



---

# EnterpriseOne Tools 8.94

## PeopleBook: PeopleSoft AutoPilot

---

**November 2004**

EnterpriseOne Tools 8.94 PeopleBook: PeopleSoft AutoPilot  
SKU E1\_TOOLS8.94TPI-B 1104

Copyright © 2004 PeopleSoft, Inc. All rights reserved.

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

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

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

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

## Open Source Disclosure

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

### Apache Software Foundation

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

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

### OpenSSL

Copyright (c) 1998-2003 The OpenSSL Project. All rights reserved.

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

### SSLey

Copyright (c) 1995-1998 Eric Young. All rights reserved.

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

### Loki Library

Copyright (c) 2001 by Andrei Alexandrescu. This code accompanies the book:

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

# Contents

## General Preface

<b>About This PeopleBook .....</b>	<b>xiii</b>
PeopleSoft Application Prerequisites.....	xiii
PeopleSoft Application Fundamentals.....	xiii
Documentation Updates and Printed Documentation.....	xiv
Obtaining Documentation Updates.....	xiv
Ordering Printed Documentation.....	xiv
Additional Resources.....	xv
Typographical Conventions and Visual Cues.....	xvi
Typographical Conventions.....	xvi
Visual Cues.....	xvii
Country, Region, and Industry Identifiers.....	xvii
Currency Codes.....	xviii
Comments and Suggestions.....	xviii
Common Elements Used in PeopleBooks.....	xviii

## Preface

<b>PeopleSoft EnterpriseOne PeopleSoft AutoPilot Preface.....</b>	<b>xxi</b>
PeopleSoft Products.....	xxi
PeopleSoft AutoPilot.....	xxi

## Chapter 1

<b>Getting Started with PeopleSoft AutoPilot.....</b>	<b>1</b>
PeopleSoft Tools PeopleSoft AutoPilot Overview.....	1
Implementing PeopleSoft AutoPilot.....	1

## Chapter 2

<b>Using PeopleSoft AutoPilot.....</b>	<b>3</b>
Understanding PeopleSoft AutoPilot.....	3
Prerequisite.....	4
Upgrading from Service Pack 13 or Later.....	5

**Chapter 3**

<b>Using the PeopleSoft AutoPilot User Interface.....</b>	<b>7</b>
Understanding the PeopleSoft AutoPilot User Interface.....	7
Opening the PeopleSoft AutoPilot Form.....	7
Understanding Opening the PeopleSoft AutoPilot Form.....	7
Opening a PeopleSoft AutoPilot Form for Scripting.....	7
Using Panes in the PeopleSoft AutoPilot Form.....	8
Understanding Panes.....	8
Using the Command Pane.....	9
Using the Script pane.....	10
Using Bars in the PeopleSoft AutoPilot Form.....	13
Understanding the Bars in the PeopleSoft AutoPilot Form.....	13
Using the Caption Bar.....	13
Using the Menu Bar.....	13
Using the Tool bar.....	19
Using the Status Bar.....	19
Manipulating the PeopleSoft AutoPilot Form.....	20
Understanding the PeopleSoft AutoPilot Form.....	20
Changing the Size of the PeopleSoft AutoPilot Form.....	20
Arranging Multiple PeopleSoft AutoPilot Forms.....	20
Sizing Panes in the PeopleSoft AutoPilot Form.....	21
Manipulating the PeopleSoft AutoPilot Toolbar.....	21
Overview of Manipulating the PeopleSoft AutoPilot Toolbar.....	21
Relocating the Toolbar.....	22
Resizing the Toolbar.....	22
Floating the Toolbar.....	22

**Chapter 4**

<b>Scripting the Context.....</b>	<b>25</b>
Understanding Context Scripting.....	25
Understanding Context Commands.....	25
Context Command Overview.....	26
Application Command.....	26
UBE Command.....	27
Application Interconnect Command.....	30
Processing Options Command.....	31
Form Command.....	32
Header Command.....	32
Grid Column Command.....	33

QBE Command.....	33
------------------	----

## Chapter 5

<b>Writing Scripts.....</b>	<b>35</b>
Understanding How to Create a Script from Event Rule Capture.....	35
Creating a Script From Event Rule Capture.....	35
Writing the Script Using Context Commands.....	36
Understanding Writing the Script Using Context Commands.....	37
Setting the Context as a UBE.....	37
Launching a UBE.....	38
Setting the Context as an Application.....	38
Launching a UBE from a Menu.....	38
Launching a UBE from a Report Menu.....	39
Launching a UBE from a Row Menu.....	39
Launching a UBE That Is Automatically Submitted.....	41
Launching a UBE from Another UBE.....	41
Submitting a UBE.....	41
Selecting Data for a UBE.....	42
Setting UBE Processing Options.....	43
Printing a UBE.....	44
Setting the Context as an Interconnected Application.....	44
Setting the Context as a Processing Option.....	45
Defining Unwanted Windows.....	46
Setting the Context as a Form.....	47
Scripting the Form Command Using the Command Menu.....	47
Setting the Context as a Grid Column.....	48
Setting the Context as a Header.....	48
Setting the Context as a QBE Line.....	48

## Chapter 6

<b>Scripting Actions.....</b>	<b>51</b>
Understanding Scripting Actions.....	51
Using the Type To Command.....	52
Understanding the Type To Command.....	52
Using the Header Control and Grid Column Lists.....	53
Using the Source of Input List.....	53
Using Literal Values.....	53
Using a Valid Values List.....	53

Using Variables.....	54
Describing Variable Scope.....	55
Using Global Variables.....	55
Using Local Variables.....	55
Using the Value Selection List.....	59
Scripting the Type To Command.....	59
Understanding Scripting the Type To Command.....	60
Using the Header Control or Grid Column List.....	60
Using a Literal Value as a Source of Input.....	61
Creating a List of Literal Values.....	61
Creating a Valid Values List from a Simple Database Query.....	62
Using Valid Values As a Source of Input.....	62
Updating the Repeat Count in a Node.....	63
Using a Variable as a Source of Input.....	63
Declaring a Variable.....	63
Changing the Scope of a Variable.....	63
Setting the Value of a Variable.....	64
Using the Value of a Variable As a Source of Input.....	65
Updating the Value of an Existing Variable.....	65
Setting Conditional Statements.....	65
Adding a Value to a Variable.....	66
Subtracting a Value from a Variable.....	67
Concatenating a Variable.....	67
Creating a Variable to Confirm Validation Success.....	68
Creating a Variable to Store a Valid Values List Count.....	68
Using a UDC Visual Assist Value as a Source of Input.....	69
Using a Form-Interconnect Visual Assist as a Source of Input.....	69
Clearing an Input from a Header Control or Grid Column.....	70
Using the Value Selection List.....	70
Assigning a Literal Value.....	70
Assigning a Valid Values List Value.....	71
Assigning a Variable Value.....	71
Assigning a Form Interconnect Visual Assist Value.....	71
Scripting the Type To Command.....	72
Typing Data in a Header Control.....	73
Selecting Options in a Header.....	73
Typing Data in a Grid Column.....	73
Assigning a UDC Visual Assist Value.....	74
Typing Data in a QBE Line.....	75
Using the Select Grid Row Command.....	75

Understanding the Select Grid Row Command.....	75
Using the Operation Type List.....	76
Using the Action on Grid Row List.....	76
Using the Grid Columns List.....	77
Using the Source of Row Number List.....	77
Scripting the Select Grid Row Command.....	77
Understanding Scripting the Select Grid Row Command.....	77
Clicking by Row Number.....	78
Clicking by Cell Content.....	78
Performing Grid Row Operations.....	79
Using the Press Toolbar Button Command.....	80
Understanding the Press Toolbar Button Command.....	81
Using the Standard Button Option.....	81
Using the Custom Button Option.....	81
Using the Select Grid Tab Option.....	83
Using the Grid Scroll Button Option.....	83
Scripting the Press Toolbar Button Command.....	83
Understanding Scripting the Press Toolbar Button Command.....	83
Clicking a Standard Button.....	83
Clicking a Custom Button.....	85
Selecting a Grid Tab.....	86
Clicking the Grid Scroll Button.....	86
Using the Press Push Button Command.....	87
Using the Push Button Options.....	87
Using the Clickable Bitmap Options.....	87
Scripting the Press Push Button Command.....	88
Clicking a Button.....	88
Clicking a Bitmap.....	88
Using the Select ComboBox Item Command.....	89
Scripting the Select ComboBox Item Command.....	90
Using the Build Tree Path Command.....	90
Scripting the Build Tree Path Command.....	91
Understanding Scripting the Build Tree Path Command.....	91
Building a Tree Path Using Variable Values.....	91
Building a Tree Path Using Literal Values.....	92
Adding a Parent Node or Child to a Tree Path.....	92
Removing a Parent Node or a Child from a Tree Path.....	93
Using the Database Validation Command.....	93
Understanding Using the Database Validation Command.....	93
Defining Validation.....	94

Using the Validation Declaration.....	94
Using the Validation Association.....	94
Executing Validation.....	95
Using the Expect No Matching Records Option.....	96
Scripting the Database Validation Command.....	96
Declaring a Validation.....	96
Associating a Validation.....	96
Executing a Validation.....	98
Using the Command Line.....	98
Scripting a Command Line Command.....	98
Using the Command Line Command.....	99
Creating a Screen Shot of the Current Form.....	99

## Chapter 7

<b>Working with the Script pane.....</b>	<b>101</b>
Understanding the Script pane.....	101
Working with the Script pane Structure.....	101
Using Command Lines.....	102
Using the Insertion Cursor.....	103
Modifying Scripts.....	104
Understanding Changes to Scripts.....	105
Collapsing the Script Tree.....	106
Expanding the Script Tree.....	106
Using Nodes.....	106
Deleting Command Lines.....	107
Changing the Sequence of Action Commands.....	107
Changing the Sequence of Context Commands.....	107
Editing an Action Command Line.....	107
Editing a Context Command Line.....	108
Using Script Retention.....	108
Understanding Script Retention.....	108
Saving Scripts.....	109
Using the Include Command.....	109
Linking Variables Between Scripts.....	109
Sharing Scripts.....	112
Reusing Scripts.....	112
Understanding Script Reuse.....	113
Including Scripts.....	113
Including One Local Script with Another.....	113



Including a Reposited Script with Another Script.....	113
Editing an Included Script.....	114
Creating Variable Links.....	114
Declaring a Variable as External.....	115
Assigning a Default Value to an External Variable.....	115

## Chapter 8

<b>Playing Back the Script.....</b>	<b>117</b>
Understanding Script Playback.....	117
Configuring Automatic Script Playback.....	118
Understanding Automatic Script Playback.....	118
Using Playback During Script Creation.....	118
Storing and Displaying Playback Data.....	118
Handling Breakpoints.....	118
Setting Playback Speed.....	118
Setting the Cancel Playback on Comm Error Option.....	119
Setting the Log Variables on Script Failure Option.....	119
Setting Event Stream.....	119
Using Manual Script Playback Options.....	119
Understanding Manual Script Playback Options.....	120
Using the Play from Top Option.....	120
Using Play from Cursor Option.....	120
Using the Play Branch Option.....	120
Playing the Script from a Selected Line Command.....	120
Playing the Script to the Next Line Command.....	120
Toggling a Breakpoint.....	120
Continuing to a Breakpoint.....	121
Using Wait Before Proceeding.....	121
Using Script Comment.....	121
Ignoring Breakpoints During Playback.....	122
Stopping Playback.....	122
Running Script Playback.....	122
Understanding Script Playback.....	122
Playing the Script from the Top.....	123
Playing the Script from a Particular Cursor Position.....	123
Playing a Branch of the Script.....	124
Playing the Script to the Next Command.....	124
Pausing Script Playback After Beginning Playback from the Top.....	124
Pausing Script Playback After Beginning Playback from a Command Line.....	125

Toggling a Breakpoint.....	125
Playing the Script to a Breakpoint.....	125
Continuing Playback to a Breakpoint.....	126
Ignoring Breakpoints During Script Playback.....	126
Inserting a Wait Command in the Script.....	126
Failing a Script.....	127
Setting Transaction Times in the Script.....	127
Inserting a Comment in the Script.....	127

## Chapter 9

<b>Creating a Sample PeopleSoft AutoPilot Script.....</b>	<b>129</b>
Understanding the Sample PeopleSoft AutoPilot Script.....	129
Creating the Sample PeopleSoft AutoPilot Script.....	129
Understanding Sample Script Creation.....	130
Launching an Application and Form.....	130
Declaring a Variable.....	131
Adding a New Form.....	131
Typing Data in a Header Control.....	132
Creating a Valid Values List.....	133
Typing Data in a Grid Column.....	134
Updating the Repeat Count.....	135
Updating the Database.....	135
Setting the Value of a Variable.....	136
Returning to a Previous Form.....	137
Entering Data to a QBE Line.....	137
Finding Records.....	138
Selecting Records and Deleting Them from the Database.....	138
Completing the Script.....	139

## Chapter 10

<b>Storing Scripts and Test Results.....</b>	<b>141</b>
Understanding Storing Scripts and Test Results.....	141
Script Overview.....	141
Script Repository.....	142
Script Categorization.....	143
Property Pages for Scripts.....	143
Naming Conventions for Saved Scripts.....	146
Add to Repository Command.....	146

Browse Repository Scripts Command.....	146
Deletion of Scripts.....	148
Get Copy Command.....	148
Checkout Command.....	148
Undo Checkout Command.....	149
My Checkouts Form.....	149
Check In Command.....	149
Where Included Command.....	149
Working with the Script Repository.....	149
Understanding the Script Repository.....	150
Assigning Properties to a Script.....	150
Adding a Script to the Repository.....	151
Browsing for Repository Scripts.....	151
Deleting a Script from the Repository.....	151
Assigning Security to a Reposited Script.....	152
Creating a Copy of a Script.....	152
Checking Out a Script.....	153
Undoing Script Checkout.....	153
Checking in a Script.....	153
Querying for Included Scripts.....	153
Using a Command Line to Load a Repository Script.....	154
Working with Script Reporting.....	154
Understanding Script Reporting.....	154
Using the Event Stream.....	154
Using the Test Results Form.....	155
Working with PeopleSoft AutoPilot Test Manager.....	156
Understanding Test Manager.....	157
Using the Script Display Pane.....	157
Using the Script Storage Pane.....	157
Using the Test Results Pane.....	158
Using the PeopleSoft AutoPilot Test Manager Toolbar.....	159
Managing Script Testing.....	159
Understanding Script Testing.....	160
Creating a Playlist.....	160
Saving a Playlist.....	161
Running a Test.....	161
Viewing Test Results.....	161
Resetting a Test.....	162

**Glossary of PeopleSoft Terms.....163**

**Index .....183**

# About This PeopleBook

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

This preface discusses:

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

---

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

---

---

## PeopleSoft Application Prerequisites

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

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

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

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

---

## PeopleSoft Application Fundamentals

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

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

---

## Documentation Updates and Printed Documentation

This section discusses how to:

- Obtain documentation updates.
- Order printed documentation.

### Obtaining Documentation Updates

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

---

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

---

### See Also

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

### Ordering Printed Documentation

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

- Web
- Telephone
- Email

#### Web

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

#### Telephone

Contact MMA Partners at 877 588 2525.

## Email

Send email to MMA Partners at [peoplesoftpress@mmapartner.com](mailto:peoplesoftpress@mmapartner.com).

## See Also

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

---

## Additional Resources

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

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

Resource	Navigation
Troubleshooting information	Support, Troubleshooting
Upgrade documentation	Support, Documentation, Upgrade Documentation and Scripts

## Typographical Conventions and Visual Cues

This section discusses:

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

### Typographical Conventions

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

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



Typographical Convention or Visual Cue	Description
. . . (ellipses)	Indicate that the preceding item or series can be repeated any number of times in PeopleCode syntax.
{ } (curly braces)	Indicate a choice between two options in PeopleCode syntax. Options are separated by a pipe (   ).
[ ] (square brackets)	Indicate optional items in PeopleCode syntax.
& (ampersand)	When placed before a parameter in PeopleCode syntax, an ampersand indicates that the parameter is an already instantiated object.  Ampersands also precede all PeopleCode variables.

## Visual Cues

PeopleBooks contain the following visual cues.

### Notes

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

---

**Note.** Example of a note.

---

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

---

**Important!** Example of an important note.

---

### Warnings

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

---

**Warning!** Example of a warning.

---

### Cross-References

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

## Country, Region, and Industry Identifiers

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

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

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

## **Country Identifiers**

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

## **Region Identifiers**

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

- Asia Pacific
- Europe
- Latin America
- North America

## **Industry Identifiers**

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

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

## **Currency Codes**

Monetary amounts are identified by the ISO currency code.

---

# **Comments and Suggestions**

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

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

Or send email comments to [doc@peoplesoft.com](mailto:doc@peoplesoft.com).

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

---

# **Common Elements Used in PeopleBooks**

### **Address Book Number**

Enter a unique number that identifies the master record for the entity. An address book number can be the identifier for a customer, supplier, company, employee, applicant, participant, tenant, location, and so on. Depending on the application, the field on the form might refer to the address book number as the customer number, supplier number, or company number, employee or applicant id, participant number, and so on.

<b>As If Currency Code</b>	Enter the three-character code to specify the currency that you want to use to view transaction amounts. This code allows you to view the transaction amounts as if they were entered in the specified currency rather than the foreign or domestic currency that was used when the transaction was originally entered.
<b>Batch Number</b>	Displays a number that identifies a group of transactions to be processed by the system. On entry forms, you can assign the batch number or the system can assign it through the Next Numbers program (P0002).
<b>Batch Date</b>	Enter the date in which a batch is created. If you leave this field blank, the system supplies the system date as the batch date.
<b>Batch Status</b>	<p>Displays a code from user-defined code (UDC) table 98/IC that indicates the posting status of a batch. Values are:</p> <p><i>Blank:</i> Batch is unposted and pending approval.</p> <p><i>A:</i> The batch is approved for posting, has no errors and is in balance, but it has not yet been posted.</p> <p><i>D:</i> The batch posted successfully.</p> <p><i>E:</i> The batch is in error. You must correct the batch before it can post.</p> <p><i>P:</i> The system is in the process of posting the batch. The batch is unavailable until the posting process is complete. If errors occur during the post, the batch status changes to E.</p> <p><i>U:</i> The batch is temporarily unavailable because someone is working with it, or the batch appears to be in use because a power failure occurred while the batch was open.</p>
<b>Branch/Plant</b>	Enter a code that identifies a separate entity as a warehouse location, job, project, work center, branch, or plant in which distribution and manufacturing activities occur. In some systems, this is called a business unit.
<b>Business Unit</b>	Enter the alphanumeric code that identifies a separate entity within a business for which you want to track costs. In some systems, this is called a branch/plant.
<b>Category Code</b>	Enter the code that represents a specific category code. Category codes are user-defined codes that you customize to handle the tracking and reporting requirements of your organization.
<b>Company</b>	Enter a code that identifies a specific organization, fund, or other reporting entity. The company code must already exist in the F0010 table and must identify a reporting entity that has a complete balance sheet.
<b>Currency Code</b>	Enter the three-character code that represents the currency of the transaction. PeopleSoft EnterpriseOne provides currency codes that are recognized by the International Organization for Standardization (ISO). The system stores currency codes in the F0013 table.
<b>Document Company</b>	<p>Enter the company number associated with the document. This number, used in conjunction with the document number, document type, and general ledger date, uniquely identifies an original document.</p> <p>If you assign next numbers by company and fiscal year, the system uses the document company to retrieve the correct next number for that company.</p>

If two or more original documents have the same document number and document type, you can use the document company to display the document that you want.

**Document Number**

Displays a number that identifies the original document, which can be a voucher, invoice, journal entry, or time sheet, and so on. On entry forms, you can assign the original document number or the system can assign it through the Next Numbers program.

**Document Type**

Enter the two-character UDC, from UDC table 00/DT, that identifies the origin and purpose of the transaction, such as a voucher, invoice, journal entry, or time sheet. PeopleSoft EnterpriseOne reserves these prefixes for the document types indicated:

*P*: Accounts payable documents.

*R*: Accounts receivable documents.

*T*: Time and pay documents.

*I*: Inventory documents.

*O*: Purchase order documents.

*S*: Sales order documents.

**Effective Date**

Enter the date on which an address, item, transaction, or record becomes active. The meaning of this field differs, depending on the program. For example, the effective date can represent any of these dates:

- The date on which a change of address becomes effective.
- The date on which a lease becomes effective
- The date on which a price becomes effective.
- The date on which the currency exchange rate becomes effective.
- The date on which a tax rate becomes effective.

**Fiscal Period and Fiscal Year**

Enter a number that identifies the general ledger period and year. For many programs, you can leave these fields blank to use the current fiscal period and year defined in the Company Names & Number program (P0010)

**G/L Date** (general ledger date)

Enter the date that identifies the financial period to which a transaction will be posted. The system compares the date that you enter on the transaction to the fiscal date pattern assigned to the company to retrieve the appropriate fiscal period number and year, as well as to perform date validations.

# PeopleSoft EnterpriseOne PeopleSoft AutoPilot Preface

This preface discusses the PeopleSoft AutoPilot PeopleBook.

---

## PeopleSoft Products

This PeopleBook refers to this PeopleSoft product line: PeopleSoft EnterpriseOne Tools.

---

## PeopleSoft AutoPilot

This PeopleBook covers PeopleSoft AutoPilot. You use PeopleSoft AutoPilot is to create scripts to test the execution of PeopleSoft EnterpriseOne applications and to perform repetitive tasks, such as loading data, entering sales orders, or creating screen shots. Its chapters describe the tool in general and then defines its features and functionality.



# CHAPTER 1

## Getting Started with PeopleSoft AutoPilot

This chapter provides an overview of preparing to use PeopleSoft AutoPilot.

---

### PeopleSoft Tools PeopleSoft AutoPilot Overview

PeopleSoft AutoPilot is an automated testing tool that you can use to create scripts to test the execution of PeopleSoft EnterpriseOne applications and to perform repetitive tasks, such as loading data, entering sales orders, or creating screen shots.

---

### Implementing PeopleSoft AutoPilot

To use PeopleSoft AutoPilot to work with your EnterpriseOne application, these tasks must be completed first:

- You must have a valid EnterpriseOne user account.
- You must have installed PeopleSoft AutoPilot.





## CHAPTER 2

# Using PeopleSoft AutoPilot

This chapter provides an overview of PeopleSoft AutoPilot and discusses how to upgrade from Service Pack 13 or later.

---

## Understanding PeopleSoft AutoPilot

You create the scripts by using PeopleSoft AutoPilot to write commands that run essential functions and processes, such as:

- Launching applications.
- Launching forms.
- Executing form interconnections.
- Running universal batch engines (UBEs).
- Setting processing options for interactive applications and for UBEs.
- Entering data in header controls.
- Entering data in grid columns.
- Entering data in Query By Example (QBE) lines.
- Clicking toolbar buttons.
- Clicking buttons and icons.
- Selecting grid lines.
- Performing database validations.
- Selecting combo boxes.
- Traversing tree paths.

PeopleSoft AutoPilot has the flexibility to run scripts on a Windows 2000 platform, on a Java platform, or in HTML. The tool can do this because it reads and loads the specifications for each operation that you perform and passes the data through the operating system to EnterpriseOne as a keyboard input. Therefore, you can use the script to test different operating systems, environments, and data mappings without making changes to the script.

PeopleSoft AutoPilot's flexibility also enables you to:

- Save scripts on your local drive or in a script repository that is shared by others.
- Run scripts in a standalone mode or with included scripts.
- Derive values for input in your scripts from a variety of sources, including literal values, valid value lists, visual assists, and variables, which you create to store values that you can easily change as needed.

- Pass variable values within a single script or among multiple scripts.
- Test the integrity of data that you add by performing a database validation.
- Capture data about error and warning messages and about script playback events, including application programming interface (API) calls.
- Send scripts to and receive scripts from others.

In summary, PeopleSoft AutoPilot offers the following advantages for scripts that test key business processes:

- Decreases the time and effort required to create automated test scripts.
- Remains usable despite changes in the EnterpriseOne software because it reads and loads specifications directly.
- Enables users to write scripts that are compatible with future releases of the software
- Presents a user interface that hides the complexity of the tool
- Provides the user flexibility to customize scripts by changing, for example, Object Configuration Manager (OCM) mappings
- Interacts with changing technologies

PeopleSoft AutoPilot contains the following components, which enable you to create scripts:

- Command pane, where you make choices to define the processes that you want to run in EnterpriseOne software, such as launching an application
- Insert button, which enables you to insert a command to a script and create a script object that defines what the command does.
- Script pane, which contains a running log of the commands that you insert into a script.
- Toolbar, which enables you to navigate the PeopleSoft AutoPilot form and to resize it to your specifications.
- Menu bar, which contains the options that you need to run PeopleSoft AutoPilot.
- Caption bar, which identifies the script on which you are working.
- Status bar, which displays information about a PeopleSoft AutoPilot session, including processes that PeopleSoft AutoPilot is running and brief definitions of PeopleSoft AutoPilot commands.

---

**Note.** All tasks in this PeopleBook are based on the assumption that you have opened the PeopleSoft AutoPilot form and that you have either started a new script or opened an existing script.

---

## Prerequisite

Before you complete the tasks in this PeopleBook:

- You should have a working knowledge of common PeopleSoft EnterpriseOne concepts, which you can find in the Foundation Guide.
- You should also have a good understanding of at least one PeopleSoft EnterpriseOne system, such as Accounts Payable or Sales Order Entry.

---

## Upgrading from Service Pack 13 or Later

If you have a previous version of PeopleSoft AutoPilot and you upgrade to Service Pack 13 or later, you should uninstall PeopleSoft AutoPilot by deleting all of the files in the PeopleSoft AutoPilot directory. Do not delete the directory itself, because it contains your scripts, valid value lists, and so on. After you upgrade to Service Pack 13 or later, change the desktop icon so that it points to <drive>\<directory>\system\bin32\PeopleSoft AutoPilot.exe.

---

**Note.** The Open Database Connectivity (ODBC) settings should remain valid after you upgrade. If you need to reconfigure your ODBC settings for PeopleSoft AutoPilot, refer to the installation guide for your platform.

---



## CHAPTER 3

# Using the PeopleSoft AutoPilot User Interface

This chapter provides an overview of the PeopleSoft AutoPilot user interface and discusses how to:

- Open the PeopleSoft AutoPilot form.
- Use panes in the PeopleSoft AutoPilot form.
- Use bars in the PeopleSoft AutoPilot form.
- Manipulate the PeopleSoft AutoPilot form.
- Manipulate the PeopleSoft AutoPilot toolbar.

---

## Understanding the PeopleSoft AutoPilot User Interface

You use panes and bars in the PeopleSoft AutoPilot form to write commands that create a script. The form consists of two panes: the command pane and the Script pane. The command pane is the area where you make selections that create commands. As you make the selections to create a script, PeopleSoft AutoPilot displays the script as command lines in the Script pane, where you can move, delete, and edit commands. The form also contains four bars—the caption bar, menu bar, toolbar, and status bar—which you use to create and identify the script.

As you work in the PeopleSoft AutoPilot form, you can also change its shape, size, and location on the desktop for ease of use. If you are working with more than one script, you can arrange child forms within the parent form by clicking options on the menu bar. In addition, you can move the toolbar to the most convenient position within the form, or you can detach it and move it to any position within the form or move it to the desktop.

---

## Opening the PeopleSoft AutoPilot Form

This section provides an overview of opening the PeopleSoft AutoPilot form and discusses how to open the PeopleSoft AutoPilot Form for scripting.

