PeopleSoft definitions 48
- SQL Editor
  - opening 145
  - understanding 24
- SQL tables
  - building 11
  - changing field definitions 77
  - creating 48
  - naming 145
  - records *See Also* records
  - understanding the Build process 147
- SQL views
  - creating select statements 145
  - defining records as 120
  - naming records as 118
  - updating select statements 137
- static images
  - adjusting size/shape 217
  - changing dynamically at runtime 71
  - drawing frames 209
  - HTML tag 346
  - inserting 302
  - setting properties 217
  - using controls 449
  - using default style classes 553
- status
  - activating/deactivating the status bar 16
  - indicating link status to users 430
  - showing via icons 200
  - viewing in the output window *See Also* output window
- style classes
  - accessing attributes 418
  - adding to style sheets 415
  - understanding 411
  - understanding attributes 416
  - understanding pseudo classes 430
  - using custom 413
  - using defaults 412, 553
  - visual effects 432
  - visual formatting 431
- styles 41
  - classes *See Also* style classes
  - default base styles for page controls 325
  - setting for page control labels 319
  - setting for page controls 314
  - style sheets *See Also* style sheets
  - understanding 41
- style sheets
  - background attributes, setting 423
  - background image URLs, referencing 425
  - classes 407, *See Also* style classes
  - colors, changing classic tab 573
  - colors, changing tab 443
  - colors, using 202
  - converting standard to free form 433
  - creating 413
  - creating free form 433
  - creating tab images 574
  - default, changing 437

- default, using 437, 438
- fixed form 409
- font attributes for specific languages, setting 420
- fonts, setting 418
- fonts, understanding 417
- free form sub style sheets 410
- inserting style classes 413
- override style classes 413
- pages, selecting for 353
- pages, selecting style sheets for secondary 239
- pages, setting for 441
- setting border attributes 426
- setting margins 427
- setting spacing and alignment 420
- specifying miscellaneous attributes 429
- standard 409
- sub style sheets *See Also* sub style sheets
- SWAN push buttons 446
- understanding 28, 407
- subpages
  - defining 234
  - inserting 303
  - inserting into grids 235
  - inserting into pages 234
  - setting informational labels 235
  - sizing 302
  - understanding 210, 233
- subrecords
  - defining records as 120
  - editing PeopleCode 123
  - inserting into records 122
  - naming 118
  - nesting 123
  - printing 145
  - viewing 122
- sub style sheets
  - creating 413
  - understanding 409
- sub-style sheets
  - inserting into style sheets 415
  - using default style classes 437, 438, 553
- Sybase
  - altering tables – impact on views 178
  - maximum image field size 71
  - performing Alter in Place operations 182
  - saving script files 164
- synchronization
  - Build process *See Also* Build process
  - synchronizing database/PeopleSoft definitions 48
  - synchronizing records with application data tables 156
  - using views online 184
  - when to run the Alter process 179
- system catalog tables
  - creating/understanding 95
  - synchronizing with PeopleTools tables 156
  - understanding the Build process 155
- system IDs, setting 59
- system variables
  - %Market 370
  - entering default field values 134

## T

- tables
  - altering *See Also* Alter process
  - application data application data tables
  - confirming builds 167
  - control *See Also* control tables
  - creating 156
  - creating audit tables 126
  - defining record groups 109
  - fields with edit tables 194
  - maximum columns 156
  - naming non-standard SQL tables 121
  - PeopleTools 95
  - PSAUDIT 126
  - PSCHGCTLHIST 530
  - PSCHGCTLLOCK 530
  - set control fields, assigning 106
  - set control fields, using 115
  - SetIDs, adding 107
  - setIDs, creating 109
  - setting create options 157
  - sharing 104
  - SQL *See Also* SQL tables
  - system catalog 95
  - tablesets *See Also* tablesets
  - Table Sharing feature 106, 115
  - tablespaces *See Also* tablespaces
  - temporary temporary tables
  - translate translate tables
  - understanding effective dates 98
  - understanding normalized 96
  - understanding performance issues 342
  - understanding the Build process 155
  - URL Maintenance 249, 425, 452
  - using child/parent 128
  - using key fields 128
  - viewing DDL 188
- tablesets
  - creating TableSet IDs 109
  - defining controls 110
  - defining record groups 109
  - PeopleSoft applications 104
  - sharing record groups 104
  - sharing tables 105
  - sharing trees 110
  - understanding 101
- Table Sharing feature 106, 115
- tablespaces
  - setting 187
  - viewing DDL 188
- tabs
  - displaying for components 376
  - grids, creating tabbed 271
  - grid tab separators, inserting 302
  - images, creating 574
  - images, naming 573
  - navigation buttons 458
  - ordering 201, 217, 336
  - order testing 301
  - setting styles 576
  - specifying component labels 368
  - understanding classic definitions 573
  - understanding definitions 443
  - using default style classes 553
- templates
  - page 304
  - record DDL 186
  - setting for homepage pagelets 402
- temporary tables