### Understanding Opening the PeopleSoft AutoPilot Form

When you double-click the desktop icon for PeopleSoft AutoPilot, a splash screen appears, followed by the PeopleSoft AutoPilot form. Using this form, you create scripts to test PeopleSoft EnterpriseOne applications and carry out repetitive tasks. The PeopleSoft AutoPilot form initially is blank.

### Opening a PeopleSoft AutoPilot Form for Scripting

To open a PeopleSoft AutoPilot form for scripting:

1. From the desktop or the appropriate directory, launch PeopleSoft AutoPilot.
2. Select File from the PeopleSoft AutoPilot menu bar.
3. Click New.

The PeopleSoft AutoPilot form appears with some of the toolbar buttons activated.

The toolbar, which is located directly beneath the menu bar, contains buttons that represent the various commands, such as Application, that you can run in PeopleSoft AutoPilot. When you pass the mouse pointer over one of these buttons, or over one of the names in the drop-down menu under Command in the menu bar, a description of the command appears in the status bar, which is located at the bottom of the PeopleSoft AutoPilot form.

When you place the mouse pointer over the splitter bar, you can change the size of the command or Script pane by dragging the bar up or down. When you initiate a PeopleSoft AutoPilot session, the command pane is blank. You make the command pane active by initiating a command, such as Application. You can view the names of the commands by selecting Command in the menu bar.

---

## Using Panes in the PeopleSoft AutoPilot Form

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

- Use the command pane.
- Use the Script pane.

### Understanding Panes

The two major components of the PeopleSoft AutoPilot form are the command pane and the Script pane.

The command pane is the top pane of the form and is divided into lists, from which you make choices that create commands that PeopleSoft AutoPilot runs in PeopleSoft EnterpriseOne. The command pane also contains the Insert button, which you click to insert a command to the script.

The command pane enables you to make the selections that define a particular script of commands that PeopleSoft AutoPilot runs in PeopleSoft EnterpriseOne. The commands that you insert appear sequentially as command lines in the Script pane. During or after script creation, you can move, edit, or delete the command lines that you have inserted to the script.

The Script pane appears in the bottom pane of the form. It displays a running log and detailed description of the commands that you insert in the script. From the Script pane, you can move the insertion cursor (which appears as a red arrow) to any spot in the script in which you want to insert a new command. You can also reorder the script using the mouse to drag and drop command lines, and you can edit command lines by using the mouse to highlight them.

The following table describes each of the panes and the specific components that you use to accomplish script-writing tasks:

Pane	Component	Purpose
Command	Lists	Make selections from or type entries in lists to define a command that PeopleSoft AutoPilot runs in PeopleSoft EnterpriseOne.
Command	Insert button	Click this button to insert a command into the script.
Script	Insertion cursor	Move this red arrow to any spot in the script where you want to insert a new command.

## Using the Command Pane

The command pane is the area where you begin writing commands to create your PeopleSoft AutoPilot script. You begin the command-writing process by selecting a command in the Command menu on the menu bar or by clicking buttons in the toolbar. When you select commands, distinct list areas appear in the command pane.

**Note.** Neither the Insert button nor lists appear until you select a command. You make selections from (or add entries to) the command pane lists. When you click the Insert button, a command line appears in the script.

The command pane might also contain options. For example, when you select the Application option in the Command menu, the command pane contains options for Use Default Form and Processing Options Only. When you select the Select Grid Line option in the Command menu, the command pane contains options that enable you to script single-clicking or double-clicking a grid row in a form.

The two main components of the command pane are the command pane lists and the Insert button.

### Command Pane Lists

Lists are distinct areas in the command pane, physically separated from one another and individually captioned. You make selections in (or add entries to) these lists in order to write the commands that you insert in a script. A command pane list can be either populated or unpopulated. You make selections from populated lists and input data into unpopulated lists. In either case, however, you use the lists to write script commands.

Populated lists in the command pane contain the items that you select to create a script command. For example, when you select the Application option in the Command menu (or click the Launch Application button in the toolbar), PeopleSoft AutoPilot displays applications and descriptions of each application in an Applications list in the command pane.

PeopleSoft AutoPilot populates a second list, the Menu list, when you click the Launch Application button. It displays menu text, or descriptions, of forms and interactive versions that are attached to the menu selections. Versions indicate that processing options exist for the application. The list also displays the Fast Path to the form.

As you write the script, the lists in the command pane reflect selections that you make in the menu bar or on the toolbar. Other populated lists include:

- Names of header controls, grid columns, forms, forms that appear next when you add a form or interconnect to another application, buttons, previously declared variables, previously declared validations, combo box items, and options (such as radio buttons and check boxes) in forms
- Names of processing options for applications

- Sources of input to forms, such as literal values, user-defined code (UDC) visual assists, valid values lists, variables, form interconnect visual assists, header controls, or grid columns
- Sources of row numbers in a form, such as literal values, valid values lists, or variables
- Values to be input to forms, which can be derived from an existing valid values list, variable, header control, or grid column
- Sources from which a repeat count value in the script can be defined

Unpopulated lists appear with a caption, but they are empty. You create or modify the script command by typing text, numbers, special characters, spaces, or a combination of these.

You can enter the following in unpopulated lists:

- Literal values to be input in header controls, grid columns, or a Query By Example (QBE) line
- The name of a variable or validation that you are declaring
- The repeat count for a node in the script, which controls how many times the node, or tree control of commands, plays when you run a script
- The length of a wait period during script playback
- Comments to be inserted in the script
- A DOS command-line message to the system
- A name for a screen shot
- A tree path that identifies a unique path to a node in an EnterpriseOne form

## Insert Button

If you make selections and add entries to lists in the command pane and then click the Insert button, PeopleSoft AutoPilot inserts a command line in the Script pane. Each inserted command becomes a part of the script that PeopleSoft AutoPilot runs in PeopleSoft EnterpriseOne. The insertion cursor, which appears as an arrow in the Script pane, follows the last command that you insert.

When you select an application and version from the command pane and click the Insert button, you automatically launch PeopleSoft EnterpriseOne if you have activated the Playback button (identified by the initials *PB*) in the toolbar.

Clicking the Insert button for the first time starts the script. As you write the script by inserting new commands, PeopleSoft AutoPilot continues to display all scripted commands, in the order of their insertion, in the Script pane.

If the Playback button is activated, as you make selections from the command pane in PeopleSoft AutoPilot and click the Insert button, PeopleSoft AutoPilot runs the scripted commands in PeopleSoft EnterpriseOne.

## Using the Script pane

As you write a script, you can observe its progress because PeopleSoft AutoPilot records each command that you insert in the Script pane, which displays each command.

The Script pane consists of two components: command lines and the insertion cursor. The command lines reflect the selections that you make in the command pane. A command line does not appear in the Script pane until you have clicked the Insert button. Command lines indicate the context in which a command runs and the action that is taken in the context. Context commands specify *where* script actions occur, and action commands specify *which* actions occur in the script.

Command line can contain the following components:



- A symbol that designates the command as a script node.
- A symbol that identifies the specific type of context or action command that you insert in the script, such as an Application command or a Press Toolbar Button command.
- A description of the general context or action in PeopleSoft EnterpriseOne—for example, Application.
- A description of the particular context or action in PeopleSoft EnterpriseOne—for example, {P0411 - A/P Standard Voucher Entry}.
- The source of the input into a form and its value.

A value of *I*, for example, means that you have inserted the literal value 1 in a header control, grid column, or QBE line.

Context commands and action commands make up the command lines in the Script pane. A context command establishes the environment in which you write other commands. For example, to click a button in a form, you first establish the context, which is the form.

The following table summarizes the context commands that you write using PeopleSoft AutoPilot and the results of those commands:

Context Command	Result
Application	Launch PeopleSoft EnterpriseOne application.
UBE (universal batch engine)	Launch UBE, application P98305 (Batch Versions) and form W98305D (Version Prompting).
Application Interconnect	With an application and form active, launch a different application or a form in the same application that is outside the normal transaction sequence.
Processing Options	Display processing options for a selected application in the PeopleSoft AutoPilot command pane.
Set Header Control Value	Specify the header control in which to input data.
Set Grid Cell Value	Specify the grid cell in which you want to input data.
Set QBE Cell Value	Specify the grid cell in the QBE line in which to input data.
Form	Specify the form in which to take additional actions.
UBE Selection	Launch the Data Selection form.
UBE Processing Options	Display processing options for a selection UBE in the PeopleSoft AutoPilot command pane.
UBE Print	Launch the Printer Selection form.

With a context established, you can write action commands. One function of action commands is to define the actions that you take within the context that you specify. If the context is a form, an action that you can take within that form is clicking a toolbar button. Therefore, the Press Toolbar Button command is an action command.

You can write other action commands independent of a specific context. For example, you can declare a variable (give it a name) and set and store a value for it before you launch an application. Likewise, you can declare a validation and associate it with a table and columns in the table independently of establishing a context. You take these actions to accomplish something in a context. For example, you store the value of a variable to use it in a header control, grid column, combo box, or tree path.

The following table summarizes the action commands that you write using PeopleSoft AutoPilot and the results of those commands:

Action Command	Result
Select Grid Row	Select a grid row in the detail area of a form.
Build Tree Path	Create a unique path to an item in a form that uses tree controls.
Press Toolbar Button	Click standard buttons in a form, perform form and row exits, submit UBEs, select a grid tab, or click the grid scroll bar button.
Press Push Button	Click special buttons that do not reside on the toolbar of forms.
Checkbox/Radio Button	Select check box or radio button options in the header portion of a form.
Select ComboBox Item	Select items in forms that use combo boxes instead of header controls.
Press Clickable Text	Click a link on a form.
Exit OneWorld	Exit PeopleSoft EnterpriseOne.
Command Line	Encapsulate a path to another program in the PeopleSoft AutoPilot script.
Comment/Wait	Write a comment about the script and insert it into the Script pane; designate a command line and time period for PeopleSoft AutoPilot to wait before proceeding with script playback.
Variables	Declare a name for a variable, designate the source of its value, set the value, store the value.
Declare New Validation	Declare a name for a database validation.
Associate a Validation Column	Associate a table and a column with the declared validation; specify a value to be validated.
Execute Validation	Write an SQL statement to validate whether an expected value is returned from the database.
If <var> == <var>	Write a conditional (If /Then) statement.

## See Also

[Chapter 4, “Scripting the Context,” page 25](#)

[Chapter 6, “Scripting Actions,” page 51](#)

[Chapter 7, “Working with the Script pane,” page 101](#)

[Chapter 7, “Working with the Script pane,” Modifying Scripts, page 104](#)

---

## Using Bars in the PeopleSoft AutoPilot Form

This section provides an overview of the bars in the PeopleSoft AutoPilot form and discusses how to:

- Use the caption bar.
- Use the menu bar.
- Use the toolbar.
- Use the status bar.

## Understanding the Bars in the PeopleSoft AutoPilot Form

In addition to the command and Script panes, the PeopleSoft AutoPilot form includes four bars that enable you to create a script. Bars in the PeopleSoft AutoPilot form identify the script, contain options and buttons for scripting commands, and identify the functions of buttons contained in the form.

### Using the Caption Bar

The caption bar is the horizontal bar that appears at the top of the PeopleSoft AutoPilot form and identifies the name or title of the script that you are writing in PeopleSoft AutoPilot. The text enclosed in the brackets in the caption bar is the name of the script.

### Using the Menu Bar

The menu bar appears beneath the caption bar. It is composed of options that contain drop-down menus from which you can make selections that enable you to write the script and to set up how PeopleSoft AutoPilot runs.

#### File Option

Many of the selections in the drop-down menu of the File option represent essential Windows functions. You can create a new script, open an existing one, close a script, save it, or print it. The following options are PeopleSoft AutoPilot features:

Feature	Action
Send To	Enables you to send a script that you have written to another user who has access to PeopleSoft AutoPilot.
Properties	Enables you to assign identifying features to the script, such as the function that the script tests.

Feature	Action
Repository	Provides access to the script repository, which is a controlled storage location for completed scripts. This location is separate from your local drive and can be accessed by PeopleSoft AutoPilot users to obtain examples of scripts that test particular functions.
Import	Enables you to import a previously saved XML script.

### Edit Option

Two functions are available when you need to modify existing scripts. These options enable you to copy and paste command lines or branches of scripts rather than deleting a command line or branch and then having to rewrite it:

Feature	Action
Copy	Copies a command line or branch of a script from one location to the clipboard of your system. Use the Paste feature to insert the command line or branch of a script to another location. The command line or branch of a script that you copied remains in the original location.
Paste	Pastes a command line or branch of a script command into the insertion point. The Paste feature does not replace the command line or branch of a script on which you are focused. Instead, it pastes the command line or branch of a script immediately below your insertion point.

### View Option

You use the View option to set up the appearance of the toolbar and status bar in the PeopleSoft AutoPilot form.

### Command Option

The drop-down menu that appears when you select Command in the menu bar contains the commands that you can write to a script. These commands match the commands represented by the toolbar buttons that you can click to write commands to the script.

The drop-down menu under the Command menu also includes two options that are not represented by toolbar buttons. The If <var> == <var> command represents the command to write a conditional statement.

### Play Option

The drop-down menu that appears when you select Play in the menu bar contains the names of the PeopleSoft AutoPilot playback functions. These functions are also represented by toolbar buttons.

Clicking Playback on Creation toggles the Playback button on the toolbar. When the Playback button is activated, PeopleSoft AutoPilot plays the commands that you write in your script as soon as you insert them.

### Tools Menu

Using the Tools menu enables you to fine-tune the way in which a script runs, to view the results of test scripts that you have run, and to generate data that you can use in scripts. The following table summarizes the options available from the Tools menu:

Tools Menu Option	Description
Generate Valid Values List	Create or select data to store in a text file that you use in the script.
Create a Script from Capture	Create a PeopleSoft AutoPilot script from the event stream that you capture.
Include Local Script	Select a script stored locally and then include that script within another script.
Include Reposited Script	Select a script stored in the repository and then include that script within another script.
Results	Review the results of PeopleSoft AutoPilot tests that you have run.
Unwanted Windows	Close unwanted windows while executing a script.
Select PeopleSoft Client	Select from options to run a script to test Windows, Java, or HTML client.
Options	<p>Launch the Options form, which contains tabs that you use to set up the following options:</p> <ul style="list-style-type: none"> <li>• EnterpriseOne and PeopleSoft AutoPilot directories.</li> <li>• Speed.</li> <li>• Playback configuration.</li> <li>• Sign-on parameters.</li> <li>• Playback against EnterpriseOne Java, playback against EnterpriseOne HTML.</li> <li>• Script generation.</li> <li>• Specifications for script creation.</li> </ul> <hr/> <p><b>Note.</b> PeopleSoft AutoPilot does not work with Interactive HTML if Multiple Application Framework (MAF) is activated. Ensure that MAF is disabled.</p>

## Generate Valid Values List

You can create a text or numeric file that contains one or more values by selecting the Generate Valid Values List option in the Tools menu. A form appears that enables you to select data and to save it in a file. You can then use this file as a source of input for a script. When you select the Generate Valid Values List option, you use the Select Data File Type form to create a valid values list either by querying the database or by manually entering values of your own.

## Create a Script from Capture Option

You can use the Create a Script from Capture option to create a script from the event stream. The Create a Script from Capture option can greatly increase scripting productivity and shorten the time it takes to learn how to use PeopleSoft AutoPilot. The option creates a framework of a PeopleSoft AutoPilot script. It does not create a script that you can run without modification.

---

**Note.** When you use the Create a Script from Capture option to create a script, use the Fast Paths instead of the menus when calling an application.

---

### **Include Local Script Option**

You can write a script and include it with another script that tests a related function. To do so, you select the Include Local Script option from the Tools drop-down menu. This option enables you to select a script that you have saved to the local directory and to include it with another script that you select.

### **Include Reposited Script Option**

While a script is open, you can include a script that has been added to the script repository. To do so, you select the Include Reposited Script option from the Tools drop-down menu. This option enables you to browse the scripts that have been checked in to the repository, and you can select one or more of these scripts to include with a script that you select.

### **Results Option**

After you have played back a script, you can save the results of the test. PeopleSoft AutoPilot collects test results that you save and displays a summary of the test results in a Test Results form. To access the Test Results form, select Results from the drop-down menu of the Tools option on the menu bar.

### **Unwanted Windows Option**

You use the Unwanted Windows option to close windows that appear as part of the system flow but are not needed for the script's execution. Unwanted windows might include windows and message boxes such as Communications Failure, Confirm Delete, or Scheduled Packages.

You can enter, edit, and delete any windows or message boxes from a list that you create. You specify the action that occurs when the window appears. You can specify, for example, that PeopleSoft AutoPilot clicks OK on the Confirm Delete message box and continue playing the script. The Unwanted Windows option works with any language, including Chinese, Japanese, and Korean.

### **Select PeopleSoft Client**

You can run your scripts as a Windows, Java, or HTML client by selecting an option from the Select PeopleSoft Client form. If you select the Java or HTML option, you must select Options from the Tools menu and specify a server on the PeopleSoft Java tab or the PeopleSoft HTML tab.

### **Options for Configuring PeopleSoft AutoPilot**

Select Options to display a form with the following eight tabs, which contain controls and options to set up PeopleSoft AutoPilot for testing:

- Directories
- Speed
- Playback
- Sign On
- PeopleSoft Java
- PeopleSoft HTML
- Script Generation
- Configure

The Directories tab enables you to specify where you start EnterpriseOne and where you store local scripts, screen shots, and so on.

You can set the path for each directory by clicking the button next to each control. When you click the button, the Choose Directory form appears. You use this form to specify the path to each directory and the network drive on which that directory resides.

The Format option enables you to select a particular screen shot extension, such as .tif. If you do not want the option of adding scripts to the repository, you can disable it by selecting the Disable Repository option.

On the Speed tab, you can set how quickly PeopleSoft AutoPilot types in a header control or grid cell in a form.

On the Sign On tab, PeopleSoft AutoPilot displays the user ID, password, and the EnterpriseOne environment to which you sign in. If you are signed in to a different environment than the one that appears in the Environment control, PeopleSoft AutoPilot displays a form indicating that you must change the sign-in environment to match the EnterpriseOne environment.

On the PeopleSoft Java tab, you can specify a Java application server against which you can run a PeopleSoft AutoPilot script.

On the PeopleSoft HTML tab, you can enter the universal resource locator for an EnterpriseOne web server, against which you can run a PeopleSoft AutoPilot script.

On the Configure tab, you can do the following:

- Set how often the script is auto-saved.
- Select EnterpriseOne specifications, such as whether hidden edit and grid controls appear, and whether the system rebuilds file specifications each time that you run an application or only when PeopleSoft AutoPilot does not find the specifications.
- Set the threshold at which EnterpriseOne idles.
- Click the Rebuild F9860.ATX button to load application names into PeopleSoft AutoPilot.

The options on the Playback tab are divided into two sections. The top section of options enables you to configure script playback. The following table summarizes the purposes of the playback configure options:

Option	Description	Suggested Initial Setting
Play Back while Creating Script	PeopleSoft AutoPilot plays back each command after you insert it in the script.	Off.
Save Results Data after Playback	PeopleSoft AutoPilot writes data about script playback events to a table, where the results are stored.	On. You must select this option if you select any option other than None from the Events Stream Capture Level section.
Display Results Data after Playback	PeopleSoft AutoPilot displays a Results form, which contains summarized information about each playback event.	On.

Option	Description	Suggested Initial Setting
Ignore Breakpoints during Playback	During playback, PeopleSoft AutoPilot ignores breakpoints that the user manually inserts into the script. If you do not select this option, playback halts at a breakpoint until the user intervenes.	Off.
Accelerated Playback	PeopleSoft AutoPilot communicates, through code, directly with the runtime engine to determine when a process is complete so that it can go on to the next command, thus speeding up playback.	Off. Select this option only if you are certain that application launch is controlled by the runtime engine and not by a business function.
Cancel Playback on Comm Error	PeopleSoft AutoPilot cancels playback if a communication error occurs between client and server. Select this option when you are testing processes on a server.	Off.
Log Variables on Script Failure	PeopleSoft AutoPilot records the current value of variables when a script fails. This information can be useful when analyzing script failures. For example, suppose the journal date variable value is 06/03/02. This causes the script to fail if the current year is not 2002. It is recommended that you activate this option.	On.

The bottom section of options enables you to set up capture of script playback data. The chronological sequence of events that occurs during script playback is called an *event stream*. Using the options on the Playback tab, you specify how much of the event stream PeopleSoft AutoPilot captures.

You capture event stream data to accomplish two main goals. You can import the data to the Analyzer Tool, which enables you to view data about each playback event in greater detail. For example, you can view the input values and return values of individual application programming interface (API) calls, and you can see the time required to run each event. This information can aid in debugging applications.

You can also import the event stream to the Virtual Script Editor, where you can create a virtual script. You can run the virtual script on a single workstation to simulate many users. This enables you to test the scalability of your system.

The following table summarizes the options on the Playback tab that enable you set up the capture of script playback data:

Option	Description
None.	PeopleSoft AutoPilot captures no data about script playback.
PeopleSoft warning and error messages.	PeopleSoft AutoPilot captures only data about warning and error messages.



Option	Description
Level 1 API calls.	PeopleSoft AutoPilot captures warning and error messages and captures API data about only those calls that initiate a business function or database call.
All API call levels.	PeopleSoft AutoPilot captures data about warning and error messages and about all API calls.

## Window Option

The Window option provides several ways to change the size and arrangement of PeopleSoft AutoPilot forms. For example, you can select to tile or cascade the forms if you have several open at once. In addition, the drop-down menu under the Window option displays the script or scripts that are currently open.

## Help Option

Selecting the About PeopleSoft AutoPilot option on the Help menu displays the version of PeopleSoft AutoPilot that is installed on your machine, as well as the date of the build.

Selecting the Contents option on the Help menu displays the PeopleSoft online help.

## See Also

[Chapter 6, “Scripting Actions,” Using a Valid Values List, page 53](#)

## Using the Tool bar

The toolbar is composed of buttons that you click to perform the following actions:

- Script context and action commands
- Run script playback

You find the context and action commands represented by toolbar buttons in the Command menu on the menu bar. You find the script playback commands represented by toolbar buttons in the Play menu on the menu bar.

## See Also

[Chapter 4, “Scripting the Context,” page 25](#)

[Chapter 6, “Scripting Actions,” page 51](#)

[Chapter 8, “Playing Back the Script,” page 117](#)

## Using the Status Bar

The status bar, located at the bottom of the PeopleSoft AutoPilot form, provides information about the PeopleSoft AutoPilot session. For example, after you have begun a session and are preparing to enter a new command, the status bar displays *Ready* to indicate that PeopleSoft AutoPilot is ready to accept a new command. When you pass the mouse pointer over a toolbar button, the status bar displays a description of the function.

Likewise, when you pass the mouse pointer over any item that appears in a drop-down menu of the menu bar, the status bar displays a description of the function of the item.

The status bar also indicates if you need to wait before proceeding. For example, when you open a script for the first time in a session, the status bar instructs you to wait while PeopleSoft AutoPilot loads the script specifications and reads the specifications for an application that it has not yet found.

### **See Also**

[Chapter 6, “Scripting Actions,” page 51](#)

[Chapter 10, “Storing Scripts and Test Results,” Understanding the Script Repository, page 150](#)

[Chapter 10, “Storing Scripts and Test Results,” Understanding Script Reporting, page 154](#)

[Chapter 10, “Storing Scripts and Test Results,” Understanding Script Reporting, page 154](#)

---

## **Manipulating the PeopleSoft AutoPilot Form**

This section provides an overview of the PeopleSoft AutoPilot form and discusses how to:

- Change the size of the PeopleSoft AutoPilot form.
- Arrange multiple PeopleSoft AutoPilot forms.
- Size panes in the PeopleSoft AutoPilot form.

### **Understanding the PeopleSoft AutoPilot Form**

You can easily change the arrangement and size of the PeopleSoft AutoPilot form and the panes within it. You can focus completely on one pane by manipulating the size of the form. On the other hand, if you are working with multiple scripts, you can keep each of them open and arrange them so that you can conveniently move between them as you work.

You can also resize the PeopleSoft AutoPilot form and its panes when you are creating a script and playing it back. Adjusting the size of the form enables you to see both the PeopleSoft AutoPilot form and the forms that are active in EnterpriseOne.

If you close PeopleSoft AutoPilot and then open it again, the size of the PeopleSoft AutoPilot form, the arrangement of the panes, and the position of the toolbar appear as they did when you closed the session.

### **Changing the Size of the PeopleSoft AutoPilot Form**

You can easily change the size of the PeopleSoft AutoPilot form by using the mouse. Moving the mouse pointer within the form produces double-headed arrows. You then can resize the form by dragging the borders.

### **Arranging Multiple PeopleSoft AutoPilot Forms**

PeopleSoft AutoPilot enables you to create and save multiple scripts during a single session or several sessions. You can open several scripts at once, resizing and rearranging them as necessary.

You use the Window menu to change the size and arrangement of PeopleSoft AutoPilot forms when you work with multiple scripts. You can arrange the scripts so that you can view them simultaneously and easily move from one script to another.

If you work with multiple PeopleSoft AutoPilot scripts during a session, you can arrange the PeopleSoft AutoPilot forms in either cascade or tile fashion, using the Window menu. The Cascade command arranges the scripts in overlay fashion.

The top PeopleSoft AutoPilot form is active. Click another form to make it active. To resize a form, place the mouse pointer on a vertical or horizontal edge, press the mouse button, and drag the form in the direction that you desire.

The Tile command divides the area of the PeopleSoft AutoPilot form so that the existing PeopleSoft AutoPilot forms appear simultaneously, adjacent to one another.

To arrange multiple PeopleSoft AutoPilot forms:

1. Select either Cascade or Tile from the Window menu.
2. Use the mouse pointer to change the size of the parent PeopleSoft AutoPilot form or of any of the child forms.

## Sizing Panes in the PeopleSoft AutoPilot Form

You can change the size of panes in the PeopleSoft AutoPilot form easily using the Split option, which appears in the Window menu. The Split option moves the mouse pointer to the splitter bar, which divides the command pane from the Script pane. To resize the panes at any time, you can manually place the mouse pointer over the splitter bar.

To size panes in the PeopleSoft AutoPilot form:

1. Select the Split option from the Window menu.  
An arrow appears at the splitter bar, which divides the top pane from the bottom pane.
2. Drag the mouse up or down, expanding or reducing the size of the panes, and click.

---

## Manipulating the PeopleSoft AutoPilot Toolbar

This section provides an overview of manipulating the PeopleSoft AutoPilot toolbar and discusses how to:

- Relocate the toolbar
- Resize the toolbar
- Float the toolbar

### Overview of Manipulating the PeopleSoft AutoPilot Toolbar

You use the toolbar frequently during a PeopleSoft AutoPilot session because you use many of its buttons to write context and action commands. To make your work easier, you can also move the toolbar and change its size and shape.

For instance, to create more vertical space for the command pane, you can move the toolbar from near the top horizontal edge of the PeopleSoft AutoPilot form to either the right or left vertical edge. You can also float the toolbar, moving it entirely out of the PeopleSoft AutoPilot form and onto the desktop. Finally, after you have moved the toolbar from one position to another, you can return it to its original position by double-clicking the bar.

## Relocating the Toolbar

You can move the toolbar using the grabber, which is represented by a vertical bar. Two toolbars actually exist. One contains the buttons that represent action and context commands that you use to write your scripts. The other contains buttons that you use to play back scripts. Each bar contains a grabber, so you can move one, the other, or both.

To relocate the toolbar:

1. In the PeopleSoft AutoPilot form, place the mouse pointer over the grabber, which is represented by a vertical bar in the PeopleSoft AutoPilot form.
2. Holding down the mouse button, drag the toolbar, which you can now place either vertically along the right or left edge or horizontally along the bottom of the PeopleSoft AutoPilot form.

You can divide the two sections of the toolbar and place one along a vertical edge and one along a horizontal edge of the PeopleSoft AutoPilot form, or you can place them together.

## Resizing the Toolbar

You can change the size and shape of the toolbar. You can do so easily by using clicking and dragging.

To resize the toolbar:

1. In the PeopleSoft AutoPilot form, place the mouse pointer at the edge of the form.
2. When a double-headed horizontal arrow appears, press the mouse button.
3. Drag the mouse up, down, or across.

As you resize the PeopleSoft AutoPilot form, the toolbar resizes along with it.

## Floating the Toolbar

You can drag the toolbar completely outside the PeopleSoft AutoPilot form and use it on the desktop. To do so, use the mouse to grab the bar and drag it to the position that you desire, or you can double-click the bar. When the toolbar is floating, you can use the mouse to resize it.

To float the toolbar by dragging:

1. In the PeopleSoft AutoPilot form, place the mouse pointer over the grabber in the toolbar, which is represented by a vertical bar.
2. Press the mouse button.
3. Drag the toolbar to any position desired.

An outline of the toolbar appears as you drag it.

4. When the outline of the toolbar appears in the shape that you want, release the mouse button.

To float the toolbar by double-clicking:

1. Place the mouse pointer anywhere within the toolbar.
2. Double-click.
3. To return the toolbar to its original position, place the cursor in the bar that runs along its top and double-click again.

To resize and reshape the toolbar from the floating position:

1. In the PeopleSoft AutoPilot form or on the desktop, with the toolbar in a floating position, place the mouse pointer over one of its corners or edges.
2. When the double-headed vertical, horizontal, or diagonal arrow appears, click the mouse.
3. While holding down the mouse button, drag the arrow away from the bar until a resized, reshaped, outline of the bar appears.
4. Release the mouse button.  
The toolbar in its new configuration appears.
5. Double-click the top of the toolbar to return it to its original configuration.



## CHAPTER 4

# Scripting the Context

This chapter provides an overview of context scripting and discusses how to use context commands.

---

## Understanding Context Scripting

To create a script, you select options from lists in the command pane. These selections create the commands that you insert in the script, and you then play back these commands to test EnterpriseOne applications.

You can insert two kinds of commands in a PeopleSoft AutoPilot script: context commands and action commands. You use context commands to establish the setting that you test. These settings include applications, universal batch engines (UBEs), interconnected applications, processing options, forms, headers, grid columns, and QBE lines. After you establish a context, you write action commands, which accomplish specified tasks that you perform in EnterpriseOne software, such as clicking a button or typing in a header control.

Context commands can depend on other context commands. For example, suppose that you write an application command to launch an application and form. You write a header command so that you can input data in one or more header controls in the form. Although applications, forms, and header controls are all contexts, you cannot type inputs to the header controls until you have established the application and form contexts in the script.

---

## Understanding Context Commands

This section discusses:

- Context command overview.
- Application command.
- UBE command.
- Application Interconnect command.
- Processing Options command.
- Form command.
- Header command.
- Grid Column command.
- Query By Example (QBE) command.

## Context Command Overview

You write context commands during a PeopleSoft AutoPilot session to establish the context in which you work. Each of these commands establishes a unique environment, and you write each command according to the PeopleSoft AutoPilot functions that you test. In general, you must write context commands before you can decide which actions to take.

The lists in the command pane reflect the context command that you select. For example, the lists that appear in the command pane when you are writing an Application command are different from the lists that appear when you are writing a Form command. Therefore, you should become familiar with the concepts for each of the context commands.

Context commands also establish a hierarchy in the Script pane. For example, you typically begin a script by writing an Application command. In writing this command, you also select another context, a form. Both of the command lines appear in the Script pane, and PeopleSoft AutoPilot indents the Form command line beneath the Application command line. This indentation indicates that the Application command is the parent of the Form command.

This hierarchy affects script playback. Changes that you make to a parent command affect the commands that are subordinate to it. For example, if you delete a parent command from the script, the system automatically deletes all of the commands that are children of it.

The following is a simple hierarchy of PeopleSoft AutoPilot commands:

- Primary context commands are Application, Application Interconnect, UBE, and Processing Options.

These commands always provide the context for other context and action commands. They appear as parents to other commands in the script.

- Secondary context commands are Form, Header, Grid Column, and QBE.

These commands generally are subordinate to primary context commands, but they provide the context for action commands. They appear as both parents and children of other commands in the script.

- Action commands, such as clicking a toolbar button, are usually subordinate to a primary or secondary context command.

They typically appear as children of other commands in the script.

Note that these are generalizations. For example, when a Form command is the parent of a Header, Grid Column, or QBE command, it is primary to that command, but it is secondary to the Application or Application Interconnect command.

## Application Command

You use the Application command to launch interactive versions of applications. Selecting the Application command enables you to select an application, the menu text for that application, and the Fast Path for the application.

The Application command is a primary context command. You must script it in order to script inputs to header controls, grid columns, or QBE lines in forms. An Application command also is often necessary to interconnect to another application.



## UBE Command

You use the UBE command to launch previously created UBE versions when you need to submit a UBE to EnterpriseOne software for processing. PeopleSoft AutoPilot enables you to launch UBE versions from a menu, from a row or report exit, from an application that calls for a blind UBE submission, or from another UBE. After you write a UBE command, you can write other commands. You can select data for your report, set UBE processing options, submit UBE versions to the printer, and instruct PeopleSoft AutoPilot to wait for the UBE to complete processing before executing additional commands in the script. If necessary, you can also write a command that instructs PeopleSoft AutoPilot to automatically exit the Batch Versions program (P98305) when you have completed scripting the UBE submission.

### Options for the UBE Command

You can write the UBE command at various points in the script. The decision to do so depends on the process that you are testing. When you click UBE in the Command menu, the command pane lists that appear resemble the lists that appear when you click Application. You can select from the lists a UBE, a menu Fast Path to the UBE, and a version.

The command pane also contains two options:

- Execute FASTPATH
- Create Work With Batch Versions commands

PeopleSoft AutoPilot automatically activates both of these options when you select UBE. The lists in the command pane change to reflect whether you have activated these options. For example, if you select a UBE but clear the Execute FASTPATH option, PeopleSoft AutoPilot removes the Menu Item list.

### The Execute FASTPATH Option

To launch a UBE from a menu, you select the Execute FASTPATH option and then select an option from each of the three lists in the command pane: UBE, Menu Item, and Version. When you click the Insert button, PeopleSoft AutoPilot sends the Fast Path command to EnterpriseOne software.

In some cases, you launch a UBE that uses a Fast Path. For example, double-clicking a grid row or clicking a button might launch a blind submission of a UBE. You can also access a UBE from the Batch Versions program (P98305). In addition, you might launch a UBE that is coded to automatically submit another UBE. In any of these cases, you clear the Execute FASTPATH option after you select a UBE, and PeopleSoft AutoPilot removes the Menu Item list that contains the Fast Paths.

### The Create Work With Batch Versions commands Option

When you select options in the command pane and click the Insert button, PeopleSoft AutoPilot automatically declares and sets a variable that stores the UBE version that you select or that is automatically submitted. When you select the Create Work With Batch Versions commands option, PeopleSoft AutoPilot performs the following tasks without your intervention:

- Interconnects to the Batch Versions program.
- Launches the Work With Batch Versions - Available Versions form.
- Writes a QBE context command to the script.
- Inputs the stored variable value to the QBE line.
- Runs a Press Toolbar Button {Find} command.
- Selects and double-clicks a row.
- Confirms the Version Prompting form.

If you clear this option, PeopleSoft AutoPilot writes no script lines to exit to the Work With Batch Versions - Available Versions form. Disable this option when the UBE that you are launching is submitted automatically from a menu, an application, or another UBE. When you launch a UBE from a menu that is hard-coded to submit the version automatically, PeopleSoft AutoPilot removes the Version list from the command pane and disables both of the options. When you click the Insert button, PeopleSoft AutoPilot automatically submits the UBE.

## Option Combinations

Depending on the operation that you are testing, you can launch UBEs from different locations. The location dictates the combination of options that you select.

The following table shows five scenarios for launching a UBE and the combination of options that you select:

UBE Launch	Execute FASTPATH Option	Create Work With Batch Versions commands Option
From a menu.	On	On
From a Reports menu in an interactive application.	Off	On
From a Row menu in an interactive application.	Off	Off
From a menu that is hard-coded to submit the UBE as a blind execution.	Disabled	Disabled
From another UBE.	Disabled	Off

## UBE Submission

When you write a script that includes the Batch Versions program, you must write the command to submit the UBE. You do so by writing a Press Toolbar Button {Submit} command. This command clicks the Submit button on the Version Prompting form. If you want to select data for your report, select the data selection option, and then submit the report. If the UBE is a blind execution, you do not work with the Version Prompting form. The software automatically submits the UBE, and you can write the command to print it.

## UBE Data Selection

If you launch the UBE with the Create Work With Batch Versions commands option, you can also use the Criteria Design Aid feature in PeopleSoft AutoPilot to select the data for the report. The UBE context command and the UBE data selection action command work together when you script in PeopleSoft AutoPilot. After you have launched a UBE and written, either automatically or manually, a series of commands that runs through the Version Prompting form, you can use the Checkbox/Radio Button command in PeopleSoft AutoPilot to select the Data Selection option, and then submit the form by writing a Press Toolbar Button command. PeopleSoft AutoPilot then enables you to script the data selection criteria in the command pane.

You script data selection by selecting the UBE Selection option in the Command menu or by clicking the CDA button on the toolbar. When you do so, the command pane appears with five lists:

- Line Number
- Operator
- Left Operand

- Comparison
- Right Operand

In addition, the command pane contains an option that enables you to click the OK button on the Data Selection form.

The command pane lists mirror the functions of the Data Selection form. After you enter a line number for data, you determine the logic for the data selection that you want to enter to formulate your criteria. Note that the Operator and Comparison lists contain a SKIP option. If, for example, you have completed your entries to the Data Selection form for one line and you want to make entries on another line, select the SKIP option to cause the Operator and Comparison entries for the new line to duplicate the entries for the previous line.

Although the data selection feature in PeopleSoft AutoPilot is essentially the same as the Criteria Design Aid feature, the PeopleSoft AutoPilot feature has some limitations. For example, you can select a left operand in the Data Selection form by clicking a selection in a drop-down menu. The drop-down menu does not exist in PeopleSoft AutoPilot. The name of the object that populates the left operand in EnterpriseOne differs from the name that appears in the drop-down list. You must manually enter the name of the object, exactly as it appears in the EnterpriseOne list.

Likewise, you must manually enter information in the Right Operand list as they appear in the drop-down menu, or you can enter one or more literal values. You can enter multiple and range values to the right operand. You separate multiple values with commas, such as *1,2,5*; you separate a range value with two hyphens, such as *1--4*.

In addition, while you can declare a variable, set its value, and then use that value in the right operand, you must enter the name manually and enclose it in angle brackets, such as *<batchno>*. In contrast, when you use a variable as a source of input to a header control or grid column, PeopleSoft AutoPilot presents in the value selection list the names of all variables that you have declared and enables you to select one.

After you enter the name of a declared variable, PeopleSoft AutoPilot displays options in the command pane that prompt you to designate the value of the variable as a literal, a range, or a list.

Creating and using variables can make the process of selecting data for your UBE more efficient. For example, you can write a script that enters transactions, and then launches a UBE and extracts particular data for the report. Creating a variable enables you to store the data, such as a list of particular cost centers, that you need for your report. When you are ready to select the data, you enter the name of the variable in the right operand of the Data Selection form. You can store in the variable a single value, a series of discrete values, or a range of values.

When you complete one set of criteria, you click the Insert button. With the Playback button activated, you can observe the way in which PeopleSoft AutoPilot enters the criteria in the Data Selection form. If you select the UBE Selection option again, you can enter selection criteria on another line. When you have completed your data selection, select the Press OK option in the PeopleSoft AutoPilot command pane and click the Insert button. If processing options exist for the UBE version that you launch, they appear next, and you script processing options commands, as necessary.

## UBE Processing Options

After you submit a UBE, some versions prompt you to set processing options. You set these options in much the same way that you set processing options for interactive applications. However, the Command menu entries and toolbar buttons that you select to set UBE processing options are distinct from those that you select to set processing options for interactive applications.

When you set processing options for a UBE version, you select the UBE Processing Options option in the Command menu. You then select options from the Processing Options list in the command pane, and then write a Press Toolbar Button {OK} command. PeopleSoft AutoPilot inserts the selected processing options in the script and runs them during playback.

## UBE Print Command

You can send your UBE to print after you have submitted it or after EnterpriseOne has automatically submitted it. You do so by selecting the UBE Print option in the Command menu or clicking the Stop button on the toolbar.

PeopleSoft AutoPilot offers three options in the command pane after you select the UBE Print option: Wait for UBE to complete before continuing, Expect no Printer Selection window, and Create exit Work With Batch Versions commands. At this point, PeopleSoft AutoPilot cannot send UBEs to the screen in Adobe Acrobat format. When you submit a version, PeopleSoft AutoPilot automatically selects the To Printer option on the Version Prompting form.

### Wait for UBE to Complete Option

When you select this option, PeopleSoft AutoPilot submits the UBE to the default printer and waits for it to finish before resuming the script. If the UBE that you submit launches additional UBEs, PeopleSoft AutoPilot selects the printer queue. When the printer completes all of the submitted UBEs, PeopleSoft AutoPilot resumes playing the script.

If you clear this option, PeopleSoft AutoPilot submits any UBEs for printing, but resumes the script without a waiting period. You can submit your UBE from either a local or a server environment, but you cannot override the location after you select it. In either environment, PeopleSoft AutoPilot does not require your intervention to handle all print windows that appear.

### Expect No Printer Selection Window Option

You use this option if the UBE that you are running does not require a printer. This option prevents PeopleSoft AutoPilot from waiting for a printer window to appear before PeopleSoft AutoPilot continues running the script. If you clear this option and a printer window does not appear, PeopleSoft AutoPilot continues to wait, and the script fails to advance. If you select this option, PeopleSoft AutoPilot does not wait for a print window and it continues running the script after you submit the UBE.

### Create Exit Work With Batch Versions Command Option

If you launch a UBE from the Batch Versions program, you can select the Create exit Work With Batch Versions command option. Then, PeopleSoft AutoPilot automatically writes a Form command line for Work With Batch Versions - Available Versions and writes a Press Toolbar Button {Close} command. These commands confirm and close the form and display the menu item. If you do not launch a UBE from the Batch Versions program, do not select this option.

## Application Interconnect Command

The Application Interconnect command enables you to script the exit from one application to another, which might occur, for example, when you press the Add button.

Selecting Application Interconnect in the Command menu enables you to use the Script pane in PeopleSoft AutoPilot to insert new Application and Form command lines that mirror the application and form that are active in EnterpriseOne software.

You script the Application Interconnect command *reactively*; that is, you script it after you have already exited to a new application. You must script an Application Interconnect command so that the PeopleSoft AutoPilot script Application and Form commands match the application and form that are active. If you do not script the Application Interconnect command, you cannot continue scripting because the Form command line in the Script pane does not match the form and application that are active.

Remember, PeopleSoft AutoPilot also automatically writes an Application Interconnect command to the script when you launch a UBE from a menu or from a Reports menu in an interactive application. In each of these cases, you select the Create Work With Batch Versions commands option in the command pane when you select a UBE. PeopleSoft AutoPilot launches the UBE and then automatically writes a series of commands to the script, including an Application Interconnect to the Batch Versions program.

You can also script an Application Interconnect command by selecting the Press Toolbar Button option in the Command menu and then selecting the Press Custom Button option. However, you cannot use the two scripting approaches interchangeably.

When you select the Press Custom Button option to script an Application Interconnect command, you *initiate* the exit to a new application or form. PeopleSoft AutoPilot inserts the Application and Form commands in the script and launches the application and form.

Suppose that you launch the Companies application (P0010) and the Work With Companies form (W0010C), and then you need to exit to a new application. You select the Press Toolbar Button option in the Command menu. When you select the Press Custom Button option, a tree node expands.

You select Form or Row, and then, by clicking one or the other, select from various form or row menu selections, which you use to script an Application Interconnect command. These menu selections match the lists that appear when you click Form or Row in the menu of the active form.

When you select a form or row selection in PeopleSoft AutoPilot, new lists appear in the command pane. You select an application and form, and then click the Insert button. PeopleSoft AutoPilot runs the form or row selection and interconnects to the application that you chose.

You can close the interconnected application and return to the previous form. In that case, you must write another Application Interconnect command and Form command in PeopleSoft AutoPilot to ensure that the command lines in the Script pane match the application and form that are active.

When you access a new form within the same application and select Form in the Command menu, the Form list displays only the forms that are included in that application. However, if you access a form that is in a different application or is outside the normal cycle of transactions for the application, the name of that form does not appear in the list when you select the Form option in the command menu. When you select Application Interconnect in the Command menu, you can select from the command pane lists the new application and form that are active.

## See Also

[Chapter 4, “Scripting the Context,” Form Command, page 32](#)

[Chapter 5, “Writing Scripts,” Setting the Context as a Form, page 47](#)

## Processing Options Command

You can use PeopleSoft AutoPilot to set processing options for interactive versions of applications that you run. During playback, PeopleSoft AutoPilot determines whether the processing options are set as you scripted them. You script the processing options for an application and the interactive version that is attached to the menu item for the application. To do so, you select Application in the Command menu, select an application and menu item, and then select the Processing options only option in the command pane.

---

**Note.** PeopleSoft AutoPilot does not distinguish between multiple processing option labels on the same template if they are spelled identically. Even if the identically spelled processing options occur on different tabs, PeopleSoft AutoPilot cannot distinguish between them during script execution. PeopleSoft AutoPilot does not report any errors, and it operates on the last instance of any identical processing options.

---

When you click the Insert button, the command pane displays the tabs with processing options for the selected application version. With playback activated, you can view the EnterpriseOne Processing Options form and its tabs.

PeopleSoft AutoPilot serializes the processing option IDs when you create the script. When you load the script for playback, PeopleSoft AutoPilot finds the matching processing option IDs in EnterpriseOne and displays processing option text that is consistent with the release for which you play the script.

If a new release changes the processing option ID and the text, PeopleSoft AutoPilot displays an error message in the processing options command line of the Script pane when you play back the script. You can correct the processing option text in the command pane.

## Form Command

When you script an Application or an Application Interconnect command in PeopleSoft AutoPilot, you select from both the Application list and the Menu list in the command pane. The selection from the Menu list specifies the form and version that appears in EnterpriseOne when you click the Insert button. The Form command line appears automatically in the Script pane whenever you select an application and form from these lists and click the Insert button.

Scripting commands in PeopleSoft AutoPilot requires that the Form command line in the Script pane mirror the form that is active. You can verify that the two mirrors one another by selecting a form from the Next Form list in the command pane, by selecting Form in the Command menu, or by clicking the Form button on the toolbar.

### Next Form List

The Next Form list ensures that the Form command line in the Script pane matches the active form. For example, suppose that you script clicking the Add button in a form, such as the Work With Addresses form, to access another form, such as the Address Book Revision form. To do so, you select the Press Toolbar Button option in the Command menu, you select Standard Button in the command pane, and you select Add from the tree. In the Next Form list, you can select the Address Book Revisions form. When you click Insert, PeopleSoft AutoPilot inserts the Form command {Address Book Revision}.

### Form List

You might not know which form appears next in the software. Suppose that you do not know that the form Address Book Revision appears when you click Add on the Work With Addresses form. In this case, you can select Unknown/None from the Next Form list.

However, when you select Unknown/None, the Form command line in the Script pane still shows {Work With Addresses}, while the active form is Address Book Revision. If you attempt to continue scripting at this point, the script fails.

To ensure that the Form command line mirrors the form that is active, you select Form in the Command menu or click the Form button on the toolbar. In the command pane, a Form list displays the names of the forms that are included in the selected application. In this case, you select Address Book Revision from the Form list.

When you insert the command, the Form command line matches the form that is active, and you can proceed with scripting. You have confirmed that the active form matches the form name that appears in the Script pane of the PeopleSoft AutoPilot form.

## Header Command

The Header command establishes the header portion of a form as the context in which additional commands—such as clicking buttons, entering control inputs, and selecting options—can take place. You begin scripting the Header command by selecting the Set Header Control Value option in the Command menu.

The header control list that appears in the command pane includes all of the controls that are in the active form. You can select to display hidden controls by selecting Tools, Options, Configure, and then selecting the Display Hidden Edit Controls option under Spec Selection Options.

To review the properties of any header control, right-click the name of the control in the Header Control list, and then select Control Properties. The system displays the Control Properties form. This form includes four sections: Control Description, Parent Application, Edit/Display Properties, and Options. To exit the form, click Cancel.

---

**Note.** After you select a header control, you can select additional options in the command pane, including a source of input for the control and the value of the input. When you click the Insert button, PeopleSoft AutoPilot inserts two command lines in the script. The context command line is Header. However, by selecting a control, a source of input, and the value for the input, you write an additional command. This command is the Type To action command, which appears in the Script pane as a command line that shows the name of the control, as well as the source of input and the value.

---

## Grid Column Command

The Grid Column command establishes the grid column in a form as the context in which additional commands—such as clicking grid buttons and entering inputs to grid columns—can take place. You begin scripting the Grid Column command by selecting the Set Grid Cell Value option in the Command menu.

The Grid Column list that appears in the command pane includes all of the columns that are in the active form. You can display hidden columns by selecting Tools, Options, Configure, and then selecting the Display Hidden Grid Columns option under Spec Selection Options.

To review the properties of any grid column, right-click the name of the control in the Grid Column list, and then select the Control Properties form. The system displays the Control Properties form, which includes four sections: Grid, Column, Column Edit/Display Properties, and Column Properties.

---

**Note.** After you select a grid column, you make additional command pane selections, including a source of input for the control and the value of the input. When you click the Insert button, PeopleSoft AutoPilot inserts two command lines in the script. The context command line appears with the words *Detail Information*. By selecting a grid column, a source of input, and a value of the input, you write an additional command. This command is the Type To action command, which appears in the Script pane as a command line that shows the name of the grid column, as well as the source of input and the value.

---

## QBE Command

The QBE command establishes the QBE line in a form containing a grid as the context in which additional commands—such as entering inputs in the QBE line and clicking the Find button—can take place. You begin scripting the QBE command by selecting the Set QBE Cell Value option in the Command menu. You then select a grid column where you want to type inputs.

---

**Note.** After you select a grid column, you make additional command pane selections, including a source of input for the control, and the value of the input. When you click the Insert button, PeopleSoft AutoPilot inserts two command lines in the script. The context command line appears containing the words *QBE Information*. By selecting a grid column, a source of input, and a value of the input, you write an additional command. This command is the Type To action command, which appears in the Script pane as a command line that shows the name of the grid column, as well as the source of input and the value.

---





## CHAPTER 5

# Writing Scripts

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

- Create a Script from Event Rule Capture
- Write the Script Using Context Commands

---

## Understanding How to Create a Script from Event Rule Capture

You can use PeopleSoft AutoPilot to translate events in the software into a PeopleSoft AutoPilot script. Using the Create a Script from Capture option in PeopleSoft AutoPilot enables you to automatically create a script from actions that you perform in the software without having to understand the internal data relationship or the intricacies of creating a PeopleSoft AutoPilot script manually. Generating a script from the event stream can also increase the productivity of scripting and shorten the learning curve for an inexperienced user.

When you create a script from event rule capture, you create a script framework, not a finished script. PeopleSoft AutoPilot captures all the events in the stream. You might not need all those events. You will have to decide how to modify your scripts generated from the tool.

Although the script might not be fine tuned to suit your needs, these unmodified scripts have many uses. For example, you can capture a process that is causing an error in a PeopleSoft AutoPilot script and then email the script to your support organization.

---

**Note.** When you use the Create a Script from Capture option to create a script, it is recommended that you use the fast paths instead of the menus when calling an application.

---

## Creating a Script From Event Rule Capture

Access PeopleSoft AutoPilot.

To create a script from event rule capture:

1. From your desktop or the appropriate directory, launch EnterpriseOne and sign on.
2. In PeopleSoft AutoPilot, from the File menu, select New.
3. From the Tools menu, select Create a Script from Capture.
4. On Create a Script with Event Capture, complete the field Script Name'.
5. Click Start Capture.
6. On PeopleSoft EnterpriseOne Solution Explorer, complete the Fast Path' field and press Enter:

Enter a command in the Fast Path, such as 3/G11. You cannot enter an abbreviation of a program, such as OMW, UDC, OL and so on. If the fast path command does not contain a menu selection, the PeopleSoft AutoPilot script will fail.

You can capture multiple applications in a sequence as long as you always start an application by entering a command in the fast path.

7. Perform your task.

PeopleSoft AutoPilot records every event capture. Ensure that you are performing actions deliberately in order to create the most accurate script. For example, if you click the OK button twice, PeopleSoft AutoPilot records two events.

8. Click Stop Capture when you have finished your task, and then click Generate Script.

Your new script loads in the PeopleSoft AutoPilot script view pane.

9. From the File menu, select Save to save your script.

10. Modify the script as needed.

---

## Writing the Script Using Context Commands

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

- Setting the Context as a UBE
- Launching a UBE
- Setting the Context as an Application
- Launching a UBE from a Menu
- Launching a UBE from a Report Menu
- Launching a UBE from a Row Menu
- Launching a UBE That Is Automatically Submitted
- Launching a UBE from Another UBE
- Submitting a UBE
- Selecting Data for a UBE
- Setting UBE Processing Options
- Printing a UBE
- Setting the Context as an Interconnected Application
- Setting the Context as a Processing Option
- Defining Unwanted Windows
- Setting the Context as a Form
- Scripting the Form Command Using the Command Menu
- Setting the Context as a Grid Column
- Setting the Context as a Header
- Setting the Context as a QBE Line

## Understanding Writing the Script Using Context Commands

You can begin scripting context commands in one of three ways: by clicking Command in the menu bar, by clicking a hot key on the keyboard, or by clicking a toolbar button. When you do so, lists appear in the command pane. You make selections from populated lists and enter information in unpopulated lists. When you click the Insert button, PeopleSoft AutoPilot inserts one or more command lines into the Script pane. The context command is identified in the Script pane with words and symbols.

In general, the sequence that you follow to write primary context commands is as follows:

- Choose a general context, such as an interactive application or UBE, by clicking the Command menu, a hot key, or a toolbar button.
- Specify a context, such as a particular application and menu item, by making choices from or entries in lists.
- Click the Insert button to write the command to the Script pane.

Some context commands depend on other context commands. For example, Header is a context command, but you set the header as the context only after you have set an application and a form as the context for the script.

The general sequence that you follow to write secondary commands is as follows:

- Choose a general context, such as a header, grid, or QBE line.
- Specify a context, such as a control or grid column. Available controls are determined by the application and form that you previously chose.
- Choose a source of input for the specific context.
- Choose a value to be input in the specific context.

## Setting the Context as a UBE

There are several ways that you can set the context as a UBE. You can begin the script by launching the UBE from the Batch Versions program (P98305), or you can launch an interactive application and then perform a report exit to the Batch Versions program. You can launch an interactive application, then perform a row exit that launches a blind execution. You can launch a UBE from a menu that is hard-coded to submit the version automatically. Finally, you can launch a UBE that launches another UBE. In this case, EnterpriseOne software launches any subsequent UBEs and then blindly submits them without any further intervention by PeopleSoft AutoPilot.

When you click UBE in the Command menu, options for executing a Fast Path and for creating a Work With Batch Versions command appear. You can use these options to establish the way that PeopleSoft AutoPilot submits the UBE, except when a menu is hard-coded to automatically submit it.