- creating via the Alter process 183
  - defining records as 119
  - naming 117
  - running Alter scripts 176
- test database 524
- Test Format dialog box 88
- testing
  - applications 11
  - field formats 88
  - page control order 337
  - reasonable dates 140
  - tab order 301
  - using the test database 524
- text
  - aligning 220
  - instructional text for pages 202, 205
  - instructional text for search pages, using 375
  - labels, setting button/link text 251, 252, 253
  - labels for page controls, setting text 319
  - label text properties, setting 220
  - multicurrency processing 220
  - setting styles 220
  - static, inserting 302
  - static text, HTML tag for 346
  - static text for page controls, adding 209
- three-tier execution 373
- time fields
  - creating 63
  - displaying time zones 316
  - field length in file layouts 477
  - setting attributes 70
  - understanding 63, 70
- time zones
  - adjusting time fields 70, 99
  - displaying for fields 316
  - understanding time zone fields 134
- toolbars
  - activating/deactivating 16
  - Application Designer window 19
  - buttons, adding 249
  - defining for components 375
  - page control 302, 307
  - page definition 301
  - using dockable 39
  - using dynamic 37
- Tools menu 17, 301
- transactions
  - button image for 459
  - components *See Also* components
  - optimizing accessibility 197
  - running in deferred mode 193, 194
  - using related displays 195
  - using the Expert Entry feature 382
- transfer menus, defining 394
- translate tables
  - adding values 91
  - attributes 90
  - changing values 92
  - deleting values 93
  - effective dates 91
  - enabling for fields 139
  - related display fields 324
  - saving 93
  - understanding 89
  - using 89
- translation
  - base and operator language databases 18
  - entering page control labels 318

- Tree Manager 460
- trees
  - dynamic tree controls (deprecated) 211
  - generating in HTML areas 240
  - inserting 303
  - sharing 110
  - Tree Manager buttons 460
- triggers
  - creating/understanding 183
  - triggering server processing 192
- type ahead
  - field configuration 134

## U

- Universal Resource Identifier (URI), setting the base 22, 402
- Upgrade Compare process
  - locking and upgrades 518
- Upgrade Copy process
  - databases, locking 518
  - databases for development, using multiple 524
- Upgrade Project report 15
- Upgrade view
  - setting filtering 16
  - understanding 50
  - using 51
  - viewing upgrade-only definitions 55
- upgrading applications
  - Change Control history
    - See Also* Change Control history
  - Change Control stamping
    - Change Control stamping
  - copying projects via command line
    - parameters 502
  - locking *See Also* Change Control locking
  - recommendations 299
  - setting Alter process options 181
  - Tools menu Upgrade item 18
  - Upgrade Copy process
    - See Also* Upgrade Copy process
  - Upgrade view Upgrade view
    - viewing actions 21
- URI, setting the base 22, 402
- URL Maintenance table 249, 452
- URLs
  - content references
    - See Also* content references
  - controlling via style classes 417
  - copying to the clipboard 377
  - referencing background image URLs 425
  - setting for images 452
  - URL Maintenance table 425
- Use Display mode 114
- user IDs
  - Change Control implementation
    - considerations 521
  - locked definitions, identifying 517
  - locking, understanding 517
  - sharing IDs 522
  - upgrades and locking 518
  - upgrades and stamping 521
- user profiles 395

## V

- validation
  - components 21
  - fields with edit tables 194
  - PeopleCode 17
  - projects 18, 21, 22, 58
  - WSRP 17
- version control 518
- View in Browser function
  - activating 301
  - testing page control order 337
  - understanding 301
  - using 344
- View in Browser mode
  - See Also* View in Browser function
- View menu 16, 39, 300
- views
  - altering tables 178
  - creating 158, 184
  - Development *See Also* Development view
  - dynamic dynamic views
  - query query views
  - SQL SQL views
  - understanding 184
  - Upgrade *See Also* Upgrade view
  - using joins 185
  - using keys 295

## W

- warnings
  - adding to Build logs 160
  - displaying in deferred mode 196
  - enabling save warnings for grid changes 286
  - enabling save warnings for leaving pages 245
  - handling for pages/page-controls 354
  - image file size 451
- WBMP images
  - converting 454
  - importing 452
  - understanding 450
- Window Listing report 587
- Window menu 18, 36
- windows
  - accessing context-sensitive help 19
  - activating 36
  - Application Designer window components 14
  - opening links in new windows 245
  - using dockable 39
  - using splitter bars 38
  - Window Listing report 587
  - Window menu 18
- Windows
  - building projects via the command line
    - 151, 153
  - functional indexes 169
  - saving script files 164
  - setting the three-tier execution location 373
  - sizing pages 353
- wireless bitmap (WBMP) *See* WBMP images
- workflow maps
  - understanding definitions 26
  - using images 452
- worklist records

- accessing/viewing 376
- naming 118
- work records
  - adding pages based on work records to
    - components 380
  - defining records as 120
  - naming 118
  - naming for SQR reports 118
  - referencing for pages 341
  - using derived/work fields 298
  - using hidden pages 299
  - varying prompt tables 139
- workspaces
  - definition workspace
    - See Also* definition workspace
  - project workspace project workspace
  - using splitter bars 38
- WriteRecord method 477, 483, 491
- WriteRowset method 477, 483, 491

## X

- XML files
  - file layouts, associating tags with 479
  - file layouts, displaying the buffer size for 479
  - generating 491
  - importing fields 487
  - properties 472
  - selecting XML tags for fields 485
  - understanding 468, 472