If you select a UBE that is not automatically submitted, you must write a command to click the Submit button on the Version Prompting form. Before you write that command, however, you can click the option that enables you to select data for your report. In some cases, after you submit the UBE and select data, you can set processing options. Doing so requires you to write a UBE Processing Options command to the script and set the options by making choices from the lists in the command pane.

Finally, you can select the way to print the UBEs that you submit. You can instruct PeopleSoft AutoPilot to wait for the UBE to print before resuming running the script, or you can send the UBE to print, but tell PeopleSoft AutoPilot to continue running the script. If it is appropriate to the function you are testing, you can also write a command to close the Batch Versions program and return to Explorer.

## Launching a UBE

You can use PeopleSoft AutoPilot to launch a UBE from a variety of contexts. You might begin your script by launching a UBE from a menu. On the other hand, you might launch a UBE after you launch an interactive application. In this case, you might launch the UBE from a report menu, or you might launch it after you perform a row exit. You might also select to launch a UBE that is automatically submitted. Finally, you can launch a UBE that in turn launches one or more additional UBEs.

## Setting the Context as an Application

You often begin a script by launching an application. This process establishes both the application and the form that you work with in your script.

---

**Note.** Concerning the Menu List and Setting Context, the Menu list includes the text of the menu item in Explorer, the Fast Path, and the application version. You can launch different versions of the same application from different Explorer menus. Be sure to select the menu item that is associated with the version and processing options that you want to test.

---

Access PeopleSoft AutoPilot.

To set the context as an Application command:

1. Choose Application from the Command menu.
2. From the Application list in the command pane, click an application.
3. From the Menu list in the command pane, click the name of a menu item.
4. Click the Insert button.

PeopleSoft AutoPilot inserts the Application, Fast Path, and Form command lines into the Script pane. In the playback mode, PeopleSoft AutoPilot launches the specified version of an interactive application. The form appears on the screen with the PeopleSoft AutoPilot form, and you can navigate between the two.

## Launching a UBE from a Menu

If you want to launch a UBE from a menu, you must make choices from each of the three lists that appear in the command pane when you click UBE in the Command menu: Application, Menu Item, and Version. You also select both of the options in the command pane: Execute FASTPATH and Create Work With Batch Versions commands. The first option establishes the Fast Path PeopleSoft AutoPilot uses to access the UBE; the second option commands PeopleSoft AutoPilot to automatically perform a QBE search in the Work With Batch Versions - Available Versions form for the UBE version that you chose. When you turn this option off, you write the command to submit the UBE from the Version Prompting form.

To launch a UBE from a menu:

1. From the Command menu, select UBE.
2. In the command pane, make a selection from each of the available lists:
  - UBE
  - Menu Item
  - Version
3. Turn on both the Execute FASTPATH and Create Work With Batch Versions commands options, if available.

4. Click the Insert button.

When you click the Insert button, PeopleSoft AutoPilot automatically inserts a series of command lines in the script. The command sequence ends at the Form {Version Prompting} command line.

## Launching a UBE from a Report Menu

You might want to launch a UBE from the Reports menu in an interactive application. In this case, you begin the script by launching an interactive application. You use the Press Custom Button option to select a report. You select the UBE without the Execute FASTPATH option, and the Menu Item list containing Fast Paths to the UBEs disappears. You select the Create Work With Batch Versions commands option, which means that you write the command to submit the UBE from the Version Prompting form.

To launch a UBE from a report menu:

1. From the Command menu, select Application.
2. In the command pane, select options from the following lists:
  - Application
  - Menu
3. Click the Insert button.
4. From the Command menu, select Press Toolbar Button.

---

**Note.** Ensure that your INSERT line in the script is always at the child level.

---

5. In the Button list, expand Custom Button, and then expand Report.
6. Choose a report.
7. Click the Insert button.
8. From the Command menu, select UBE.
9. In the command pane, select a UBE.

Do not select the Execute FASTPATH option. If it is turned on, turn it off.

---

**Note.** When you clear the Execute FASTPATH option, the Menu Item list disappears. Do not clear this option until you have chosen the UBE. If you clear the option before you select the UBE, PeopleSoft AutoPilot turns the option on again after you have chosen the UBE, and you must turn it off again.

---

10. Choose a version from the Version list.
11. Turn on the Create Work With Batch Versions commands option.
12. Click the Insert button.

Because you turned on the Create Work With Batch Versions commands option, PeopleSoft AutoPilot automatically writes a series of script commands that ends at the Form {Version Prompting} command line.

## Launching a UBE from a Row Menu

You can launch a UBE from a Row menu in an interactive application. To do so, you begin by launching an interactive application. After you have written a row exit command using the Press Custom Button option, you select the UBE command. You click neither of the command pane options. The Menu Item list disappears, and you select a version. If you clear the Batch Versions option, PeopleSoft AutoPilot blindly submits the UBE.

Access the Command Menu.

To launch a UBE from a Row menu:

1. From the Command menu, select Application.
2. In the command pane, select options from the following lists:
  - Application
  - Menu
3. Click the Insert button.
4. From the Command menu, select Set QBE Cell Value.
5. In the command pane, select options from the following lists:
  - Grid Column
  - Source of Input
  - Value selection
6. Click the Insert button.
7. From the Command menu, select Press Toolbar Button.
8. In the Button list in the command pane, under the Standard Button heading, select Find.

---

**Important!** Do not select options from the Next Form list. If you select these options and insert the command, PeopleSoft AutoPilot launches the new interactive application that you chose.

---

9. Click the Insert button.
10. From the Command menu, click Select Grid Row.
11. In the command pane, click the following options:
  - Click by row number
  - Single click
12. In the Source of Row Number list, select a value source.
13. In the value selection list, enter a row number or select a variable or valid values list.
14. Click the Insert button.
15. From the Command menu, click Press Toolbar Button.
16. Click Custom Button.
17. Click Row.
18. Choose a Row selection.

---

**Important!** Ensure that you do not make choices from the Application or Next Form lists because they launch applications. You need to launch a UBE.

---

19. Click the Insert button.
20. From the Command menu, click UBE.
21. In the command pane, select a UBE from the UBE list.

---

**Important!** Turn off both the Execute FASTPATH and Create Work With Batch Versions commands options.

---

22. Choose a version from the Version list.
23. Click the Insert button.

PeopleSoft AutoPilot automatically submits the UBE.

## Launching a UBE That Is Automatically Submitted

If you launch a UBE that is hard-coded to submit the version automatically, you cannot click the options in the command pane. When you select the UBE, PeopleSoft AutoPilot disables both of the options and the Version list disappears. You select from the Menu Item list and click the Insert button, and then PeopleSoft AutoPilot launches the UBE and blindly submits it.

To launch a UBE that is automatically blindly submitted:

1. From the Command menu, click UBE.
2. In the command pane, select a UBE that will be blindly submitted from the UBE list.

When you select the UBE in this scenario, PeopleSoft AutoPilot disables both options and the Versions list disappears.

3. Choose a menu item from the Menu Item list.
4. Click the Insert button.

## Launching a UBE from Another UBE

You might launch a UBE from a menu or from an application that in turn launches one or more subsequent UBEs. In this case, EnterpriseOne software automatically launches any UBEs that are called by the first one and blindly submits them. You do not select versions or printing options, or set processing options. Without further direction from the PeopleSoft AutoPilot script, EnterpriseOne software completes all of the processes that are associated with the UBEs that the first UBE launches.

To launch a UBE from another UBE:

1. In the Command menu, follow the steps for creating a script that launches a UBE from a menu, a Report menu from an interactive application, or a Row menu from an interactive application.
2. In the command pane, select a UBE from the UBE list.

If the UBE is coded to launch another UBE, PeopleSoft AutoPilot disables the Execute FASTPATH option and removes the Menu Item list from the command pane.

3. Choose a version from the Version list.
4. Turn off the Create Work With Batch Versions command option.
5. Click the Insert button.

## Submitting a UBE

You write a command to submit the UBE only when you have turned on the Create Work With Batch Versions commands option in the command pane. When you select this option, PeopleSoft AutoPilot automatically writes commands that culminate with the Form {Version Prompting} command line. You then use the Press Standard Button option to write a command to press the Submit button on this form.

To submit a UBE:

1. Write commands through the command line Form {Version Prompting}.
2. From the Command menu, select Press Toolbar Button.
3. In the Button list in the command pane, click Standard Button.
4. Choose Submit.
5. Click the Insert button.

## Selecting Data for a UBE

After you launch a UBE, you might want to refine the data that appears in your report. If so, you can use the Criteria Design Aid feature in PeopleSoft AutoPilot, which you access either by clicking UBE Selection in the command menu or by clicking the CDA button on the toolbar. This feature enables you to script entries to the Data Selection form.

You can use the Criteria Design Aid feature when you launch a UBE with the Create Work With Batch Versions commands option turned on. If you launch the UBE from a menu, PeopleSoft AutoPilot automatically inserts a series of commands that ends at the Form {Version Prompting} command line. If you launch the UBE from a Report menu, you write a series of commands that culminates at the same point. In either case, however, when your script reaches the Form {Version Prompting} command line, you can write a command to click the Data Selection option and a command to submit the report for data selection.

At this point, you can click the UBE Selection command and use the PeopleSoft AutoPilot command pane to script entries to the Data Selection form. When you are finished, you can click the OK option in the command pane. If you stored values in a variable earlier in your script, you can use these values in the right operand of the Data Selection form. You must, however, type the name of the variable in the Right Operand list of the command pane. In addition, the variable name must be enclosed in angle brackets (<>).

After you enter a variable for the right operand, PeopleSoft AutoPilot displays options that you use to designate the value of the variable as a single value, a range of values, or a list of values.

You use Criteria Design Aid in conjunction with writing a UBE command, not as a stand-alone command. In addition, some UBEs enable you to set processing options. You use PeopleSoft AutoPilot to set the processing options for the UBE after you have selected the data that you want to appear in your report. You complete the UBE submission process by sending the report to the printer.

---

**Important!** Concerning object names, enter the object name in the left operand list exactly as it appears in the drop-down menu of the list on the Data Selection form. Likewise, enter an object name in the right operand list exactly as it appears in the drop-down menu of the list in the Data Selection form, unless you enter a literal value. If you enter a literal value, you can enter a single value, multiple values, or a range of values. You separate multiple values with commas; you separate a range of values with a hyphen.

---

To select data for a UBE:

1. From the Command menu, select UBE.
2. In the command pane, select options from the following lists:
  - UBE
  - Menu Item
  - Version
3. Turn on both the Execute FASTPATH and Create Work With Batch Versions commands options.



4. Click the Insert button.
5. From the Command menu, select Checkbox/Radio Button.
6. In the command pane, click DataSelectionYN in the Radio Button or Check Box list.
7. In the Source of Input list, click Check.
8. Click the Insert button.
9. From the Command menu, select Press Toolbar Button.
10. From the Button list, select Standard Button, and then Submit.
11. Click the Insert button.
12. From the Command menu, select UBE Selection.
13. In the command pane, complete the following fields:
  - Line Number
  - Operator
  - Left Operand
  - Comparison
  - Right Operand
14. If you enter a variable in the Right Operand list, click one of the following options that appear in the command pane in order to specify the type of value:
  - Single value
  - Range of values
  - List of values
15. Click the Insert button.
16. After you write as many UBE Selection commands as you need, click the Press OK option and then click the Insert button.

## Setting UBE Processing Options

After you submit a UBE version, you might see the Processing Options form appear. In this case, you must set processing options for the UBE before you can run the report. To set the processing options for the UBE, you select UBE Processing Options in the command menu, and then select options from the command pane.

To set UBE processing options:

1. Submit a UBE.
2. From the Command menu, select UBE Processing Options.
3. In the Processing Options list of the command pane, click the node of a processing options tab.
4. Choose a processing option.
5. Choose a source of value from the Source list (if applicable).
6. If the value is literal, enter it in the unpopulated Literal Value field. If you select Variable as the value source, PeopleSoft AutoPilot populates the Variables list, which contains the UBE version with which you are working, as well as the names of any variables for which you have set values.
7. Click the Insert button.

8. In the command pane, click the Press Toolbar Buttons node in the Processing Options list.
9. If you are satisfied with the processing options that you set up, click OK. If you are not, click Cancel.
10. Click the Insert button.

## Printing a UBE

Clicking UBE Print in the Command menu produces three options in the command pane. If you select the option Wait for UBE to complete before continuing, PeopleSoft AutoPilot submits the UBE to the printer and waits for it to complete before it resumes the script playback. If you do not select this option, PeopleSoft AutoPilot continues playing back the script without waiting for the UBE to run.

If the UBE you run does not print, click the Expect No Printer Selection Window option. This option ensures that PeopleSoft AutoPilot does not wait for a printer window to appear before it resumes the script.

Turning on the option Create exit Work With Batch Versions commands, enables you to automatically write a Form command line for Work With Batch Versions - Available Versions and a Press Toolbar Button {Close} command to return to the form that was active before launching the UBE. You select this option only if you launched your UBE with the Create exit Work With Batch Versions commands option turned on.

To print a UBE:

1. Submit the UBE and set any necessary processing options, or after PeopleSoft AutoPilot blindly submits the UBE, from the Command menu, select UBE Print.
2. In the command pane, click one or more of the following options:
  - Wait for UBE to complete before continuing
  - Expect no Printer Selection window
  - Create exit Work With Batch Versions commands

---

**Note.** Click the Create exit Work With Batch Versions commands option only if you launched your UBE from the Work With Batch Versions - Available Versions form.

---

3. Click the Insert button.

## Setting the Context as an Interconnected Application

In scripting a command to press a standard button, such as Add, you might exit from one EnterpriseOne application to another. When this occurs, you must script an application interconnection by clicking Application Interconnect in the Command menu.

For example, you might want to write a script using the Customer Ledger Inquiry application (P03B2002). If you launch the application, select the menu item Work With Customer Ledger Inquiry, then script pressing the Add button and Unknown/None from the Next Form list, EnterpriseOne software exits to a new application, Standard Invoice Entry (P03B11).

If you click Form in the Command menu, the Standard Invoice Entry form does not appear in the Form list in the command pane. This tells you that by pressing the Add button, you exited to another application. You cannot continue scripting until the Application and Form command lines in PeopleSoft AutoPilot mirror the application and form that are active.

Using the Application Interconnect command, you can ensure that your script includes the new application and form in the Script pane so that you can continue scripting. Remember that you use the Application Interconnect command *after* you exit to a new application.

To set the context as an interconnected application:

1. From the menu bar in the form to which you have exited, click Help.
2. Choose About PeopleSoft.
3. Note the application ID and form name and click OK.
4. In PeopleSoft AutoPilot, in the Command menu, Click Form.

Note that Standard Invoice Entry does not appear in the Form list.

5. In the Command menu, click Application Interconnect.
6. In the command pane, select from the lists that appear:
  - Application (select the application that is active)
  - Menu (select the form that is active)
7. Click the Insert button.

PeopleSoft AutoPilot interconnects to the new program or form, and the Application and Form command lines in the Script pane now mirror the program and form that are active. You can now script additional commands.

## Setting the Context as a Processing Option

You might want to set processing options for a particular application before you begin writing secondary commands for the application. To do so, you select an application and menu item from the command pane as if you are launching an application. However, before clicking the Insert button, you click the Processing options only option in the command pane. This option enables you to select processing options from lists in the command pane.

To set the context as a processing option:

1. From the Command menu, select Application.
2. From the Application list in the command pane, click an application.
3. From the Menu list in the command pane, click the name of a menu item.
4. In the command pane, select the Processing options only option.
5. Click the Insert button.

The command pane now displays a list of the processing options tabs for the application version you have chosen. In the Script pane, the command line shows the Launch Processing Options symbol, the template for the application, and the version of the application that you chose.

6. Expand the node of one of the tabs.
7. Choose a processing option.

PeopleSoft AutoPilot populates the Source list in the command pane with two sources of input: literal and variable.

8. Choose a source of value.

When you do so, a value selection list appears in the command pane.

9. If the value is literal, enter it in the unpopulated Literal Value field. If the value is a variable, PeopleSoft AutoPilot populates the Variables list with the names of any variables whose values you have set.
10. Click the Insert button.

PeopleSoft AutoPilot enters to the Script pane a command line that summarizes the processing options you have chosen.

---

**Note.** With playback turned on, when you insert a processing option value in PeopleSoft AutoPilot, PeopleSoft AutoPilot inserts the value to the corresponding control in the Processing Options form.

---

11. In the command pane, click the Press Buttons node in the Processing Options list.
12. If you are satisfied with the processing options you have set up, click OK. If you are not, click Cancel.
13. Click the Insert button.

---

**Note.** You can insert as many processing options to the script as you wish. You can then launch the application, if you desire. When you do so, be sure not to select the Processing options only option in the command pane.

---

## See Also

Chapter 6, “Scripting Actions,” Using a Variable as a Source of Input, page 63

## Defining Unwanted Windows

When you create scripts, windows and message boxes such as *Communication Fail*, *Confirm Delete*, or *Scheduled Packages* might appear. These windows and message boxes are not routinely recognized by PeopleSoft AutoPilot scripts and their unexpected appearance might cause your scripts to fail. To prevent script failure, you can specify what you want PeopleSoft AutoPilot to do when certain windows and message boxes appear.

You can enter, edit, and delete any windows or message boxes from a list that you create. You can also designate a subscript for PeopleSoft AutoPilot to play when a particular window or message box appears. The Unwanted Windows option works with any language, including Chinese, Japanese and Korean.

To define unwanted windows:

1. From the Tools menu, select Unwanted Windows.
2. On Close Unwanted Windows, click New to define a new unwanted window.

Alternatively, click a row in the Window To Search For And Close pane, and then click Edit.

---

**Note.** To delete an entry, click a row in the Window To Search For And Close pane, and then click Delete. The system *does not* prompt you with a confirm delete message. Once you delete the selection, you cannot recover it. You will have to enter a new unwanted window entry if you need to replace the deleted one.

---

3. On New Unwanted Window Entry, complete the 'If This Window Is Found' field:  
Enter the exact name of the form. Ensure that you enter the exact form name the way it appears in language (for example, English, French, German, and so on).
4. Choose one of the following options:
  - Then Click Button  
Choose this option if you know that the form contains a button to click. Ensure that you enter a command in the combo box (for example, OK, Cancel, and so on).

- Or Send Numeric Command

Choose this option to bypass the actual button. In Windows, when you click buttons on forms, numeric messages are sent to the window message handler. Typically the OK button sends 1 and Cancel sends 2. This method is more flexible, but not as intuitive (You can actually see what these messages are by using DevStudio's SPY++ utility.).

- Or Play Script

Choose this option to play your designated subscript if PeopleSoft AutoPilot finds the form that you specified in the If This Window Is Found field.

- Fail Script

Choose this option to make the script fail whenever the unexpected window appears. The Fail Script option overrides both the Then Click Button and the Or Send Numeric Command options.

5. Click OK.
6. On Close Unwanted Windows, click Apply, and then click OK.

## Setting the Context as a Form

When you insert an Application command in the script, PeopleSoft AutoPilot inserts in the Script pane the name of the form that you selected from the menu list. When you move to a new form, you must script the Form command to establish the context in PeopleSoft AutoPilot. You can script the Form command either from the Next Form list or from the Command menu.

To script the Form command using the Next Form list:

1. Insert an application into a script.
2. From the Command menu, select a command that enables you to switch forms, for example, Press Toolbar Button.
3. In the Button list, click a button-pressing option, such as Add, that takes you to another form.
4. In the Next Form list, click the name of the form that appears next.

The Next Form list contains the names of the forms that are included in the current application.

5. Click the Insert button.

## Scripting the Form Command Using the Command Menu

Access PeopleSoft AutoPilot.

To script the Form command using the command menu:

1. Insert an application into a script.
2. From the Command menu, select Form.

The Form list appears in the command pane. It displays the names of all forms that are included in the current application.

---

**Note.** You can also display the Form list in the command pane by clicking the Form button in the toolbar.

---

3. Choose a new form to be confirmed that matches the active form in the software.
4. Click the Insert button.

In the Script pane, the new Form command line contains the name of the active form in the software.

## Setting the Context as a Grid Column

After you launch an application and select a form, you can establish a grid column in the form as a context for further scripting. When you click Set Grid Cell Value in the Command menu, a Grid Column list appears in the command pane if the active form has a grid detail area. From this list, you can select a specific column to further refine the context.

To set the context as a grid column:

1. From the Command menu, select Set Grid Cell Value.
2. In the Grid Column list, select a grid column in which you want to enter data.
3. Choose a source of input from the Source of Input list.
4. In the value selection list, select or enter a value and click the Insert button.

When you click the Insert button, PeopleSoft AutoPilot writes both a Grid (or Detail Information) context command line and a Type To action command line in the Script pane.

## Setting the Context as a Header

After you launch an application and select a form, you can establish the header portion of the form as the context for further scripting. When you click Set Header Control Value in the Command menu, a Header Control list appears in the command pane from which you can select a specific control to further refine the context.

To set the context as a header:

1. In the Command menu of the PeopleSoft AutoPilot form, click Set Header Control Value.
2. In the Header list, select a control to which you want to input data.
3. Choose a source of input from the Source of Input list.
4. In the value selection list, select or enter a value and then click the Insert button.

When you click the Insert button, PeopleSoft AutoPilot writes both a Header context command line and a Type To action command line in the Script pane.

## Setting the Context as a QBE Line

The QBE line provides another context in which you can script commands after you launch an application and select a form.

When you click Set QBE Cell Value on the command menu, a Grid Column list appears in the command pane, if the active form has a grid detail area. You can select a specific column to further refine the context.

To set the context as a QBE line:

1. Insert an application into a script.
2. From the Command menu, select Set QBE Cell Value.
3. In the Grid Column list, select a grid column in which you want to enter data.
4. Choose a source of input from the Source of Input list.
5. In the value selection list, select or enter a value and click the Insert button.

When you click the Insert button, PeopleSoft AutoPilot writes both a Grid (or Detail Information) context command line and a Type To action command line in the Script pane.





## CHAPTER 6

# Scripting Actions

This chapter provides an overview of scripting actions and discusses how to:

- Use the Type To command.
- Script the Type To command.
- Use the Select Grid Row command.
- Script the Select Grid Row command.
- Use the Press Toolbar Button command.
- Script the Press Toolbar Button command.
- Use the Press Push Button command.
- Script the Press Push Button command.
- Use the Select ComboBox Item command.
- Script the Select ComboBox Item command.
- Use the Build Tree Path command.
- Script the Build Tree Path command.
- Use the Database Validation command.
- Script the Database Validation command.
- Associate a validation.
- Script a Command Line command.

---

## Understanding Scripting Actions

Action commands designate actions that a script carries out—for example, when users click buttons, select options, and enter data—within a context such as an application or form. Action commands require a context. They are essential because they specify the unique steps that the system takes within the context. For example, you must write action commands to create a transition between forms; to enter data in header controls, grid columns, or Query By Example (QBE) lines; to select lines in a grid; to perform database queries and updates; and so on.

Action commands also enable you to use a script to access a non-EnterpriseOne application, such as Microsoft Excel. You do this by sending a message to the system from a command line in PeopleSoft AutoPilot. You can also use the Command Line command to capture screens and store the images in a file for later use.

In addition, you can use action commands to enhance existing scripts. For example, you can write an action command to include a previously-created script within another script. For instance, in a script that requires entry of dates and then tests functions, you might include a standalone script that tests the date entry.

When you play back a created script, you can use action commands to configure the playback. For example, you can insert a Wait command in the script. This command instructs PeopleSoft AutoPilot to wait the specified length of time at a particular point in the script before it proceeds with playback. In addition, you can insert comments in the script to document the goal of the testing or to describe what occurs at a particular point during playback.

After you script the entry of form data, you can verify that PeopleSoft AutoPilot has entered the information in the specified database. The Database Validation command enables you to do that.

Action commands enable you to:

- Build scripts that test a particular set of processes.
- Test whether data is properly entered in the database.
- Modify and add comments to existing scripts.
- Configure the way that scripts run.
- Use applications external to EnterpriseOne to run a script or perform other tasks.

---

## Using the Type To Command

This section provides an overview of the Type To command and discusses how to:

- Use the Header Control and Grid Column lists.
- Use the Source of Input list.
- Use literal values.
- Use a valid values list.
- Use variables.
- Describe variable scope.
- Use global variables.
- Use local variables.
- Use the value selection list.

## Understanding the Type To Command

You use the Type To command to script inputs for header controls, grid columns, or QBE lines in a form. Unlike some action commands—such as the Press Toolbar Button command—there is no item on the Command menu or toolbar button that represents the Type To command. To write it, you use lists to specify the context as a header, grid, or QBE line; you specify a header control or grid column; you designate a source of input for the control or column; and you select a value to input into the control or column. The value can be a literal or you can derive it from a variable, a list of valid values, a user-defined command (UDC) visual assist, or a form-interconnect visual assist.

The lists that you use to write the Type To command appear in the command pane of the PeopleSoft AutoPilot form.

## Using the Header Control and Grid Column Lists

The Header Control and Grid Column lists are populated with alphabetic descriptions of the data dictionary items that are located in the header, grid, and QBE areas of the form, which is the context you have set. You click a control or column in which to script an input.

## Using the Source of Input List

After you select a header control or grid column, use the Source of Input list to select one of the following sources from which to retrieve a value to input into the header control or grid column:

- Literal value
- Valid values list
- Variable
- UDC visual assist value
- Form interconnect visual assist
- Clear source of input

## Using Literal Values

When you select the Literal Value option as a source of input, you specify that an entry in a control, grid column, or QBE line of a form appears exactly as it appears in the value selection list. For example, the literal value of a NameAlpha control entry might be *Jane Meade*, which is the exact text that PeopleSoft AutoPilot enters in the header control of a form when the script runs.

## Using a Valid Values List

When you select a literal value as a source of input, PeopleSoft AutoPilot assigns only one value to a header control, grid column, or QBE line in a form. If you select a list of valid values as a source of input, you can create a text or numeric file that contains multiple values, any of which you can enter in the header control, grid column, or QBE line. You can create a valid values list either by assigning your own values or by selecting a database and querying it for values to include in the list.

You can use a list of valid values as an input source to run a script multiple times and, each time, enter a different value in a specified header control, grid column, or QBE line. As PeopleSoft AutoPilot loops through the script, the value that it enters in the control or column changes to reflect the values in the list. Alternatively, you might run the script once but enter five different values in a grid column. You can do this by creating a single list of valid values that contains five items.

If you exit a script and exit PeopleSoft AutoPilot, and then you open the script again, PeopleSoft AutoPilot resets the list of valid values. That way, when you play back the script, the first value that you entered in the list appears first. If you exit a script without exiting PeopleSoft AutoPilot and then open the script again, PeopleSoft AutoPilot uses the value that is next in order on the list when you exit the script.

## Using Variables

You can select a value, store it, and then use it at a subsequent place in the script. You can also use the value more than once in the script. In this case, use a variable as a source of input and store its value anywhere in the script. You declare the variable to assign a name to it. Then you set and store its value, which you can retrieve from a list of valid values, header control, grid column, another variable, or from the literal input.

The Command pane of the PeopleSoft AutoPilot form displays the following components when you write a variable command:

- New Variable list.

You enter the name of the variable in this list, thereby declaring the variable.

- External Variable option.

By selecting this option, you specify that the variable can be linked to a variable in another script so that its value can be passed between scripts.

- Default Variable list.

You can enter a value that PeopleSoft AutoPilot uses even if you do not set a value for the variable.

- Existing Variable list.

PeopleSoft AutoPilot displays the names of any existing variables that you have declared in the script.

- Source of Value list.

You select a source of value for the variable, such as a literal value, a list of valid values, a header control, a grid cell, or another variable. In addition, you can select variable manipulations, such as adding or subtracting a literal value or a variable value from the variable.

- Value Selection list.

You enter a literal value or select the object that contains the value that you store in the variable. For example, if you select a header control as the source of value, you use the value selection list to select the specific control that contains the value. You can also obtain variable values from lists of valid values, from variables to which you have previously assigned a value, or from PeopleSoft EnterpriseOne sources, such as error and warning messages or grid row counts.

The following table explains other key terms that are related to variables in PeopleSoft AutoPilot scripts:

Term	Description
Variable scope	The range of commands within a script in which the value for a variable can be used.
Global variable	A variable for which the value can be used throughout an entire script.
Local variable	A variable for which the value can be used only within a portion of a script.
External variable	A variable that can be linked to a variable in another script so that a variable value can be passed between scripts.
Default value	A value that you assign to a variable that PeopleSoft AutoPilot uses when you do not set the value of the variable elsewhere in the script.

Term	Description
Conditional statement	An If /Then statement that you write by comparing the values of two variables. The statement stipulates that if a condition exists in the script, then the script should run other commands.
Variable concatenation	The practice of stringing together two or more variables to create a new variable.
System variable	A variable for which the value is derived from EnterpriseOne data, such as error and warning messages.
Valid values count	A variable for which the value is derived from the number of items in a list of valid values.
Variable watch list	A list that tracks variable values that are used during script playback.
Validation success	A variable for which the value indicates the success or failure of a database validation.

## Describing Variable Scope

The term *variable scope* refers to how broadly you can use the value of the variable within a script. You create a node each time that you write a context command. The node in which you declare a variable determines its scope. For example, if you declare a variable within an Application command node, the scope of the variable extends to that node only, and you can use a value that you set for the variable only within that node. If, for example, you declare a variable within an Application command node and then you launch another application, you cannot use the value that you set for this variable within the new Application command node. If you declare a variable within a Form command node, its scope extends only to that form.

## Using Global Variables

The scope of a variable is global when you can use its value throughout the entire script. To establish global scope for a variable, you must make the Declare variable command a child of the Begin Script node, which is always the first node in the script.

## Using Local Variables

You can use the value of a local variable only within a portion of the script, specifically the node to which you attach it. For example, you might declare the variable immediately after you launch an application and a form. In this instance, you can set the value of the variable and use this value only for any command lines that you script within the Form command node because the Declare variable command that you write is a child of the Form command node.

You can expand the scope of the variable by dragging it to another node that is higher in the script. For example, you might drag the Declare Variable command from the Form command node to the Application command node, which makes it a child of the Application command. This action broadens the scope of the variable, and you can use the value that you set for it anywhere within the application.

However, the scope of the variable remains local. If you launch another application later in the script, you cannot use the value of the variable within that new Application command unless you make this command a child of the first Application command.

## External Variables

A variable with global scope enables you to pass a value to header controls, grid columns, and QBE lines throughout a single, standalone script. PeopleSoft AutoPilot also enables you to declare a variable as external, which means that you want to link the variable to a variable in another script. You use external variables when you want to pass variable values between scripts. For example, you might store a batch number in one script. If you declare the variable that stores the batch number as external, you can link that variable to external variables in one or more other scripts, and pass the batch number value to other scripts.

## Default Values for Variables

You can create a script that you can use both in standalone mode and with other scripts. To do so, you assign a default value to the variable. For example, you might create script B that links to script A, which passes along a batch number value. Suppose, however, that you want to use script B by itself. If you set a default value in script B, PeopleSoft AutoPilot uses that value when you use script B in standalone mode.

You can also assign a default value to a variable that you do not declare as external. PeopleSoft AutoPilot uses the default value of the variable each time that you use it as a source of input for a header control, grid column, or QBE line. If you write a command to set a value for the variable, that value overrides the default value.

## Conditional Statements

Conditional statements enable you to write If/Then/Else commands that compare the values of two variables for which you have declared names and set values. If the script meets the conditions that appear in the statement, then PeopleSoft AutoPilot runs an additional branch of the script. Conversely, you can write commands that are connected to an Else branch in the script, or you can enable the script to end if it does not meet the condition that appears in the statement.

You can write a conditional statement to ensure that a script tests an application even if the script does not meet the conditions that you expect to exist. For example, suppose that you test making revisions to an existing Address Book number. If the Address Book number exists, then PeopleSoft AutoPilot selects the grid line in the Work With Addresses form, double-clicks the line, and then revises the existing Address Book number in the Address Book Revision form. However, if the Address Book number does not exist, the script fails unless you write a conditional statement stipulating that if the Address Book number does not exist, PeopleSoft AutoPilot adds a form and runs commands to create a new Address Book entry.

A different conditional statement might test the converse. If the Address Book number does not equal the number that PeopleSoft AutoPilot returns to the QBE line of the Work With Addresses form, the Address Book number does not exist. If that condition is met, PeopleSoft AutoPilot clicks Add and creates a new entry. If the Address Book number does exist, PeopleSoft AutoPilot double-clicks the grid line and revises the Address Book entry. This is the Else portion of the statement.

You can also compare variables between scripts by declaring a variable as external in one script, including the script with a parent script, and linking the external variable to a variable in the master script. You then build your conditional statement on these two variables.

You can use PeopleSoft AutoPilot to set conditional statements of data equality or inequality, but the tool does not enable you to develop compound conditional statements that link together.

## Variable Addition

Variable addition enables you to scroll through a grid from top to bottom. For example, you can write a command that adds 1 to the row number of the grid each time PeopleSoft AutoPilot plays back the node of the script. If you set the repeat count of the node to match the number of lines in the grid, PeopleSoft AutoPilot scrolls through the entire grid, one line at a time, from top to bottom.

## Variable Subtraction

Variable subtraction enables you to scroll through the entire grid, one line at a time, from bottom to top. You write a command that subtracts 1 from the row number of the grid each time PeopleSoft AutoPilot plays back the node of the script.

## Variable Concatenation

Variable concatenation enables you to string together existing variable values to create a new variable. For example, you might have created two variables with values of 10 and 25, respectively. You can concatenate them to create a new variable that has a value that you use to select a range of values for a Universal Batch Engine (UBE).

## System Variables

System variables obtain their values from EnterpriseOne, rather than from the information that you enter in PeopleSoft AutoPilot. To use a system variable, you do not need to declare a variable or set its value because its value is determined during script playback.

The following table names the system variables and presents examples of how they might be used in script writing:

System Variable	Description	Possible Uses
Errors	PeopleSoft AutoPilot records the number of EnterpriseOne error messages that occur during script playback.	Use in conjunction with a conditional statement. For example, set a condition specifying that if the number of error messages returned is greater than 0, then PeopleSoft AutoPilot should run the Exit PeopleSoft command.
Warnings	PeopleSoft AutoPilot records the number of EnterpriseOne warning messages that occur during script playback.	Use in conjunction with a conditional statement. For example, set a condition that specifies that if EnterpriseOne sends a warning message when the OK button is clicked, then PeopleSoft AutoPilot should click the OK button twice.
Grid Row Count	PeopleSoft AutoPilot records the number of completed rows in a grid.	Set a repeat count for a node to ensure that PeopleSoft AutoPilot accesses each line in the grid during script playback.

**Note.** The value of the Grid Row Count system variable is determined by the number of rows that EnterpriseOne completes after you click Find, not by the total number of rows in a completed grid. If you want the Grid Row Count value to reflect the actual number of rows in the grid, use the Select Grid Row command to go to the bottom of the grid. After you script this command, the Grid Row Count value is the total number of completed rows in the grid.

## Valid Values Count

If you select the Valid Values Count option as an input source, you also select a created valid value from the value selection list. The list can be stored either on your local drive or on a server. PeopleSoft AutoPilot counts the number of items in the list and stores that number as a variable. You can use the valid values count as a source of input and establish the repeat count for a node by setting the value of the grid row count from the valid values count. The number of times that PeopleSoft AutoPilot repeats the script node matches the number of items in the valid values list, and you can write a command for PeopleSoft AutoPilot to enter the values from the valid values list to a cell in each line of the grid.

## Variable Watch List

The Watch list, which you can select from the PeopleSoft AutoPilot View menu, is a separate form that displays the values of variables in the script during playback. Because the Watch list does not exist in the PeopleSoft AutoPilot form, you can display it at all times, even as you exit and open scripts.

The Watch list contains two column headers: Variable and Value. During script playback, each time that PeopleSoft AutoPilot sets the value of a variable, it adds the variable name and its value to the list. If the variable value that PeopleSoft AutoPilot enters in a header control or grid column is invalid and script playback stops, PeopleSoft AutoPilot stops adding values to the Watch list. Each time that you stop the script and replay it, PeopleSoft AutoPilot clears the Watch list.

## Validation Success

The Validation Success variable enables you to quickly verify the existence of data that you expect as a result of running a script. After you declare a validation, you select Validation Success as an input source, and use the value selection list to select the name of the validation to verify.

After you associate and run the validation and run the script, PeopleSoft AutoPilot uses the Watch list to display the name and value of the validation success variable that you created. A value of 1 indicates that the validation was successful, and a value of 0 indicates that the validation failed. In the rare instance that you declare and associate a validation but do not run the validation, PeopleSoft AutoPilot returns a value of 2 when playback ends.

## UDC Visual Assist Value

You can also use a value from a UDC visual assist as an input source. Selecting the UDC Visual Assist Value option from the Source of Input list populates the UDC Visual Assist Value list with the same UDC values that appear when you click the flashlight icon on the form. When you insert the value in your script in the playback mode, PeopleSoft AutoPilot tests the code that opens the form and selects the value that you specified for your script.

The UDC Visual Assist Value option appears in the Source of Input list only if a header control or grid column in the form contains a visual assist.

## Form Interconnect Visual Assist

You might need to use a visual assist that is not associated with a UDC. For example, suppose that you run a company master search. In EnterpriseOne, when you click the flashlight icon for company master search, a new application appears.

You create this form interconnect in the PeopleSoft AutoPilot script by selecting the Form Interconnect Visual Assist option in the Source of Input list. When you select the Form Interconnect Visual Assist option and insert the command in the script, PeopleSoft AutoPilot tests the code that triggers a series of events in EnterpriseOne. PeopleSoft AutoPilot clicks the visual assist, runs an Application Interconnect command, and confirms the new form. PeopleSoft AutoPilot inserts both the Application Interconnect and Form commands in the Script pane, and you can script any additional commands that you need.



The Form Interconnect Visual Assist option appears in the Source of Input list only when a header control or grid column contains a visual assist that requires an exit to a new application.

---

**Note.** You do not use the value selection list when you select the Form Interconnect Visual Assist option. This source of input requires that you select a value after you exit to the new application.

---

## Clear Source of Input

The Clear command in the Source of Input list enables you to remove an entry to a header control, grid column, or processing option form control. Each time that you select a control or grid column from the command pane and select the Literal Value option from the Source of Input list, PeopleSoft AutoPilot provides a reminder in the value selection list that you can clear the content of the control or column.

## See Also

[Chapter 6, “Scripting Actions,” Changing the Scope of a Variable, page 63](#)

[Chapter 7, “Working with the Script pane,” Using Nodes, page 106](#)

## Using the Value Selection List

After selecting a source of input, you must specify the value or values to enter in the header control, grid column, or QBE line. You can use any of the following methods to supply the value:

- Enter a literal value of numbers, letters, spaces, or a combination of these.
- Select a valid values list that you created.
- Select a variable that you declared and set.
- Select a system variable.
- Select a UDC or form-interconnect visual assist value.

Remember, the caption of the value selection list changes to reflect the option that you select in the Source of Input list.

---

## Scripting the Type To Command

This section provides an overview of scripting the Type To command and discusses how to:

- Use the Header Control or Grid Column list.
- Use a literal value as a source of input.
- Create a list of literal values.
- Create a valid values list from a simple database query.
- Use valid values as a source of input.
- Update the repeat count in a node.
- Declare a variable.
- Change the scope of a variable.
- Set the value of a variable.

- Use the value of a variable as a source of input.
- Update the value of an existing variable.
- Set conditional statements.
- Add a value to a variable.
- Subtract a value from a variable.
- Concatenate a variable.
- Create a variable to confirm validation success.
- Create a variable to store a valid values list count.
- Use a UDC visual assist value as a source of input.
- Use a form interconnect visual assist as a source.
- Clear an input from a header control or grid column.
- Assign a literal value.
- Assign a valid values list value.
- Assign a variable value.
- Assign a form-interconnect visual assist value.
- Type data in a header control.
- Click options in a header.
- Type data in a grid column.
- Assign a UDC visual assist value.
- Type data in a QBE line.

## Understanding Scripting the Type To Command

You use the Header Control, Grid Column, Source of Input, and Value Selection lists to create a Type To command. You create this action command when you create the context commands Header, Grid (or Detail Information), and QBE. When you click the Insert button, PeopleSoft AutoPilot writes the context command and the Type To command and indents the Type To command beneath the context command to reflect the Script pane command hierarchy.

## Using the Header Control or Grid Column List

You begin writing the Type To command when you select Set Header Control Value, Set Grid Cell Value, or Set QBE Cell Value in the Command menu. When you select one of these commands, the Header Control or Grid Column list appears, populated by the controls or columns from the active form. You click the control or column in which you type data.

To use a Header Control or Grid Column list:

1. On the Command menu, select Set Header Control Value, Set Grid Cell Value, or Set QBE Cell Value.
2. In the Header Control or Grid Column list of the command pane, select a header control or grid column in which to enter data.
3. Click the Insert button.

## Using the Source of Input List

After you select a header control or grid column, you must select a source of input for it. Depending on the process that you are testing and the way that you want your script to run, you select from any of four possible sources of input: Literal Value, Valid Values List, Variable, or UDC Visual Assist Value.

After you select a source of input, you select a value from the value selection list, which is documented separately.

## Using a Literal Value as a Source of Input

When you run a script, a literal value appears in the form exactly as you type it in the unpopulated Literal Value field.

To use a literal value as a source of input:

1. In the command pane of the PeopleSoft AutoPilot form, select the Literal option from the Source of Input list.
2. In the Literal Value field, type an input as you would enter it in the header control, grid column, or QBE line of the form.

The entry can be letters, numerals, special characters, spaces, or a combination of these.

3. Click the Insert button.

## Using Valid Values List as a Source of Input

After you create a valid values list, you can use it as a source of input for a header control, grid column, or QBE line. You use a valid values list to enter values multiple times or to run a script multiple times and input a different value each time.

For example, if a valid values list contains five items, you can enter 5 for the repeat count for the node that contains the list. During one script playback, PeopleSoft AutoPilot loops through the node five times and inserts a different item from the list each time. Conversely, you can enter 1 in the repeat count for the node that contains the list but change the repeat count at Begin Script to 5. During each of the five playbacks of the entire script, PeopleSoft AutoPilot inputs a different value from the list.

If you close the script, leave PeopleSoft AutoPilot open, and then open and rerun the script, PeopleSoft AutoPilot inputs the next value in the list in the appropriate context. After you create the valid values list, it contains stored values that you can use as a source of values in subsequent scripts.

You must create a valid values list before you select it as a source of input for a form. To create a list of valid values, select the type of valid values list to create, enter values in the list, and name the list.

## Creating a List of Literal Values

A list of literal values is a list that contains values that you select. Before you create the list, you should verify that the values that you create are valid for the application that you want to test.

To create a list of literal values:

1. On the menu bar of PeopleSoft AutoPilot, click Tools.
2. Select Generate Valid Values List.
3. On the Select Data File Type form, select List of Literal Values and click Next.
4. On the Enter File Name & Date form, type a file name in the File Name field.

5. If you type the name of an existing file, the values in it automatically populate the list.
6. Type one or more values in the Enter Values list, separating them by pressing ENTER. The values should appear vertically stacked in the box.
7. Click Finish.

## Creating a Valid Values List from a Simple Database Query

A simple database query produces a valid values list that contains values that PeopleSoft AutoPilot retrieves from the database and includes in the list, based on the table and column that you select. You can limit the number of records in the list, and you can specify the method that PeopleSoft AutoPilot uses to sort the records, such as in ascending, descending, or random order.

To create a valid values list from a simple database query:

1. On the Tools menu, select Generate Valid Values List.
2. On the Select Data File Type form, select Simple Database Query and click Next.
3. On the Select Table form, double-click a table.
4. On the Column Name form, select a table column and click Next.
5. Specify format options and sort options by clicking the appropriate options.

---

**Note.** To view the contents of the valid values list, click the Preview button. Advanced users can also enter SQL statements.

---

6. Click Preview.
7. Click Next.
8. On the Finish form, assign a file name to the valid values list by typing in the control.
9. Click Finish.

## Using Valid Values As a Source of Input

After you create a valid values list, you return to the Command menu and click the context for writing commands. You select a header control or grid column, and then select the Valid Values List option from the Source of Input list. When you select the Valid Values List option, the name of the list that you created appears in the value selection list.

To use a valid values list as a source of input:

1. On the Header Control or Grid Column list in the command pane of the PeopleSoft AutoPilot form, select a header control or grid column.
2. Select the Valid Values List option in the Source of Input list.
3. Select the name of a valid values list that you have created.
4. Click the Insert button.

## Updating the Repeat Count in a Node

The list of valid values can contain multiple values. To use all the values as inputs for a header control, grid column, or QBE line, change the repeat count in the Form command node of the script to match the number of items in the list. This ensures that PeopleSoft AutoPilot successively types each value in the control, grid column, or QBE line during playback until it exhausts the list.

To update the repeat count in a node:

1. In the Script pane of the PeopleSoft AutoPilot form, click the Form line for the node in which you scripted the list of valid values.
2. In the Repeat Count list in the command pane, type the number of times that you want the node to run.
3. Click the Update button.

## Using a Variable as a Source of Input

To use a variable as a source of input, you must first declare it (that is, give it a name). You can declare the variable at any point in the script. However, if you make the variable global, the value that you assign to it can be used at any subsequent point in the script. After you declare the variable, you can set it (assign a value to it) and store the value for later use in the script in the variable. After you have set the value of the variable, you can change it at any point in the script. If you declare and set the value of more than one variable, you can write conditional statements to compare their values. For example, you might use a conditional statement to verify that a value exists in the database. If the conditional statement shows that the value does not exist, you can modify the script with commands to add the value.

## Declaring a Variable

When you declare a variable, you name it to indicate the place where a value that you set can be stored. You can insert the Declare variable command at any point in the script. Where you declare the variable determines its scope. However, you can change the scope of a variable by dragging it from one point in the script to another.

To declare a variable:

1. On the Command menu, select Variables.

---

**Note.** You can perform this step at any point in the script.

---

2. In the command pane, type a name for the variable in the New Variable field.
3. In the Source of Value list, click Unknown/None.
4. Click the Insert button.

---

**Note.** After you complete these steps, the variable is named but it does not yet have a value assigned to it.

---

## Changing the Scope of a Variable

The scope of a variable is the context within which you can use an assigned value. The scope of the variable can extend locally (to a form or a single application) or globally (throughout the entire script), regardless of how many applications you launch. If you make the variable global, you can select any point in the script to set its value, and you can use the value at any point in the script.

You can declare the variable at any point in the script. To change its scope, move it up or down in the hierarchy of Script pane commands by clicking the Declare command and dragging it to the point that you select.

As you move the mouse pointer, an arrow appears over the Declare command line. An upward-pointing arrow indicates that the command line that you are dragging will be placed above the line that is highlighted when you release the mouse button. A downward-pointing arrow indicates that the command line that you are dragging will be placed below the line that is highlighted when you release the mouse button.

To change the scope of a variable:

1. In the Script pane of the PeopleSoft AutoPilot form, click the Declare command line.
2. Drag the Declare command line to another context by pressing and holding the mouse button.  
To make the variable global, drag the Declare command to the top of the script.
3. When the Declare command line is over the Application command line and the arrow is pointing up, release the mouse button.

## Setting the Value of a Variable

After you have declared the variable, you set its value. You store the value in the declared variable so that you can use it at various points in the script that are determined by the scope of the variable.

To set the value of a variable:

1. On the Command menu, select Variables.
2. In the Existing Variable list of the command pane, select the name of the declared variable to which you want to assign a value.
3. In the Source of Value list, select one of the following sources for the value:
  - Literal Value
  - Valid Values List
  - Variable
  - Header Control Data
  - Grid Cell Data
4. Depending on the source, perform one of the following tasks:
  - If you assign a literal value, type that value into the Literal Value field in the command pane.
  - If you created a list of valid values or declared and set the value of another variable, click the name of one of the values in the list.
  - To derive the value from a header control or grid column, click the name of the control or column that populates the list.

---

**Note.** If you select Grid Cell Data as the source of value, PeopleSoft AutoPilot displays an unpopulated Row Number list and the grid columns for the form in which you are working. If you enter a grid row number in this list and click the Insert button, PeopleSoft AutoPilot stores the value from the row that you specified.

---

5. Click the Insert button.  
PeopleSoft AutoPilot sets the value that you select and stores it in the declared variable.

## Using the Value of a Variable As a Source of Input

After you have declared a variable and set its value, you can use the value as a source of input. The scope that you establish for the variable determines where you can use its value in the script.

To use a variable in the script:

1. In the Header Control or Grid Column list of the command pane, click a header control or grid column of a form to establish the context in which to input the value of the variable.
2. In the Source of Input list, select Variable.
3. In the Variable list, select the name of the declared variable for which you set a value when you began the scripting process.
4. Click the Insert button.

PeopleSoft AutoPilot enters the variable that you set in the header control, grid column, or QBE line.

## Updating the Value of an Existing Variable

After you declare a variable and set its value, you can change the value at any point in the script.

To update the value of an existing variable:

1. In the Script pane of the PeopleSoft AutoPilot form, click the Set command line.  
You assigned a value to the declared variable on this line.
2. In the Variable list of the command pane, select the name of the variable to update.
3. Select an option from the Source of Value list.
4. Select or enter a value.
5. Click the Update button.

## Setting Conditional Statements

To use PeopleSoft AutoPilot to compare the values of two variables, click **If <var> == <var>** in the Command menu. When you do so, the command pane displays three populated lists that enable you to write a conditional If/Then statement, which also includes an Else statement. PeopleSoft AutoPilot populates two of the lists with the names of the variables that you have declared. You write the left and right side of the conditional statement by selecting from each of these lists. To compare the left variable to a literal on the right (rather than a variable), enter a literal value in the Right Literal list.

The third list contains conditional operators such as equal to, not equal to, greater than, and so on. The command pane also includes a check box, for numeric comparison. When you select this option, PeopleSoft AutoPilot converts the text variables to numeric values before it compares them. To write a conditional statement that uses a string, rather than a numeric value, select the Is Not In option from the list of conditional operators.

---

**Note.** You are not required to write any commands as part of the Else branch of the script. You write commands that are part of the Else branch to have PeopleSoft AutoPilot run a series of commands only if the first part of the conditional statement is not true.

---

To set a conditional statement:

1. In the PeopleSoft AutoPilot form, declare and set the value of two variables.

---

**Note.** You can declare the variables and set their values at any point in the script before you write the conditional statement.

---

2. On the Command menu, select `If <var> == <var>`.
3. In the command pane, select options from the following lists:
  - Left Variable
  - Operator
  - Right Variable
  - Right Literal
4. (Optional) If the values of the variables are numeric, select the Numeric check box.
5. Click the Insert button.

PeopleSoft AutoPilot enters the If portion of the conditional statement in the Script pane. The Then and Else portions are blank.
6. Write and insert in the script the commands that constitute the Then branch of the conditional statement.
7. Click the Insert button.
8. To add an Else condition, drag the insertion cursor beneath the Else command line in the Script pane.
9. Write and insert in the script the commands that constitute the Else branch of the conditional statement.
10. Click the Insert button.

You can include branches of script for which the execution depends on the conditional statement.

## Adding a Value to a Variable

You can add to a variable the value of another variable or you can add a literal value. You can add a value to a variable regardless of whether you have merely declared the variable or both declared the variable and set its value. However, if you declare a variable and plan to set its value by adding a value later in the script, enter a default value so that PeopleSoft AutoPilot has an original value to supplement by the addition. To add the value of one variable to another variable, you must first declare and set the value of the variable that contains the value that you want to add.

To add a value to a variable:

1. On the Command menu, select Variables.
2. Select a variable from the Existing Variable list.
3. Select one of the following options from the Source of Value list:
  - Add Variable
  - Add Literal
4. To add a variable value, select the name of a variable from the Variable list.
5. To add a literal value, enter a literal value in the Literal Value field.
6. Click the Insert button.



## Subtracting a Value from a Variable

You can subtract from a variable either the value of another variable or a literal value. You can subtract a value regardless of whether you have set the value for the variable, provided that you set a default value when you declare the variable.

To subtract a value from a variable:

1. On the Command menu, select Variables.
2. Select a variable from the Existing Variable list.
3. Select one of the following from the Source of Value list:
  - Subtract Variable
  - Subtract Literal
4. To subtract a variable value, select the name of a variable from the Variable list.
5. To subtract a literal value, enter a literal value in the Literal Value field.
6. Click the Insert button.

## Concatenating a Variable

Concatenating a variable enables you to create alphanumeric strings from multiple variables. You can construct a concatenated variable from other variables or from literal values. The following table illustrates the principle of variable concatenation:

Variable Name	Variable Value
X	1
Y	1
X concatenated with Y	11

---

**Note.** Before concatenating a variable, declare at least one variable, and either set its value or enter a default value.

---

To concatenate a variable:

1. On the Command menu, select Variables.
2. Select a variable from the Existing Variable list.
3. Select one of the following from the Source of Value list:
  - Concatenate Literal
  - Concatenate Variable
4. To concatenate a literal value, enter a literal value in the Literal Value field.
5. To concatenate a variable, select a variable from the Variable list.
6. Click the Insert button.

PeopleSoft AutoPilot creates a concatenated value for the variable selected from the Existing Variable list.

## Creating a Variable to Confirm Validation Success

To confirm the success or failure of a validation, you declare a variable and select Validation Success as a source of value. When you select the Validation Success option, PeopleSoft AutoPilot displays the names of the declared validations in the value selection list. When you run the validation, PeopleSoft AutoPilot sets the value of the variable to 1 if the validation succeeds. If the validation fails, PeopleSoft AutoPilot sets the value of the variable to 0. If you declare the validation and assign values but do not run it, PeopleSoft AutoPilot sets the value of the validation variable to 2.

Select the Watch List option on the PeopleSoft AutoPilot View menu to determine whether the data validation is successful following playback. The Watch list displays the name of the validation variable in the Variable column. If the validation is successful, PeopleSoft AutoPilot displays 1 in the Value column.

---

**Note.** Before creating a variable to confirm validation success, declare and associate a validation. You must run the validation to have PeopleSoft AutoPilot indicate success or failure by returning a value of 1 or 0.

---

See [Chapter 6, “Scripting Actions,” Using the Database Validation Command, page 93](#).

1. On the Command menu, select Variables.
2. Enter the name of a variable in the New Variable list, or select the name of a variable from the Existing Variable list.
3. Select Validation Success (0/1) from the Source of Value list.
4. Select the name of a validation from the value selection list.
5. Click the Insert button.

If the validation runs successfully, PeopleSoft AutoPilot displays a value of 1 for the validation variable in the Watch list.

## Creating a Variable to Store a Valid Values List Count

To capture and store the value that equals the number of items in a list of valid values, you select the Valid Values Count option as a value source for a new or existing variable. After you select this source of value, you select a valid values list. PeopleSoft AutoPilot stores the number of items in the list as the value for the variable.

To use the valid values count as a source of value, you must create a valid values list, either by generating a simple database query or by creating a list of literal values.

To create a variable to store a valid values list count:

1. On the Command menu, select Variables.
2. Enter the name of a variable in the New Variable list, or select the name of a variable from the Existing Variable list.
3. Select the Valid Values Count option from the Source of Value list.
4. Select the name of a valid values list from the value selection list.
5. Click the Insert button.

The Watch list displays the number of items in the valid values list as the value of the variable.

6. To use the valid values count to set the repeat count for a node, select the node in the Script pane that contains the valid values list used to determine the value of the Valid Values Count variable.
7. In the Define Repeat Count list, select Variable.
8. In the Repeat Count list, select the variable that stores the valid values count value.

- Click the Update button.

PeopleSoft AutoPilot updates the repeat count value, which now matches the item number value in the valid values list. When you run the script, PeopleSoft AutoPilot enters a value from the valid values list in a grid cell, one grid row at a time, until it has used each value in the valid values list.

## Using a UDC Visual Assist Value as a Source of Input

Some header controls, grid columns, and QBE lines in forms contain UDC visual assists. When you select the UDC Visual Assist Value option as a source of input, the UDCs in the value selection list in the command pane correspond to the codes in the visual assist forms.

The flashlight icon identifies the UDC visual assist, but PeopleSoft EnterpriseOne software also identifies other visual assists in this way. The UDC Visual Assist Value options appears in the Source of Input list only if a header control or grid column in the active form has a UDC visual assist.

To use a UDC visual assist value as an input source:

- On the Command menu, select Set Header Control Value or Set Grid Cell Value.
- In the command pane of the PeopleSoft AutoPilot form, select a header control or grid column from the Header Control or Grid Column list.

---

**Note.** These values correspond to the UDC values that appear when you click the visual assist icon for a control or column in a form. If the selection does not have a visual assist icon, the UDC Visual Assist Value option does not appear in the Source of Input list.

---

- Select the UDC Visual Assist Value option from the Source of Input list.
- In the UDC Visual Assist Value list, select a UDC value.
- Click the Insert button.

PeopleSoft AutoPilot runs the code path and inserts the UDC value in the script.

## Using a Form-Interconnect Visual Assist as a Source of Input

Some header controls and grid columns contain visual assists that require you to exit the current application and access a new one. If you have selected a header control or grid column that includes this type of visual assist—such as Address Book Master Search—PeopleSoft AutoPilot displays the Form Interconnect Visual Assist option in the Source of Input list. If you select this input source, PeopleSoft AutoPilot clicks the flashlight icon in the form and writes an Application Interconnect command and Form command. You can then write any additional commands that you need.

---

**Note.** You do not use the value selection list when you select this option. After you access a new application and form, you can write the additional commands that you need as part of your script.

---

To use a form-interconnect visual assist as a source of input:

- On the Command menu, select either Set Header Control Value or Set Grid Cell Value.
- In the command pane of the PeopleSoft AutoPilot form, select a header control or grid column from the Header Control or Grid Column list.
- Select the Form Interconnect Visual Assist option from the Source of Input list.
- Click the Insert button.

PeopleSoft AutoPilot automatically writes a command to click the flashlight icon in the active form, and it then runs an Application Interconnect command and a Form command.

## Clearing an Input from a Header Control or Grid Column

You can clear the contents of a header control or grid column. PeopleSoft AutoPilot enables you to do this by selecting the Clear option from the Source of Input list in the command pane.

To clear an input from a header control or grid column:

1. On the Command menu, select Set Header Control Value, Set Grid Cell Value, or Set QBE Cell Value.
2. In the PeopleSoft AutoPilot command pane, select the name of a header control or grid column from the Header Control or Grid Column list.
3. In the Source of Input list, select Clear.
4. Click the Insert button.

PeopleSoft AutoPilot clears the value from the selected header control or grid column.

### See Also

Chapter 6, “Scripting Actions,” Using a Valid Values List, page 53

## Using the Value Selection List

After you select a header control or grid column and an input source for it, you complete the Type To command by selecting a value from the value selection list. PeopleSoft AutoPilot displays this list when you select a source of input and provides a caption according to the source of input that you select. You can also delete an entry in a control in a header or a processing option form or a grid column. To do this, select Clear from the Source of Input list.

To use the value selection list:

1. To input a literal value in the header control or grid column, type that value in the unpopulated value selection list.
2. To input the values that you assigned to a valid values list or variable, select the name of the list or variable from the value selection list.
3. To input a UDC visual assist value, you select one from the value selection list.

## Assigning a Literal Value

The literal value that you assign as an input in a header control or grid column can be numbers, letters, special characters, or a combination of these. Verify that the literal value that you assign is a valid input.

To assign a literal value:

1. In the command pane of the PeopleSoft AutoPilot form, select a header control or grid column name from the Header Control or Grid Column list.
2. Select the Literal Value option from the Source of Input list.
3. In the value selection list labeled Literal Value, input the value assigned to the control or grid column.
4. Click the Insert button.

## Assigning a Valid Values List Value

For input for a header control or grid column, you can assign a value from a valid values list. PeopleSoft AutoPilot selects the first value in the list the first time that you run the script. If you run it more than once, PeopleSoft AutoPilot selects the second value in the list on the second loop. This pattern continues until the loop ends or the item list is exhausted.

To assign a valid values list value:

1. In the command pane of the PeopleSoft AutoPilot form, select a header control or grid column name from the Header Control or Grid Column list.
2. Select the Valid Values List option from the Source of Input list.
3. In the value selection list labeled Valid Values List, select the name of a valid values list that you created.
4. Click the Insert button.

## Assigning a Variable Value

You can declare a variable, set its value, and assign that value to a header control or grid column. After you set the value, PeopleSoft AutoPilot stores it and it is available for use at any point in the script.

To assign a variable value:

1. In the command pane of the PeopleSoft AutoPilot form, select a header control or grid column name from the Header Control or Grid Column list.
2. Select the Variable option from the Source of Input list.
3. In the value selection list labeled Variable, select the name of a declared variable to set a value.
4. Click the Insert button.

To assign a UDC visual assist value:

1. In the command pane of the PeopleSoft AutoPilot form, select a header control or grid column name from the Header Control or Grid Column list.
2. Select the UDC Visual Assist Value option from the Source of Input list.
3. In the value selection list labeled UDC Visual Assist Value, select the name of a UDC value.
4. Click the Insert button.

## Assigning a Form Interconnect Visual Assist Value

PeopleSoft AutoPilot enables you to select a form interconnect visual assist value for header controls or grid columns in forms that contain a visual assist that accesses a different application. You do not use the value selection list when you select the Form Interconnect Visual Assist option as a source of input. Instead, after you access the other application, you script an input in a header control or grid column in the active form by using the Press Toolbar Button command to find and select a value.

To assign a form interconnect visual assist value:

1. In the command pane of the PeopleSoft AutoPilot form, select a header control or grid column name from the Header Control or Grid Column list.
2. Select the Form Interconnect Visual Assist option from the Source of Input list.
3. Click the Insert button.

PeopleSoft AutoPilot clicks the visual assist button on the selected header control or grid column, and it then writes an Application Interconnect command and a Form command to the Script pane.

4. On the Command menu, select Set QBE Cell Value.
5. In the command pane, select the name of a grid column from the Grid Column list.
6. Select an input source from the Source of Input list.
7. Enter or select a value from the value selection list.
8. Click the Insert button.
9. On the Command menu, select the Press Toolbar Button option.
10. In the command pane, select the Standard Button option.
11. Select Find.
12. Click the Insert button.
13. On the Command menu, select the Press Toolbar Button option.
14. In the command pane, select the Standard Button option.
15. Click Select.
16. Click the Insert button.

PeopleSoft AutoPilot enters the value from the form interconnect visual assist in the header control of the selected form.

17. On the Command menu, select the Press Toolbar Button option.
18. In the command pane, select the Standard Button option.
19. Click Close or Cancel.
20. Click the Insert button.
21. On the Command menu, select Form.
22. In the command pane, select the name of the form that PeopleSoft AutoPilot exited by using the form interconnect visual assist.

---

**Note.** This command confirms that PeopleSoft AutoPilot has returned to the previous form. If you do not confirm the form, you cannot continue scripting.

---

23. Click the Insert button.

## Scripting the Type To Command

After you select from each of the three command pane lists, you click the Insert button to script the Type To command. PeopleSoft AutoPilot uses the selected information to write two command lines in the Script pane. One command line contains the context—header, grid, or QBE—and the repeat count for the node. The other contains the name of the selected header control or grid column; a symbol that indicates whether you selected a literal value, a valid values list, a variable, or a UDC visual assist value as the input source; and the assigned value.

## Typing Data in a Header Control

You use the command pane lists to script inputs for the header portion of a form. Selecting the Set Header Control Value option in the Command menu establishes the header as the context in which you type data. The options that you select from the lists in the command pane create the Type To action command. When you click the Insert button, the command line that specifies the header control as the context appears as a node. The command line identifies the control that you select and the value that you type in it. PeopleSoft AutoPilot inserts subsequent Type To commands subordinate to the node.

To type data in a header control:

1. On the Command menu, select the Set Header Control Value option.
2. Select the name of a control from the Header Control list.

When PeopleSoft EnterpriseOne software is running, a BlueCue highlights the control in the form that corresponds to the control that you selected in PeopleSoft AutoPilot.

3. Select an input source from the Source of Input list.
4. In the value selection list, enter a literal value or select the name of a valid values list, variable, or UDC visual assist value.
5. Click the Insert button.
6. Script inputs to additional header controls by selecting the Set Header Control Value option in the Command menu and repeating steps 1-5.

## Selecting Options in a Header

Some forms contain options that you can select. You select the Check Box/Radio Button option in the Command menu to write commands for these options in a script. The command to select an option is a different command than the Type To command, which you use to type data in header controls, grid columns, or QBE lines. However, when you work with a form that contains these options in its header, PeopleSoft AutoPilot inserts the command in the Header node along with any Type To commands that you write.

To script selecting options in a header:

1. On the Command menu, select the Check Box/Radio Button option.
2. In the command pane, select an option from the Radio Button or Check Box list.
3. If the option is a check box, select the Check or Uncheck option in the Source of Input list.

If the option is a radio button, the Source of Input list is unpopulated. PeopleSoft AutoPilot selects the radio button when you insert the command.

## Typing Data in a Grid Column

You use the command pane lists to script inputs in the grid area of a form. Selecting the Set Grid Cell Value option in the Command menu establishes the grid area as the context in which you type data. The options that you select from the lists in the command pane create the Type To action command. When you click the Insert button, the command line that specifies the grid detail area as the context appears as a node. The Type To command is indented beneath and attached to the node. The command line identifies the selected column and the value that you typed in it. PeopleSoft AutoPilot inserts subsequent Type To commands subordinate to the node. You can script different inputs to multiple rows of the grid, or you can use a playback loop in PeopleSoft AutoPilot to script the input in one row multiple times.

To type grid column inputs:

1. On the Command menu, select the Set Grid Cell Value option.
2. In the command pane, select a grid column from the Grid Column list.  
A BlueCue appears in the appropriate grid column in the form.
3. Select an input source.
4. Enter a literal value or select the name of a valid values list, variable, or UDC visual assist value.
5. Click the Insert button.
6. Script inputs for additional grid columns by clicking the name of another column in the Grid Column list and repeating steps 1-5.

To type inputs to additional grid rows:

1. In the Script pane in PeopleSoft AutoPilot, click a Detail Information command line that you inserted in the script.  
The insertion cursor is connected to the selected node.
2. On the Command menu, select the Set Grid Cell Value option.
3. In the command pane, select a grid column from the Grid Column list.
4. Select an input source.
5. Enter a literal value or select the name of a valid values list, variable, or UDC visual assist value.
6. Click the Insert button.
7. Repeat steps 1-6 each time that you want to script commands in a new row.

To script a playback loop:

1. Follow the steps for typing inputs in a grid row.
2. (Optional) Deactivate the Playback button in the toolbar.
3. In the Script pane, click the Detail Information command line that you want to play multiple times.
4. In the command pane, select an input source, such as literal value or variable, in the Define Repeat Count From list.
5. Enter a literal value or select the name of a variable that you created.

The value specifies the number of times that you want the inputs to loop (that is, to be entered in successive grid rows).

6. Click the Update button.
7. (Optional) Click the Playback button.

During script playback, PeopleSoft AutoPilot enters the inputs for a single row of the grid as many times as you specified in the Repeat Count list. For example, if the repeat count for row 1 is 3, the system completes rows 1 through 3 with the inputs that you originally scripted for row 1.

## Assigning a UDC Visual Assist Value

PeopleSoft AutoPilot enables you to select a UDC value for those header controls or grid columns in forms that contain UDC visual assists. The value selection form displays the valid UDC values for the selected control or column. If the control or column does not contain a UDC value, the UDC Visual Assist Value option does not appear in the Source of Input list.



## Typing Data in a QBE Line

You can script commands to enter data in the QBE line of a form that has a QBE line. Selecting the Set QBE Cell Value option in the Command menu establishes the QBE line as the context in which you type data. The options that you select from the lists in the command pane create the Type To action command. When you click the Insert button, the command line specifying the QBE line of the grid as the context appears as a node. The Type To command is indented beneath and attached to the node. The command line identifies the selected grid column and the value that you typed in it. PeopleSoft AutoPilot inserts subsequent Type To commands subordinate to the node. You can script different inputs in multiple rows of the grid, or you can script the input in one row multiple times, using a playback loop in PeopleSoft AutoPilot.

To type data in a QBE line:

1. On the Command menu, select the Set QBE Cell Value option.
2. In the Grid Column list in the command pane, select the name of a grid column.
3. Select an input source.
4. Enter a literal value, or the name of a valid values list, variable, or UDC visual assist value.
5. Click the Insert button.
6. To script an input in the QBE line of another grid column, select another name in the Grid Column list and repeat steps 1-5.
7. On the Command menu, select the Press Toolbar Button option.
8. Select the Press Standard Button option.
9. Select Find.
10. Click the Insert button.

---

## Using the Select Grid Row Command

This section provides an overview the Select Grid Row command and discusses how to:

- Use the Operation Type list.
- Use the Action on Grid Row list.
- Use the Grid Columns list.
- Use the Source of Row Number list.

## Understanding the Select Grid Row Command

The Select Grid Row command enables you to perform and test several important functions. You use it to work within a detail area of a form. You can select records, delete records, add to a grid row, or edit the entries in a grid row. You can also script an exit to another form in another application. You use this command in conjunction with several other action and context commands, including the Press Toolbar Button command, the Application Interconnect command, and the Grid Data command.

PeopleSoft AutoPilot enables you to click either a row or a grid cell, specify the row number or grid column, and perform a specific action, such as double-clicking the row or editing the content of the cell.

---

**Note.** When you select the Select Grid Row command, PeopleSoft AutoPilot also populates the command pane with a value selection list that enables you to enter a literal value or select the name of a previously created valid values list or variable. The caption of this list reflects the option that you select in the Source of Row Number list.

---

## Using the Operation Type List

The Operation Type list in the PeopleSoft AutoPilot command pane enables you to select a row in a grid area either by specifying a row number or by specifying the value of a particular cell in a grid column.

### Click by Row Number Option

The Click by Row Number option enables you to select a grid row by number. Because you specify a single number, this option works particularly well for grid areas with a large number of rows. PeopleSoft AutoPilot finds the designated row and performs the action selected in the Action on Grid Row list. You can designate the grid row by entering a literal value, a value from a valid values list, or a value from a variable in the value selection list.

### Click by Cell Content Option

To select a grid row that contains a particular value, such as an item number, use the Click by Cell Content option and select a grid column and an action that you want to perform on the cell. PeopleSoft AutoPilot selects the grid cell rather than the entire grid row.

## Using the Action on Grid Row List

The Action on Grid Row list enables you to specify the purpose for selecting the row. In PeopleSoft AutoPilot, you can script the following types of grid row operations:

- Single-click a grid row.
- Single-click a grid row button.
- Double-click a grid row.
- Double-click a grid row button.
- Position grid row for add or edit.

### Single-Click a Grid Row

You write a command to single-click a grid row when the form that is active does not have a row button. Although you can write this command when the active form has a row button, it is not recommended because single-clicking the grid row sometimes selects only a cell.

### Single-Click a Grid Row Button

You write a command to single-click the grid row button when the form that is active contains rows with buttons. Clicking the grid row button selects the row, rather than a cell. Do not script this grid row operation in forms that do not contain a button.

### Double-Click a Grid Row

To move from the detail area of one form to another form, exit to a new application, or double-click a row to perform a process, you write a command to double-click the grid row.

## Double-Click a Grid Row Button

You write a command to double-click the grid row button when the form that is active contains grid rows with buttons. Do not script this grid row operation in forms that do not contain grid row buttons.

## Position Grid Row for Add/Edit

To enter changes in the detail area of a form, you write a command to position the grid row for an add or an edit. You use the command to add an entry to a grid row or to edit an existing value in a grid cell.

## Using the Grid Columns List

You use the Grid Columns list only when you have selected the Click by Cell Content option. When you select this option, PeopleSoft AutoPilot populates the Grid Columns list with the names of all the columns in the grid that is active. You can scroll through this list to find the name of a column.

After you select Click by Cell Content, you can use the Source of Row Number list to specify a value that might exist in a particular cell of that column. PeopleSoft AutoPilot searches for the value and selects the first row that contains that value in the specified column.

## Using the Source of Row Number List

In the Source of Row Number list, you select the source of the value that you use to select the grid row. You can select the Literal Value option, in which case you type a row number or a grid cell value in the value selection list. You can select the Valid Values List option, in which case you select from the value selection list an existing valid values list. PeopleSoft AutoPilot uses the first value in the list to select a grid row either by row number or by cell content. You can select Variable, in which case you select from the value selection list a variable for which you have declared a name and set a value. PeopleSoft AutoPilot uses this value to select a grid row either by row number or by cell content.

---

# Scripting the Select Grid Row Command

This section provides an overview of scripting the Select Grid Row command and discusses how to:

- Click by row number
- Click by cell content
- Perform grid row operations

## Understanding Scripting the Select Grid Row Command

To begin the scripting, open a form. Populate the detail area by writing a command to click Find. When the grid is populated, use the PeopleSoft AutoPilot command pane to select the row that you want to target for the script and the operation that you want to perform on that row. You can select a grid row either by row number or by cell content.

In either case, you select the type of operation that you want to perform on the grid row. Operations include single-clicking the row, single-clicking the row button, single-clicking to perform an add or edit, double-clicking the row, or double-clicking the row button.

To finish scripting the Select Grid Row command, select a value source for row selection. Possible sources include a literal value, a valid values list, or a variable. You then either enter a literal value in the value selection list or use the value selection list to select a valid values list or variable.

## Clicking by Row Number

After you populate the grid, you can use PeopleSoft AutoPilot to select a row by searching for a row number that you designate. To complete this command, you must also select an action that you want PeopleSoft AutoPilot to perform on the grid row, select a source of value that PeopleSoft AutoPilot uses to select the row, and select or enter the value of the row.

To click a grid row by row number:

1. On the Command menu, select the Press Toolbar Button option.
2. Select the Press Standard Button option.
3. Click Find.  
PeopleSoft AutoPilot fills the detail area in the active form.
4. On the Command menu, select the Select Grid Row option.
5. In the PeopleSoft AutoPilot command pane, select the Click by Row Number from the Operation Type list.
6. Select a grid row action from the Action on Grid Row list.
7. Select a value source from the Source of Row Number list.
8. In the value selection list, enter a literal value or select the name of a valid values list or variable.
9. Click the Insert button.

## Clicking by Cell Content

You can have PeopleSoft AutoPilot search the detail area for a particular value, and then select the row after it finds a cell that contains that value. In this case, you can write a command to click the grid row using cell content rather than row number.

Writing a command to select a grid row based on cell content involves the same steps that you use to write a command to select the row based on row number. However, when you select a row based on cell content, you must also select a grid column as a search criterion.

To click a grid row by cell content:

1. On the Command menu, select the Press Toolbar Button option.
2. Select the Press Standard Button option.
3. Click Find.  
PeopleSoft AutoPilot fills the detail area in the active form.
4. On the Command menu, select the Select Grid Row option.
5. In the PeopleSoft AutoPilot command pane, select the Click by Cell Content from the Operation Type list.
6. Select a grid row action from the Action on Grid Row list.
7. In the Grid Columns list, select a grid column in which you want PeopleSoft AutoPilot to search for a value.
8. Select a value source from the Source of Row Number list.
9. In the value selection list, enter a literal value or select the name of a valid values list or variable.

10. Click the Insert button.

## Performing Grid Row Operations

After you select a row, you can perform the following operations on it:

- Single-click a grid row.
- Single-click a grid row button.
- Double-click a grid row.
- Double-click a grid row button.
- Position a grid row for add or edit.

---

**Note.** Select the single-click and double-click grid row operations when you are writing commands to test a form that contains a detail area that does not have row buttons. You can use these options before you write other PeopleSoft AutoPilot commands. For example, single-clicking on a grid row or grid row button enables you to write a command to click the Select or Delete button. Double-clicking a row enables you to access another form or application. Clicking on a row and positioning for add or edit enables you to edit the selected grid row

---

To single-click a grid row:

1. On the Command menu, select the Select Grid Row option.
2. Select an operation from the Operation Type list.
3. Select the single-click option in the Action on Grid Row list.
4. Select a value source from the Source of Row Number list.
5. In the value selection list, enter a literal value or select a valid values list or variable.
6. Click the Insert button.

To double-click a grid row:

1. On the Command menu, select the Select Grid Row option.
2. Select an operation from the Operation Type list.
3. Select the double-click option in the Action on Grid Row list.
4. Select a value source from the Source of Row Number list.
5. In the value selection list, enter a literal value or select a valid values list or variable.
6. Click the Insert button.
7. On the Command menu, select either the Form option or the Application Interconnect option.
8. In the command pane, select the next form or application that appears when you double-click the row in the form.
9. Click the Insert button.

To single-click a grid row button:

1. On the Command menu, select the Select Grid Row option.
2. Select an operation from the Operation Type list.
3. Select the single-click row button option in the Action on Grid Row list.

4. Select a value source from the Source of Row Number list.
5. In the value selection list, enter a literal value or select a valid values list or variable.
6. Click the Insert button.

To double-click a grid row button:

1. On the Command menu, select the Select Grid Row option.
2. Select an operation from the Operation Type list.
3. Select the double-click row button option in the Action on Grid Row list.
4. Select a value source from the Source of Row Number list.
5. In the value selection list, enter a literal value or select a valid values list or variable.
6. Click the Insert button.
7. On the Command menu, select either the Form option or the Application Interconnect option.
8. In the PeopleSoft AutoPilot command pane, select the next form or application that appears when you double-click the row button on the form.
9. Click the Insert button.

To position a grid row add or edit:

1. On the Command menu, select the Select Grid Row option.
2. Select an operation from the Operation Type list.
3. Select the Position for Add/Edit option in the Action on Grid Row list.
4. Select a value source from the Source of Row Number list.
5. In the value selection list, enter a literal value or select a valid values list or variable.

---

**Note.** If you are adding a row, enter a high row number, such as 999, to ensure that you reach the bottom of the grid when PeopleSoft AutoPilot inserts the command.

---

6. In the Command menu, select the Set Grid Cell Value option.
7. Select a grid column in the Grid Column pane.
8. Select an input source in the Source of Input pane.
9. Enter a literal value or UDC value, or select a valid values list or a variable from the value selection list.
10. Click the Insert button.

PeopleSoft AutoPilot adds to or edits the column in the selected grid row.

---

## Using the Press Toolbar Button Command

This section provides an overview of the Press Toolbar Button command and discusses how to:

- Use the Standard Button option.
- Use the Custom Button option.
- Use the Select Grid Tab option.

- Use the Grid Scroll Button option.

## Understanding the Press Toolbar Button Command

The Press Toolbar Button command enables you to script many of the important actions that you can perform in a form, such as:

- Moving from one form to another within the same application.
- Moving to a form in a new application, using either the Form menu or the Row menu.
- Populating a grid on a form.
- Selecting a row in a grid.
- Updating the database.
- Deleting a record.
- Exiting a form.
- Select a grid tab.
- Clicking the grid scroll button.
- Submitting UBEs that are not hard-coded to run automatically

## Using the Standard Button Option

The Standard Button option in PeopleSoft AutoPilot contains button-clicking selections that match the buttons in the toolbars of forms. For example, the toolbar in some forms features 10 button-clicking options. When one of these forms is active, the Standard Button tree in the Button list in PeopleSoft AutoPilot contains the same options.

When you script one of these button-clicking options, PeopleSoft AutoPilot runs the command exactly as it would be run in a PeopleSoft EnterpriseOne form. For example, you might script clicking the OK button to update the database after you have entered new data on a form.

When you select the Standard Button option, the Next Form list also appears. The option that you select from this list indicates the form that is active when you click the Insert button.

---

**Note.** Clicking a standard button, such as Add, occasionally takes you to another application. In this case, you must write an Application Interconnect command instead of completing the Next Form field.

---

When you launch a UBE, the standard button options in PeopleSoft AutoPilot also match the buttons in the menu bar of the form. For example, when you need to submit a UBE using the Version Prompting form, select Submit from the Standard Button options in PeopleSoft AutoPilot.

### See Also

Chapter 5, “Writing Scripts,” Submitting a UBE, page 41

## Using the Custom Button Option

Use the Custom Button option to script a selection from the Row menu or the Form menu. Selecting from the Row menu or the Form menu usually results in accessing a different application, in which case you must also script an Application Interconnect.

Although you use both the Custom Button option and the Application Interconnect command to interconnect applications, they perform slightly different functions. You script an Application Interconnect command *after* you have exited to a new application—for example, by clicking the Add button. You can use the Custom Button option *before* you exit to a new application. The Custom Button option enables you to select the application and form in the command pane and insert the commands. PeopleSoft AutoPilot interconnects the applications, and the form in the new application appears.

When you select the Custom Button option from the Button list in the command pane, the following options appear in the tree:

- Form
- Row

Selecting either or both of these options further expands the tree and displays the available menu options.

---

**Note.** The options in the Form and Row menus correspond to the options that appear in the drop-down menu when you click Form or Row in the menu bar of the active form. When you insert a command to click a standard button and to click a custom button, the same symbol appears in the command lines of the script. However, the command line for clicking a standard button describes the command as the Press Toolbar Button command, whereas the command line for clicking a custom button describes the command as the Select Menu Exit command. When you click a standard button, you usually access forms in the same application. When you click a custom button, you usually access a new application.

The command line for clicking a custom button contains the type of menu exit, either Form or Row, and the name of the menu item that you select. The Select Menu Exit command line should be followed by the Application Interconnect command line, which records the application that you access.

---

## Form Exit

PeopleSoft AutoPilot enables you to script a selection from the Form menu, just as you might select a menu option in a PeopleSoft EnterpriseOne form. Typically, you perform a form exit when you want to access a form that is related to the current form. However, the new form might exist in a different application, and the exit represents a change in the standard sequence of forms that you access when you perform a transaction.

## Row Exit

PeopleSoft AutoPilot enables you to script a selection from the Row menu, just as you might select an directly in a PeopleSoft EnterpriseOne form. Typically, you perform a row exit when you want to move from a particular grid row in a form to a related form. That form might be in a different application, and the exit represents a change in the standard sequence of forms that appear when you perform a transaction.

You can also use the Custom Button option to perform a row exit that launches a UBE version. In this case, you write a UBE command after you script the row exit, and PeopleSoft AutoPilot automatically submits the UBE.

## See Also

Chapter 5, “Writing Scripts,” Launching a UBE from a Row Menu, page 39



## Using the Select Grid Tab Option

You use the Select Grid Tab option in PeopleSoft AutoPilot to test whether the system accesses customized grid tabs that you have created. In PeopleSoft EnterpriseOne software, you customize the grid by selecting Preferences, Grid, New Format. To create a new tab, you select fonts, the number of grid columns, the width of grid columns, and so on. Each grid configuration creates a new tab. After you create as many custom tabs as you need, you can use PeopleSoft AutoPilot to script the selection of tabs. You determine whether the tab is selected, and you can also determine whether the customized changes appear.

## Using the Grid Scroll Button Option

You use the Grid Scroll Button option in PeopleSoft AutoPilot to script clicking the up and down arrows in the detail area of a form. PeopleSoft AutoPilot moves the arrows up or down by line or by page.

---

## Scripting the Press Toolbar Button Command

This section provides an overview of scripting the Press Toolbar Button command and discusses how to:

- Click a standard button.
- Click a custom button.
- Select a grid tab.
- Click the grid scroll button.

## Understanding Scripting the Press Toolbar Button Command

You use the Press Toolbar Button command in PeopleSoft AutoPilot to script many important functions. For example, clicking the Add button enables you to access a new form, either in the same application or a new one. By selecting the Standard Button option for the Press Toolbar Button command, you can write a script command to click the Add button. Other Press Toolbar Button options enable you to perform form and row exits, select a grid tab, or scroll through a grid.

## Clicking a Standard Button

In general, you select the Standard Button option to click a toolbar button in a form. When you select this option, the options in the command pane match the toolbar buttons on the active form.

To move from one form to another using the Add button:

1. On the Command menu, select the Press Toolbar Button command.
2. Select the Standard Button option, and then select Add.
3. Select the name of a form from the Next Form list or select Unknown/None for another application.

---

**Important!** If you select the Unknown/None option and then click the Insert button, ensure that the Form command line in PeopleSoft AutoPilot matches the active form. On the Command menu, select Form, select the name of the active form, and then click the Insert button.

---

4. Click the Insert button.

The next form appears.

If you select the Unknown/None option and the form that appears is part of a different application, complete steps 5 through 8 to script an Application Interconnect command.

5. On the Command menu, select the Application Interconnect option.
6. From the Application list in the command pane, select the name of the active application.
7. From the Form list in the command pane, select the name of the active form.
8. Click the Insert button.

To move from one form to another by using the Select button:

1. On the Command menu, select the Select Grid Row option.
2. In the Source of Row Number list in the command pane, select literal value, valid values list, or variable.
3. To specify the row number to select, enter or select a value from the value selection list.
4. Click the Insert button.
5. On the Command menu, select the Press Toolbar Button option.
6. Select the Standard Button option, and then select the Select option.
7. In the Next Form list, select the form that appears next.
8. Click the Insert button.

---

**Note.** If you select the Unknown/None option from the Next Form list, use the Form command to confirm the new form after it appears. If the form that appears is part of a different application, you must script an Application Interconnect command.

---

To update the database:

1. On the Command menu, select the Press Toolbar Button option.
2. Select the Standard Button option, and then click OK.
3. If you are accessing a new form, select it from the Next Form list; if not, do not make a selection from the list.
4. Click the Insert button.

To fill a grid:

1. On the Command menu, select the Press Toolbar Button option.
2. Select the Standard Button option, and then select Find.
3. Click the Insert button.

To delete a record:

1. On the Command menu, select the Select Grid Row option.
2. In the Source of Row Number list in the command pane, select literal value, valid values list, or variable.
3. To specify the row number to be selected, enter or select a value from the value selection list.  
The row number should contain the record that you want to delete.
4. Click the Insert button.
5. Select the Press Toolbar Button option from the Command menu.
6. Select the Standard Button option, and then select Delete.

7. Click the Insert button.

To exit a form:

1. On the Command menu, select the Press Toolbar Button option.
2. Select the Standard button option, and then select either Cancel or Close, depending on the button that is available on the form.
3. Click the Insert button.

## See Also

[Chapter 5, “Writing Scripts,” Launching a UBE from a Row Menu, page 39](#)

[Chapter 5, “Writing Scripts,” Launching a UBE from a Row Menu, page 39](#)

## Clicking a Custom Button

You use the Custom Button option to script selections from the Row menu and the Form menu. When you select this option, the selections in the command pane match the form and row exits in the active form.

Remember that a form or row exit might result in an application interconnect. To script the application interconnect, use the command pane to select the menu option, the next form, and the application before you click the Insert button.

To script a form exit:

1. On the Command menu, select the Press Toolbar Button option.
2. Select the Custom Button option, and then select Form.
3. Select a form exit.
4. If the form exit results in the system launching a new application, select the new application from the Application list.
5. From the Next Form list, select a form, if necessary.
6. Click the Insert button.

To script a row exit:

1. On the Command menu, select the Press Toolbar Button option.
2. If the detail area in the active software form is empty, select Find.
3. Click the Insert button.

In the application form, the detail area loads.

4. On the Command menu, select the Select Grid Row option.

---

**Note.** To display the new form, you can select Row from the menu bar in PeopleSoft EnterpriseOne, click one of the forms in the list, or click Select.

You can determine the name of the newly active application and form by clicking the About PeopleSoft option in the menu bar.

---

5. Select the single-click row button from the Actions on Grid Row pane.
6. In the Source of Row Number list in the command pane, select a literal value, valid values list, or variable.

7. To specify the row number to be selected, enter or select a value from the value selection list.
8. Click the Insert button.
9. On the Command menu, select the Press Toolbar Button option.
10. From the Button list in the PeopleSoft AutoPilot command pane, select the Custom Button option.
11. Click Row.
12. Select a Row exit.

---

**Note.** If you run a row exit to launch a UBE, you do not make a selection from the Application or Next Form list that appears. You select the row exit only, click the Insert button, and then write a UBE command.

---

13. If you are exiting to an interactive application, select that application from the Application list.
14. Select a form name from the Next Form list.
15. Click the Insert button.

## Selecting a Grid Tab

You can select customized grid tabs by using the Select Grid Tab command. PeopleSoft AutoPilot selects the grid tab number that you script and, in playback mode, displays the grid with your changes.

To script the Select Grid Tab option:

1. In a form, create as many tabs as you need.
2. On the Command menu, select the Press Toolbar Button option.
3. In the Button list, select the Select Grid Tab option.
4. In the Literal Value list, enter the number of the grid tab that you want to select.  
From left to right, the first grid tab is 1, the second grid tab is 2, and so on.
5. Click the Insert button.

## Clicking the Grid Scroll Button

When you are working in a form with a populated detail area, you can scroll through the detail area from top to bottom or from page to page.

To script the Grid Scroll Button option:

1. On the Command menu, select the Select Grid Row option.
2. In the command pane, under Operation Type, select the option to click by row number.
3. In the Action on Grid Row list, select single-click.
4. Select a source of row number and enter a literal value or select a variable or valid values list.
5. Click the Insert button.
6. On the Command menu, select the Press Toolbar Button option.
7. Select the Grid Scroll Button option.
8. Select one of the following scrolling options in the tree:
  - Page Up

- Page Down
9. Click the Insert button.

---

## Using the Press Push Button Command

You can use the Press Toolbar Button command to script clicking standard toolbar buttons, performing custom functions, selecting grid tabs, and selecting grid scroll buttons. However, some applications contain special buttons that do not appear on the toolbar or menu bar.

You use the Press Push Button command to script clicking these oversized buttons and clickable bitmaps. PeopleSoft AutoPilot displays uses the command pane to display the button options and clickable bitmaps in the active form.

### Using the Push Button Options

Some applications, such as System Setup (P0000), contain forms that use oversized push buttons. Pushing these buttons enables you to select forms on which you can set up, for example, constants for general accounting, accounts payable, accounts receivable, and so on.

---

**Note.** You cannot click these buttons using a Press Toolbar Button command because they do not appear on the toolbar. Therefore, if you use the Press Toolbar Button option in the PeopleSoft AutoPilot Command menu, none of these push button options appears in the command pane. Alternatively, if you select the Press Push Button option, the command pane uses the Select Button to Press list to display the button options that appear on the form.

---

When you write a command to click a button, you often access a new form. You can write a Form command for the new form so that the PeopleSoft AutoPilot script matches the actions that you take in the software. Sometimes, when you click a button, you access a new application. In that case, you must write an Application Interconnect command in PeopleSoft AutoPilot. You can verify whether you have accessed a new application by clicking Help and About PeopleSoft in the toolbar of the active form.

### Using the Clickable Bitmap Options

Some applications, such as Cross Reference (P980011), use clickable bitmaps that enable you to access a new form. Because you cannot use these options by clicking a button on the toolbar, you cannot use the Press Toolbar Button command.

Instead, to click a bitmap option, you use the PeopleSoft AutoPilot Press Push Button command. If a form that contains clickable bitmaps is active, the PeopleSoft AutoPilot command pane displays the available bitmaps.

---

**Note.** For each clickable bitmap, PeopleSoft AutoPilot displays the system-assigned name, such as Bitmap 184, rather than the descriptive label that appears next to the bitmap on the form. In addition, the control properties for the bitmaps do not indicate their identities. Therefore, you must identify the particular bitmap that you want PeopleSoft AutoPilot to click. To help you identify the name of the bitmap, a BlueCue appears with the descriptive label on the form when you click the corresponding name in the PeopleSoft AutoPilot command pane.

---

When you script pressing a clickable bitmap, you typically access another form, and you must write a Form command in PeopleSoft AutoPilot to match the actions. If you access a new application when you click a clickable bitmap, you must write an Application Interconnect command in PeopleSoft AutoPilot so that the current application in the script matches the active application.

---

## Scripting the Press Push Button Command

When you write a Press Push Button command, you enable a script to perform actions that you cannot script with the Press Toolbar Button command. You need to script a Press Push Button command when you are working on a script that tests applications and forms that contain buttons and clickable bitmaps that do not appear on the toolbar of the form. Writing a command to click a button or clickable bitmap enables you to access another form in the application. In some cases, this command enables you to exit to another application.

### Clicking a Button

Suppose that you need to test an application, such as System Setup (P0000), that includes forms with oversized buttons that do not appear on the toolbar. In most cases, you click these buttons to access a new form in the same application or to exit to a new application.

Scripting the Press Push Button command in PeopleSoft AutoPilot enables you to click the button in the form. You cannot write a Press Push Button command to perform this action because the Press Push Button command applies only to buttons that appear on the toolbar.

To click a button in a form:

1. On the Command menu, select the Press Push Button option.  
PeopleSoft AutoPilot populates the command pane with button options only when the active form contains buttons.
2. In the PeopleSoft AutoPilot command pane, select a button from the Select Button to Press list.
3. Click the Insert button.
4. If clicking the button accesses another form in the same application, select Form from the command menu.
5. In the Form list of the command pane, select the name of the active form.
6. Click the Insert button.
7. If clicking the button accesses a new application, select Application Interconnect from the Command menu.
8. In the command pane, select one of the following options:
  - Application (to select the active application)
  - Form (to select the active form)
9. Click the Insert button.

### Clicking a Bitmap

Some applications contain forms that use clickable bitmaps that do not appear on the toolbar. To test one of these applications, you must script a Press Push Button command to click a bitmap. In the command pane, PeopleSoft AutoPilot displays the system-assigned name for each bitmap, rather than the descriptive text that appears next to each bitmap. If you click the system-assigned name in the command pane, PeopleSoft AutoPilot identifies the corresponding clickable bitmap on the form by enclosing it in a BlueCue.

Scripting a Press Push Button command to click a bitmap enables you to access another form or application. You cannot click the bitmap by writing a Press Push Button command because the bitmaps do not appear on the toolbar.

To click a bitmap in a form:

1. On the Command menu, select the Press Push Button command.
2. From the Select Button to Press list, select the name of a tab in the active form.
3. Click the node next to the tab name.
4. From the drop-down menu, select the system-assigned name of a clickable bitmap.

---

**Note.** When you click the bitmap name, the BlueCue that appears in the active form identifies the corresponding bitmap.

---

5. Click the Insert button.
6. If clicking the button accesses another form in the same application, select Form from the Command menu.
7. In the Form list of the PeopleSoft AutoPilot command pane, select the name of the active form.
8. Click the Insert button.
9. If clicking the button accesses a new application, select the Application Interconnect option in the Command menu.
10. In the command pane, select one of the following:
  - Application (to select the active application)
  - Form (to select the active form)
11. Click the Insert button.

---

## Using the Select ComboBox Item Command

Some applications—such as Object Management Workbench (P98220) and Expense Report Review / Entry (P09E2011)—use combo box controls. These controls can appear on forms as edit text fields, pop-up menus, or scrolling lists. PeopleSoft AutoPilot provides the text from the combo box in the command pane. After you write a Select ComboBox Item command and play back the script, PeopleSoft AutoPilot locates the combo box and sends a message to the form to select the text string specified in the command pane.

When you select the command, PeopleSoft AutoPilot populates the Combo Box list in the command pane with either a list or a tree control. A tree control appears in the command pane only when the combo box is under tab controls in the active form.

In the Combo Box list, when you click the name of a control or an item in a scrolling list or pop-up menu, PeopleSoft AutoPilot populates the Choices list with the text names that appear in the combo box, along with the user-defined system codes, which it retrieves from table F0005.

Forms in some applications include hidden combo box controls that are not used. PeopleSoft AutoPilot displays these controls in the Choices list, just as it displays hidden header controls and grid columns in the command pane. You cannot select a default value, such as None, to enter in the combo box.

---

## Scripting the Select ComboBox Item Command

When you write scripts that use forms that contain combo box lists, you select the Select ComboBox Item command from the Command menu of the PeopleSoft AutoPilot form and select a combo box from the Combo Box list in the command pane.

The Select ComboBox Item option is available in the Command menu of the PeopleSoft AutoPilot form only if you launch an application and form that use combo boxes.

After you select an option from the Combo Box list, PeopleSoft AutoPilot populates the Choices list with the available items in the combo box.

Some applications, such as Object Management Workbench (P98220), use more than one combo box list, and the lists depend on one another to establish, for example, search criteria. In these cases, the selection that you make in the Combo Box list in the PeopleSoft AutoPilot command pane changes the items in the Choices list.

To script the Select ComboBox Item command:

1. From the Command menu, select Application.
2. In the command pane, select an application and Fast Path, and then click the Insert button.
3. From the Command menu, select the Select ComboBox Item option.
4. From the Combo Box list, select a combo box.

The system populates the Choices list with the items in the combo box.

5. From the Choices list, select a combo box item.
6. Click the Insert button.

PeopleSoft AutoPilot enters the item from the Combo Box list to the control in the form.

---

## Using the Build Tree Path Command

Some applications use tree path controls. To write script commands for forms in these applications, you must use the Build Tree Path command to create a unique path that uses the tree path in a form.

You use the Build Tree Path command using any combination of literal text or variables. For example, the first node in a tree might consist of a parent, one child, and one grandchild. To use PeopleSoft AutoPilot to write the Build Tree Path command, you first designate the data type that represents the first node in the tree path. Available data types include:

- Literal values, which are the precise text that designates a node in a tree control.
- Variable values, which you set as the text that designates a node in a tree control.
- Ordinal values, which represent the order in which a node appears in a tree path, such as first, second, third, and so on.

You select *Ordinal* as the data type when you want to build a path to the first node in the tree (the parent), the first child of the parent, and the first grandchild of the parent. Clicking the Add button populates the tree path list with a leaf node, which is a node without children. You can create parent-child relationships by clicking the Add button to add nodes.



During playback, PeopleSoft AutoPilot uses the search string that you create to identify the coordinates of each node in the tree in the active form. You can modify the tree path as necessary using the Add and Remove buttons. Removing the parent node also removes all children from the path. To add a node, you must add it to a leaf node.

If you attempt to add a child node to a parent that already has a child, PeopleSoft AutoPilot displays a dialog box indicating that you cannot add a child to the node. You must click a leaf node to create a new node with a child.

---

## Scripting the Build Tree Path Command

This section provides an overview of scripting the Build Tree Path command and discusses how to:

- Build a tree path using variable values.
- Build a tree path using literal values.
- Add a parent node or child to a tree path.
- Remove a parent node or a child from a tree path.

## Understanding Scripting the Build Tree Path Command

You use the Build Tree Path command to write scripts that test applications that use tree controls. You can use any combination of variables or literal text to build a unique path to nodes that exist in the tree path in an active form. You can also modify the tree path by adding and removing nodes from the path that you build.

## Building a Tree Path Using Variable Values

You can select one of the following methods to use variable values to build a tree path:

- Declare a variable and set its value as the text that represents a node in the tree.

In this case, you select *Variable* as the data type for the node.

- Set a numeric value for the variable.

If you select 3 as the value, PeopleSoft AutoPilot selects the third node in the tree or the third child of a parent. In this case, you select *Ordinal* as the data type for the node.

Whether you select *Variable* or *Ordinal* as the data type for the node, the names of the variables you declared appear in the Select Variable list of the command pane.

---

**Note.** Before building the tree path, declare a variable and set its value.

---

See [Chapter 6, “Scripting Actions,” Using a Variable as a Source of Input, page 63](#).

To build a tree path using variable values:

1. From the Command menu, select Variables.
2. Declare a variable and set its value.

---

**Note.** You can assign any value to the variable. However, remember that if you assign a numeric value, that value represents the position of a parent node or a child in the tree path. For example, if you set the value of the variable to 3, the value represents the third node in the tree control.

---

3. When a form that uses a tree control is active, from the Command menu, select Build Tree Path.  
The PeopleSoft AutoPilot command pane contains a Tree Path list and a Data Type list.
4. Select one of the following options from the Data Type list:
  - Text Variable  
Select Text Variable if the variable that you want to use has a text value.
  - Ordinal Variable  
Select Ordinal Variable if the variable that you want to use has a numeric value.
5. From the Select Variable list, select a variable for which you have set a value.
6. Click Add.  
PeopleSoft AutoPilot inserts the variable as a node in the Tree Path list.

## Building a Tree Path Using Literal Values

You can build a tree path using a literal value as the data type to represent a node. You type the name of the node in the Enter Node list exactly as it appears on a form. To create the tree path, you can use literal values as a data type in combination with variable and ordinal values.

To build a tree path using literal values:

1. On the PeopleSoft AutoPilot form, launch an application and form that uses tree controls.
2. From the Command menu, select the Build Tree Path option.
3. Select one of the following options from the Data Type list:
  - Text Literal  
Select Text Literal to use a text value.
  - Ordinal Literal  
Select Ordinal Literal to use a numeric value.
4. In the Enter Node list, type a literal value.
5. Click Add.  
PeopleSoft AutoPilot inserts the literal value as a node in the Tree Path list.

When you play back the script, PeopleSoft AutoPilot finds the node with the literal value, based on the tree path that you build.

## Adding a Parent Node or Child to a Tree Path

After you select a data type and enter a literal value or select a variable, you add a tree path node by:

- Clicking the Add button with the Tree Path list in the command pane unpopulated
- Selecting a node in the tree path list that does not have a child and clicking the Add button.

Remember, you cannot add to a node that already has a child.

To add a parent node or child to a tree path:

1. In the Tree Path list of the PeopleSoft AutoPilot form command pane, click a node that does not have a child.
2. Select one of the following options from the Data Type list:
  - Text Literal
  - Text Variable
  - Ordinal Literal
  - Ordinal Variable
3. Select a variable from the value selection list or enter a literal value in the Enter Node list.
4. Click Add.

To remove a parent node or a child from a tree path:

1. In the Tree Path list of the PeopleSoft AutoPilot form command pane, click a parent node or a child.
2. To remove a parent node and its child, select the parent node, and then click Remove.
3. To remove the child only, select the child, and then click Remove.

## Removing a Parent Node or a Child from a Tree Path

You can modify a tree path by selecting a node or a child and clicking the Remove button. Remember, removing a node also removes the child of the parent.

---

## Using the Database Validation Command

This section provides an overview of using the Database Validation command and discusses how to:

- Define validation.
- Use the validation declaration.
- Use the validation association.
- Execute validation.
- Use the Expect No Matching Records option.

## Understanding Using the Database Validation Command

Use the data validation action commands to verify the data that you entered in the database using commands in your PeopleSoft AutoPilot script.

The data validation process in PeopleSoft AutoPilot consists of the following steps:

- Declaration, in which you give the validation a name.
- Association, in which you pair the values that you enter in header controls or grid columns with columns in a database table.

- Execution, in which you use record selection criteria to verify whether the records that you entered during scripting actually were entered in the database.

You can also verify that the system deleted records from the database.

## Defining Validation

You script validation commands to compare a data set that you created in PeopleSoft AutoPilot and ran in PeopleSoft EnterpriseOne software with a data set that was written to the database. These commands confirm that the system has entered records in the database as you expected. You also use validation for process testing, to verify that data moved as you intended it to move through a sequence of applications that are included in a transaction cycle. Finally, you use validation to validate values that cannot be accessed through PeopleSoft AutoPilot, such as century information. In most transaction fields, you make no entry for century. Using validation commands, you can verify that century data actually appears in the database.

## Using the Validation Declaration

You script a new database validation command each time that you declare a validation. You can declare a validation at any point in the script.

The declared validation includes:

- A validation name that you select
- A table to be validated, which you select from a list

Declaring the validation is similar to declaring a variable in that it provides only a name for the validation. In essence, it provides the means to store values.

## Using the Validation Association

Associating a validation enables you to store data. You can gather this data from different points in the script. No action is taken when you associate a validation. Rather, you access the storage space that you created when you declared the validation and you store the data of your choosing.

In associating a validation, you define the values to be validated against chosen columns in the database. You select a column, which cannot be associated more than once in a script, such as ABALPH—NameAlpha in the Address Book program (P01012)—and you associate that column with a value, which can be derived from a literal value, variable, header control, or grid column. You then select a database value type that PeopleSoft AutoPilot uses to validate the data that you enter in the database when you run the script.

You must specify the following information to script the validation association:

- Validation name (declared validation).
- Database column identification, which you select from a list.
- Source of expected data, such as a header control.
- Database value type, either key selection or validation value.

Validation association can occur at as many different points in the script as you select, but you must use both a key selection value and a validation value.

## Key Selection Value

The key selection value specifies the database column that contains the specific records that you want. When you mark a validation association as a key selection value, a database record that matches an associated column must exist or the validation fails. When PeopleSoft AutoPilot runs the validation, it uses the key selection value to verify whether the system has updated the correct database column with the values that you stored during association.

When you select a key selection value, you select a value that all records that you are validating have in common. For example, to validate data for the Address Book Revision form, you might select the database column A5AN8 or ABAN8 as a key selection value because all address book records have an address book number.

## Validation Value

When you mark a validation association value, you select a specific record set to be validated. However, while the key selection value indicates the database where the records that are to be validated exist in the database, the validation value specifies the values, such as names, that you expect to find in the database.

## Executing Validation

You run the validation using the record selection criteria that you establish in validation association. PeopleSoft AutoPilot retrieves the specified data from the database through SQL. You can then compare the retrieved data with the data that you expected the system to return. Running the validation indicates whether the data that you entered and stored during association actually updated the database in the condition that you specified.

The following actions occur when you script running the declared and associated validation:

- An SQL statement is generated
- The system queries the database for the specified data
- The system compares the returned data to the expected data

The generated SQL statement contains the table that you select when you declare the variable, the validation value columns that you select during association, and the key selection that you make during association.

An SQL statement that you generate when you run the validation might contain the identity of the table that contains the expected data (for example, F0101), the columns that contain the expected data (for example, ABALPH, ABAT1 and ABAN8), and the key column that joins all of the columns (such as ABAN8).

The SQL statement queries the database and retrieves data that conforms to the statement elements. You compare the results of that query against the results that you expected, based on the data that you stored during validation association.

If an error exists in associating the data to be validated, PeopleSoft AutoPilot displays an error message. For example, if you do not select a key column during association, PeopleSoft AutoPilot notifies you that no record selection criteria have been chosen. If you associate the key column with an incorrect header control, grid column, or variable, PeopleSoft AutoPilot indicates that the SQL statement contains an error.

In either of these cases, you can make any necessary corrections to the script. In addition, running a validation that fails does not stop the script from playing through to completion. The test results compare the data that you expected to return with the data that is actually returned.

## Using the Expect No Matching Records Option

If you enter a record, successfully validate it, and then write a command to delete the record, you can validate that the system successfully deleted the record from the database. To do so, you run the validation again and select the Expect No Matching Records option. When you run the script, PeopleSoft AutoPilot again checks the database. The validation runs successfully if PeopleSoft AutoPilot indicates that the record that you deleted from the database no longer exists. When you select this option, PeopleSoft AutoPilot verifies that no records match the criteria in the SQL statement.

---

## Scripting the Database Validation Command

You enter database validation action commands as you write a script. No formula exists to determine where the commands must occur. However, each of the three phases—declaration, assignment, and execution—must occur for the validation to take place.

### Declaring a Validation

You can declare one or more validations as soon as you begin a script. You do not have to place the validation declaration command line at the top of the script if you want the validation to be effective within any node in the script, as you do when you declare a variable. However, declaring the validation early enables you to easily store data through association as you write the script.

To declare a validation:

1. From the Command menu, select the Declare New Validation option.
2. In the command pane, enter a name in the Validation Name list.
3. Select an option in the Database Table list.

This selection identifies the database table against which you validate data from PeopleSoft AutoPilot.

4. Click the Insert button.

---

## Associating a Validation

After you declare the validation, you have a place in which you can store values. During validation association, you select values that you want to validate and you pair (or associate) those values with values in the database.

You can write scripts that test scenarios that involve multi-currency with both accounting methods Y and Z. For example, you can use divisors instead of multipliers for exchange rate calculations.

There are two types of validation associations:

Association Type	Description
Key selection value	Associations that determine which database record (row) you verify.
Validation value	Associations that specify columns within that row whose values you want to verify.

Each validation must include at least one key selection value association and at least one validation value association.

To associate a validation:

1. From the Command menu, select the Associate a Validation Column option.
2. Select a validation name that you created using the Declare New Validation option by scrolling through the list and clicking a name.
3. Select a column name from the Database Column list.
4. In the Value Type group box, select Validation Value.

This option associates the selected column with a particular value.

5. (Optional) In the Currency Type group box, select one of the following currency options:
  - Domestic
  - Foreign

---

**Note.** If you have not declared a currency validation, the system hides and disables these options. You must declare a currency validation to use these options.

---

6. Click a selection from the Source of Expected Data list.  
The source can be a literal value, variable, header control, or grid column.
7. In the value selection list, enter a literal value or select the name of a variable, header control or grid column.  
This step specifies the value that you associate with the database column.

---

**Note.** With playback activated, if you select a header control or grid column, PeopleSoft AutoPilot uses a BlueCue to highlight the designated control or column in the form. Be sure to select a header control or grid column to which you have previously entered a value. Likewise, if you have selected the name of a variable, be sure that you have followed the steps outlined previously for setting the variable's value.

---

8. Click the Insert button.
9. Before you finish the association, ensure that you write a command that follows steps 1-7, but select a key selection value from the Value Type options.

Data from all the individual columns from which you expect to have data returned is now associated with a single key column.

---

## Executing a Validation

After you have declared the validation and associated the data with database tables and columns, you can execute the validation.

To execute a validation:

1. From the Command menu, select the Execute Validation option.
2. From the drop-down menu in the Validation Name list, select a validation.

---

**Note.** PeopleSoft AutoPilot populates the SQL Statement list. The statement contains the data dictionary aliases of the tables and columns that you associated with the data that you entered, the name of the validation, and the key selection value.

---

3. Click the Insert button.

Later in the script, you might delete the records. You can validate that these deleted records are no longer in the database.

4. Repeat steps 1-3.
5. Select the Expect No Match Records check box.
6. Click the Insert button.

---

**Note.** You can declare and set the value of a variable to test validation success.

---

---

## Using the Command Line

Sometimes, you may need to run other programs within a PeopleSoft AutoPilot script. For example, you might prepare a PowerPoint or Excel presentation that you want to include within the script. After you run that presentation, you might decide to close the program and then return to PeopleSoft AutoPilot to continue scripting inputs.

You can complete these tasks by selecting the Command Line option from the Command menu. Type the path to the program that you want to run. (This is similar to using the Run function in Microsoft Windows.) PeopleSoft AutoPilot opens the program and the document or presentation that you created.

You can also send a command line message to create screen shots of PeopleSoft EnterpriseOne software forms at designated points in the script playback process. You can create the screen shots in a particular language version, and you can store them in a directory and file that you create.

---

## Scripting a Command Line Command

Scripting the Command Line command requires that you know the path to the program that you want to run and, for example, a document in the program that you want to open. You can also create a Command Line command that creates a screen shot of the current form. You specify a folder where PeopleSoft AutoPilot stores the screen shots.



## Using the Command Line Command

To open a program, type the path for the program in the Command Line field. On playback, PeopleSoft AutoPilot reads the path and opens the program.

To use the Command Line command:

1. From the Command menu, select the Command Line option.

The command pane displays the unpopulated Command Line list and options.

2. Select the Command Line option.
3. Complete the Command Line field.

Type the path to the program that you want to open.

---

**Note.** Deactivate playback if you do not want the program to open while you are writing the script.

---

4. Click Insert.

## Creating a Screen Shot of the Current Form

You can set up the path and file extension for any screen shots that you create using the Capture Current PeopleSoft Window command. You do so by clicking Tools, selecting Options from the drop-down menu, and then selecting the Directories tab on the Options form.

In the Screen Capture field on the Directories tab, you type the path where PeopleSoft AutoPilot stores the screen shots of forms. You then select the format in which you want to save the images.

To use the Command Line command to create a screen shot of a form, type a name for the image in the File Name field. PeopleSoft AutoPilot creates a screen shot of the current form and saves it in the specified format in the specified directory.

To create a screen shot of the current form:

1. From the Tools menu, select Options.
2. On the Options form, select the Directories tab.
3. Complete the following field:

- Screen Capture

Type the path where you want PeopleSoft AutoPilot to store the screen shots.

- Format

Using the drop-down menu, select the file extension, such as .tif, that you want to use for the screen shots.

4. Click OK.
5. With a script open, select Command Line from the Command menu.
6. Click the Capture Current PeopleSoft Window option.
7. With a PeopleSoft EnterpriseOne form active, complete the File Name field.
8. Click Insert.

PeopleSoft AutoPilot stores the screen shot in the specified location.



## CHAPTER 7

# Working with the Script pane

This chapter provides an overview of the Script pane and discusses how to:

- Modify scripts.
- Use script retention.
- Reuse Scripts.

---

## Understanding the Script pane

You can work in the Script pane to delete, modify, and move the commands that you have created. Working with the Script pane requires that you understand the structure of the script tree that you build as you insert commands. You must also learn how to work with the tree to change its structure.

Parent-child relationships make up the script tree. Every script begins with the Begin Script command, from which any number of commands descend. Subsequent context commands are the parents of action commands and sometimes of other context commands. These parent context commands and their children make up nodes in the Script pane. PeopleSoft AutoPilot indents any command that is the child of another command. You can change the sequence of commands and the relationship between commands by dragging and dropping. For example, you can make one context command the child of another. Any changes that you make to the parent command affect the child command.

All modifications change the way that the script is run. The changes that you make should be based on what you want to accomplish by running the script. For example, you might drag a declared variable command line to the top of the script to make it global because you need all the commands in the script to have access to the value that you set for the variable.

## Working with the Script pane Structure

As you write PeopleSoft AutoPilot scripts, you build a tree structure in the Script pane that is based on parent-child relationships. If you understand the structure of scripts, you can more easily modify scripts and adapt them to your specifications.

Because scripting requires a context, context commands are the basis of each script that you write. Each context command that you write and insert in the script creates a node, which appears in the Script pane with a plus or minus sign that you can use to expand or collapse the node.

Each command that you write, whether it is a context command or an action command, becomes a command line in the script. Action command lines are attached to context command nodes, are indented beneath the node, and are affected by any changes that you make to the node, such as changing the repeat count.

In some cases, a context command line that forms a node is also a child of another context command line that forms a node. In these cases, PeopleSoft AutoPilot indents the child node, which is affected by any changes that you make to the parent command.

The Script pane contains an insertion cursor, which indicates the position of the next command that you write. You can change the position of the insertion cursor either by clicking a command line or by dragging the insertion cursor. You can drop the insertion cursor into the script tree as a child of a node. In that case, the next command you write is a child. Alternatively, you can make the next command a parent that is independent of changes to other nodes in the script.

## Using Command Lines

Command lines illustrate the selections that you make in the command pane to create context or action commands. The command lines in the Script pane express either the context in which you create the script or the actions that you take within the context. For example, a header in a form is a context; the action that you take within the context might be typing data in a specified field.

Context commands direct PeopleSoft AutoPilot to applications, universal batch engines (UBEs), processing options, interconnected applications, forms, header controls, grid columns, and Query By Example (QBE) lines. They therefore express the environment in which actions—such as typing data and clicking buttons—are carried out. Context commands form the trunk of the script tree.

The context command that initializes a series of action commands and other context commands forms a node. The expand/collapse button in the Script pane identifies the node. The expand/collapse button that represents the node appears in the Script pane as a plus or minus sign next to the context command line.

Context commands can perform the following functions:

- Form nodes that can be identified in the script with the node symbol or button, which appears as a plus or minus sign.

Nodes can be expanded or collapsed by clicking the expand/collapse button. Nodes can function as the parents of action commands and, sometimes, other context commands. The child commands are indented beneath the parent context commands in the Script pane.

- Form discrete command line units.

If two nodes are parallel, commands that you add to one node do not affect the node that is parallel to it.

- Initiate a sequence of other commands.

The sequence can consist of the action commands and other context commands that are children of the parent command.

- Run multiple times if you change the repeat count of a node.

Any commands that are attached to a context command in the Script pane are played back as many times as you specify in the repeat count.

After you script context commands, you script action commands to specify actions. In contrast to context commands, action commands have the following characteristics:

- They must be attached to (or be the child of) a context command.
- They cannot have child commands attached to them.
- They cannot have repeat counts
- They are always indented beneath context commands in the Script pane.

This indicates that they are subordinate to context commands in the command hierarchy.

## Using the Insertion Cursor

The insertion cursor, which appears in the Script pane as a red arrow, indicates the position in the script where you can insert a new command. If you insert commands sequentially without adjusting the script, the insertion cursor appears at the end of the script each time that you insert a command.

You can move the insertion cursor from one point in the script to another by clicking a command line. This moves the insertion cursor to the position directly below an action command line. If you then create and insert a new command, it appears at the point of the insertion cursor.

If you click a context command line, the insertion cursor appears at the end of the selected branch. If you leave the insertion cursor in this position, a new context command that you write creates a node, indicated by a minus or plus sign, that is parallel to the node on which you clicked. New commands that you write are attached to this node.

### Expand/Collapse Button

The button that identifies the node also enables you to expand or collapse it. The expanded node reveals all command lines that are attached to the node. The collapsed node reveals only the context command that you scripted to initiate the node. When you expand a node, the button displays a minus sign. When you collapse a node, the button displays a plus sign.

### Parallel Nodes

When you change a node that is parallel to another node, the parallel node is unaffected. For example, if you write an Application command and write two Form commands in the same application, the Form commands are represented by nodes that are parallel to one another in the Script pane. Any change that you make to one does not affect the other.

### Indented Nodes

A node that is indented beneath another node in the Script pane is affected by any change that you make to the parent. The indentation of nodes reveals the hierarchy of context commands. For example, to work with a particular form, you must first select an application; therefore, PeopleSoft AutoPilot inserts and indents the Form command line below the Application command line. Similarly, to add a form by clicking a button, PeopleSoft AutoPilot inserts and indents the Press Toolbar Button command below the Form command line.

The hierarchy of nodes expresses the logic that you follow to build PeopleSoft AutoPilot scripts. For example, you might write the following sequence of commands to enter data to the header of a form and update the database by clicking OK:

1. Application
2. Form
3. Header
4. Type To
5. Press Toolbar Button {OK}

Because you must launch an application before you can write any of the subsequent commands, the Application command is a parent node in the Script pane. Likewise, you must launch a form before you can enter data in a header control, so the Form command is a parent of the Header command, which is indented. Finally, the header is the context for performing the action of entering data to a control, so the Type To command is indented beneath the Header command line.

Any context or action command that you insert in the script as a child of another command is affected by changes that you make to the parent. For example, if you change the repeat count in an Application command line to 3, during playback the application launches three times, and any action commands that you write—such as clicking a toolbar button in the form—are performed three times.

## Drag and Drop

You change the sequence of commands and the structure of the script by using the mouse to drag and drop commands. Note these points about the drag-and-drop capability in the Script pane:

- An action command within one context command node cannot be dragged into another context command node.  
For example, when a Type To command is attached to a Form command node, you cannot drag it to another Form command node.
- A context command node that is attached to one Application node cannot be dragged into another Application node.
- A context command node that you drag onto another context command node and insert as a child includes all commands that are attached to it.
- A context command node that you insert as a child is included in the playback of the parent context command node.
- The repeat count of the parent context command node applies to that node and to any other nodes that are attached to it as children.

If two nodes are parallel, you can make one node a child of the other by dragging and dropping the node. PeopleSoft AutoPilot indicates the parent-child relationship by indenting one node beneath the other.

Before PeopleSoft AutoPilot creates the parent-child relationship, it displays the dialog box that prompts you to confirm the action.

## Repeat Count

Every context command that creates a node in the Script pane contains a repeat count. The repeat count specifies the number of times that PeopleSoft AutoPilot plays the node and all of the commands attached to it. You can change the repeat count by selecting the node, entering a new repeat count in the command pane, and clicking the Update button.

## See Also

Chapter 6, “Scripting Actions,” Updating the Repeat Count in a Node, page 63

---

# Modifying Scripts

This section provides an overview of changes to scripts and discusses how to:

- Collapse the script tree.
- Expand the script tree.
- Use nodes.
- Delete command lines.
- Change the sequence of action commands.

- Change the sequence of context commands.
- Edit an action command line.
- Edit a context command line.

## Understanding Changes to Scripts

You can configure scripts to your precise specifications to test applications. You can also alter the structure of the script tree by expanding or collapsing nodes, or you can make substantive changes by adding, deleting, editing, or dragging commands.

You can change the order of the commands and, therefore, the structure of the scripts that you create, either as you are scripting or after you have completed scripting a series of commands. You can use the PeopleSoft AutoPilot command pane to add, delete, or edit commands. You can also modify the structure of the script tree by moving the insertion cursor and command lines.

### Expanding and Collapsing a Node

When you expand all the nodes in a script, you can view the script in its entirety, with all command lines exposed. However, with long scripts, it is useful to view only a portion of the scripted commands. In that case, you can collapse nodes so that only the context commands that originated them are visible in the Script pane. You can collapse or expand the entire script or certain portions of the script by clicking the node buttons.

You can select any point at which to collapse the tree. For example, when you click the expand/collapse button on a parent node, any nodes that you have inserted in it also collapse. When you click a child node, only that node collapses.

### Adding Command Lines

You can insert a command in a script after you have passed the point where you want to insert it. For example, suppose that you want to script an input to a header control after you have scripted the move to another form. You can accomplish this by placing the insertion cursor at the point in the script where you want to insert a new command.

You can use the insertion cursor to insert a new command in the script only if the buttons in the toolbar are active. If they are not active, press and hold the mouse button, drag the insertion cursor on top of a command line, and release the mouse button. If the system prompts you to insert the cursor as a child, click Yes.

### Moving Command Lines

You can also modify an existing script by moving command lines. You can move action commands that are attached to a context command to change, for example, the order in which the action commands play back.

You can change the order of action commands within a context command node by using the drag-and-drop capability in PeopleSoft AutoPilot. During playback, PeopleSoft AutoPilot replays these commands within the node in the new order. However, you can also change the structure of a script by moving a context command and inserting it as a child of another context command.

### Editing Command Lines

You can also make substantive changes to the content of the script by editing the command lines in the Script pane. You make these changes by selecting a command line in the Script pane and then using the command pane to make new selections from lists to update the content of the command.

---

**Note.** The command lines for the Press Toolbar Button command cannot be edited. These must be deleted or added to the script as necessary.

---

## Collapsing the Script Tree

Access the PeopleSoft AutoPilot form.

To collapse the script tree:

1. In the Script pane of the PeopleSoft AutoPilot form, expand nodes by clicking node buttons that displays a plus sign.
2. Click a node that shows a minus sign.

The node collapses and displays only the context command line.

---

**Note.** You can also collapse all nodes in a branch by right-clicking on the first node of the branch and choosing Collapse All. You can collapse all nodes in the script by right-clicking on the Begin Script line and choosing Collapse All.

---

## Expanding the Script Tree

Access the PeopleSoft AutoPilot form.

To expand the script tree:

1. In the Script pane of the PeopleSoft AutoPilot form, collapse nodes by clicking any node button that displays a minus sign.
2. Click a node that displays a plus sign.

The node expands and shows the context command line and any commands attached to it.

---

**Note.** You can also expand all nodes in a branch by right-clicking on the first node of the branch and selecting Expand All. You can expand all nodes in the script by right-clicking on the Begin Script line and selecting Expand All.

---

## Using Nodes

A node consists of a parent context command and any related context commands and action commands that you attach to it. After you create a node by scripting a context command, you can write action commands and sometimes other context commands to develop the node and the script.

To add a command line to an existing script:

1. In the Script pane of the PeopleSoft AutoPilot form, click the insertion cursor.
2. Press and hold the mouse button.
3. Drag the insertion cursor to the point in the script where you want to insert a new command.

As you drag the cursor, an indicator arrow appears. The arrow, which points up or down, indicates the placement of the cursor above or below a selected command line.

4. Release the mouse button at the appropriate point.
5. Follow the steps required to insert the command in the script.

---

**Note.** If you cannot insert a command at a particular point in the script, PeopleSoft AutoPilot displays a disallow symbol as you drag the insertion cursor.

---



## Deleting Command Lines

You can delete lines from a script. If you delete a parent command line, you also delete all of its children.

To delete a command line:

1. In the Script pane of the PeopleSoft AutoPilot form, select a command by clicking it in the script.
2. Right-click.
3. Select Delete.

## Changing the Sequence of Action Commands

Access the PeopleSoft AutoPilot form.

To change the sequence of action commands:

1. In the Script pane in the PeopleSoft AutoPilot form, select an action command line by clicking it.
2. Press and hold the mouse button, and drag the selected command line until it is on top of another command.  
The mouse pointer indicates whether the command appears above or below the targeted command line in the Script pane.
3. Release the mouse button.

## Changing the Sequence of Context Commands

Access the PeopleSoft AutoPilot form.

To change the sequence of context commands:

1. In the Script pane of the PeopleSoft AutoPilot form, select a node by clicking it.
2. Press and hold the mouse button, and drag the node.  
A disallow symbol indicates that you cannot drop the node in a particular spot in the script.
3. When the target node is highlighted and the indicator arrow is pointing downward, release the mouse.
4. In the dialog box, click Yes or No when the system prompts you to insert as a child.  
PeopleSoft AutoPilot inserts the dragged node as a child to the target node if you click Yes. If you click No, the nodes are parallel.

## Editing an Action Command Line

Although you can change only the repeat counts of context command lines, you can change the content of many action commands by clicking the command line and then selecting options in the command pane.

To edit an action command line:

1. In the Script pane of the PeopleSoft AutoPilot form, click an action command line.  
The command pane displays lists from which you can make choices to update the content of the command line.  

---

**Note.** PeopleSoft AutoPilot highlights the original options that you made in the command pane.

---
2. In the command pane lists, select any new options.

3. Click Update.

PeopleSoft AutoPilot updates the command line in the Script pane to reflect the changes.

## Editing a Context Command Line

The substance of the context command itself cannot be changed unless you delete it and insert a new one. However, you can change the number of times that PeopleSoft AutoPilot loops through the node during script playback.

To edit a context command line:

1. In the Script pane of the PeopleSoft AutoPilot form, click the context command line.
2. In the command pane, select a value from the Define Repeat Count list, such as literal or variable.
3. In the Repeat Count list, type the number of times that you want PeopleSoft AutoPilot to loop through the node during playback.
4. Click Update.

---

## Using Script Retention

This section provides an overview of script retention and discusses how to:

- Save scripts.
- Use the Include command.
- Link variables between scripts.
- Share scripts.

## Understanding Script Retention

PeopleSoft AutoPilot enables you to save, modify, reuse, combine, and send scripts. These capabilities broaden the scope of and audience for tests. With PeopleSoft AutoPilot, you can perform the following tasks, which are integral to building a system of scripts:

- Save scripts, which you can reuse or modify.
- Include scripts with other scripts to broaden the scope of testing.
- Pass variables between scripts in a master script that consists of a parent script and one or more children.
- Share scripts.

You can save scripts either on your local drive or in the PeopleSoft AutoPilot script repository. When you build scripts by including one or more scripts with another, you can retrieve the scripts either from the local drive or from the repository. Scripts that you create by including other scripts can pass variable values; you accomplish this by declaring variables as external and creating links between variables in separate scripts. In addition, you can send an email message to colleagues with any script that you create.

## Saving Scripts

You can save scripts as you work by using the File menu. You assign a name to the script, which is saved in the directory you specify on the Directories tab of the Options form. Give the file a name that relates to the application that you are testing. If you continue to work on the script, you can save it as you work.

If PeopleSoft EnterpriseOne or PeopleSoft AutoPilot fails during a PeopleSoft AutoPilot session, PeopleSoft AutoPilot saves any scripts that are open. You can use the Configure tab on the Options form to set the conditions under which PeopleSoft AutoPilot auto-saves scripts as you work.

## Using the Include Command

You can expand your testing scope by including one or more scripts on your local drive or in the script repository with a parent script. Do so by selecting the Include Local Script option or the Include Reposited Script option from the Tools menu. PeopleSoft AutoPilot creates a copy of the script that you want to include and inserts it at the point in the open script that you have placed the insertion cursor. The included script becomes a child in a master script.

Whether you include a script from your local drive or a script from the repository, PeopleSoft AutoPilot displays a form that includes all of the scripts that you have stored locally or a form that you can use to select from the repository. You select the scripts to include, and PeopleSoft AutoPilot inserts them as children of the master script. An Include command line contains the path to the included script—for example: `[C:\atg\ats\UBE blind app.ats]`.

You sometimes must edit a script before you include it with another. For example, if you script data input in one script, and that data is also included in another script, you must delete the data from the included script before you write an Include command. If you do not, when PeopleSoft AutoPilot plays back the included script, it loads data into a form twice, which results in an error. You can open a script that has scripts included in it and edit any of the included scripts. PeopleSoft AutoPilot reloads the original script with the changes that you make to the included script.

Each time that you write a command to include a script, PeopleSoft AutoPilot displays a dialog box that prompts you to specify whether to continue script playback on an Include branch error. If you click Yes, when an error occurs during the playback of the included script, PeopleSoft AutoPilot reports the error but continues with playback of any other included scripts that exist. This feature is particularly useful if you are running long scripts or batches of scripts.

### See Also

[Chapter 6, “Scripting Actions,” Using a Variable as a Source of Input, page 63](#)

## Linking Variables Between Scripts

When you include scripts with a parent script, you can also share variable values between the scripts. You use *variable linking* to do this. When linking variables, you declare a variable in a parent script. When you write a script that you include in the parent script, you also declare a variable; however, to link the variable, you declare it as external. PeopleSoft AutoPilot enables you to link the externally declared variable to any variable that you declare in the parent script. With the link, the value that you set for the variable can be passed between scripts.

To increase the versatility of scripting, PeopleSoft AutoPilot enables you to designate a default value for any variable that you designate as external. Doing so enables you to run an included script in standalone mode. PeopleSoft AutoPilot uses the default value wherever the value is needed in the script.

You can link variables between locally generated scripts and reposit scripts, or you can link variables between a local script and a reposit script. If the included script contains a variable that you have declared as external, PeopleSoft AutoPilot prompts you to identify the variable to which you want to establish a link in the parent script.

### Default Values for External Variables

You can declare a variable as external and assign a default value to it. You give a variable a default value so that you can run a script in standalone mode. For example, you might write a script that tests one set of functions by itself. You might then include this script with one or more others. You can pass values between variables in the scripts by declaring a variable as external and linking it to a declared variable in the parent script. However, you might also need to play back the original script in standalone mode. If you assign a default value to the variable and run the script in standalone mode, PeopleSoft AutoPilot uses the default value to run the script. If you leave the default value of the external variable blank, PeopleSoft AutoPilot reads the value as a null string.

When you create a script with an included script and linked variables, PeopleSoft AutoPilot can pass a default value in the parent script to linked variables in the child script. However, if you assign a default value to an external variable in the child script, PeopleSoft AutoPilot does not pass this value to the linked variable in the parent script. In this case, PeopleSoft AutoPilot either overrides the default value with a value that you set for the variable in the parent script, or it passes the value as a null string if you do not set a variable value in the parent script.

The following table summarizes three scenarios and the results that occur when you write scripts with default variable values:

Location Where Default Value Is Set for External Variable	Variable to which Link Is Created	Result
Parent script	External variable in child (included) script	PeopleSoft AutoPilot passes the default value to linked variables in any child (included) scripts.
Child script	Variable in parent script	PeopleSoft AutoPilot overrides the default value during playback, either with the value that is set for the variable in the parent script or with a null string if no value is set for the variable in the parent script.
Standalone script	Not applicable	The variable with the default value behaves as a local variable. PeopleSoft AutoPilot uses the default value wherever the script indicates.

### External Variables for Script Linking

You use external variables to pass values between scripts. An external variable can receive a value from a variable in another script to which it is linked, or it can pass a value to a variable in another script. An option in the command pane enables you to designate a variable as external.

For example, suppose that you create script A and declare a variable X. You then create script B, declare a variable X, and designate it as external. Next, you include B with A. Script A is the parent script, and script B is the child. PeopleSoft AutoPilot prompts you to link the externally declared variable to a variable in script A. You link variable X to variable X. In this case, you have provided the mechanism for passing a variable value from one script to another. However, suppose that, in script A, you set the value of variable X to an address number, such as 4245. With the link established, PeopleSoft AutoPilot can now pass this value to script B when you run the two scripts together.

## Variable Links

When other scripts that contain external variables are included with a parent script, you must define the links to each external variable so that PeopleSoft AutoPilot can pass values between scripts. Defining the links enables PeopleSoft AutoPilot to store data in a declared variable in the parent script or retrieve data from the parent script and reuse it in included scripts that contain external variables.

## Continue Script Playback on Include Branch Error

Before you run an Include command, PeopleSoft AutoPilot displays a message box that prompts you to continue the script playback on an Include branch error.

Continuing the script on include branch error means that, if PeopleSoft AutoPilot encounters an error in an included script during playback, playback moves on to the next included script rather than failing the entire playback session. Clicking Yes is recommended if you intend to test a long series of included scripts.

After you respond to the prompt, PeopleSoft AutoPilot displays another message that prompts you to link any unlinked external variables.

## Link Variable Form

You use the Link Variable form to establish the link between the variable that you declared as external in your included script and a variable that you declared in the parent script.

The Link Variable form contains a link path that identifies the name of the included script and its externally declared variable, along with the name of the variable in the parent script. The following path indicates that the included script is child.ats and that its externally declared variable <x> is linked to the variable <x> in the parent script.

```
Link C:\PeopleSoft AutoPilot\ats\child.ats <x> to <x>
```

To change the link path, you click the name of a different variable from the parent script.

When you select a variable, PeopleSoft AutoPilot establishes the link and inserts a Link script object and command line in the Script pane. PeopleSoft AutoPilot inserts and maintains the link relationship in the parent script.

If you click Cancel on the Link Variable form and then attempt to save the script, PeopleSoft AutoPilot continues to prompt you to supply the variable link. You can save the script without establishing the link; however, when you open it, the system displays the Variable Link form again.

## Recursive Value Searching

PeopleSoft AutoPilot searches recursively for links. This means that the search continues until it meets one of the following conditions:

- The linked variable in the parent script is not external.
- The linked variable is in the master parent script; PeopleSoft AutoPilot has searched until it reached the top of the script tree.

PeopleSoft AutoPilot can repeatedly search for a value; therefore, you can establish links between variables in multiple script parent-child relationships. When you play back a script that has external variables, PeopleSoft AutoPilot searches the parent script for the link to the external variable. When PeopleSoft AutoPilot finds the link, the Link script object determines whether the Declare command for the variable is external. If the declared variable is not external, PeopleSoft AutoPilot stops its search. If the declared variable is external, PeopleSoft AutoPilot continues to search for links.

For example, suppose that you create four separate scripts: A, B, C, and D. Each script contains variables that you declare as external. You set the value of the variable in script D, and then include scripts C and D with B. When you insert the Include command, PeopleSoft AutoPilot prompts you to create the links between the parent script (B) and the included scripts (C and D). When you run the script, PeopleSoft AutoPilot searches for the link in the parent script. If you did not declare the linked variable in the parent script as external, the recursive process ceases.

Suppose, however, that you decide to include script B with script A. Script A now becomes the master parent script. You now declare the variable in script B as external and include it with script A. When you run the script, PeopleSoft AutoPilot searches for the link in the parent script. When PeopleSoft AutoPilot reaches script B, it reads the declared variable. Because that variable is external, it continues to search the tree until it reaches the master parent script, script A.

You can create as many parent-child relationships between scripts as you need. Because it continues to search for values until it finds none, PeopleSoft AutoPilot can maintain as many links as you need.

The master parent script—that is, the script at the top of the tree—should not contain variables that you declare as external. These are unresolved variables. They have been declared as external, but they have not been linked. Declaring these variables as external causes the script playback to fail. In a well-written script, all external variables link to other variables in a parent script.

## Sharing Scripts

After you create a script and modify it as necessary, you can email the script to other people. However, only people who have PeopleSoft AutoPilot installed can run the script.

### Broken Links

If you remove an external attribute from a variable, you break its link to the link object, which can no longer find its reference. If you find a link object in a parent script, you should verify that a variable in the child script is designated as external in the Script pane. If that designation does not exist, the link is broken.

PeopleSoft AutoPilot notifies you that broken links exist when you attempt to load the script containing the parent and the included scripts. The Research Broken Links form displays the broken links.

---

**Note.** The Research Broken Links form contains the name of the broken link path and a description of the link error. If you click the broken link, PeopleSoft AutoPilot highlights the corresponding Link command line in the Script pane.

---

You might break the links by inadvertently deleting the linked variable in the parent script or by deleting one of the included scripts before you remove the links in the parent script. In either case, you must repair the broken links.

---

## Reusing Scripts

This section provides an overview of script reuse and discusses how to:

- Include scripts.
- Include one local script with another
- Include a repositied script with another script
- Edit an included script

- Create variable links.
- Declare a variable as external.
- Assign a default value to an external variable.

## Understanding Script Reuse

After you create and modify scripts in PeopleSoft AutoPilot, you can save them and then reuse them, as necessary. You can include scripts that you have created in other scripts that you are working on, and you can email saved scripts to other script writers and testers, as long as the other people have PeopleSoft AutoPilot installed.

Saving and reusing scripts enables you to reduce the workload of other people who test the same applications that you are testing.

PeopleSoft AutoPilot also enables you to declare variables as external. You can link the variable that you declare as external to a variable in another script and pass the value of the variable between scripts. Moreover, you can change the value of the variable to which you link the external variable. Because you can pass values between scripts and change those values, scripts are versatile, reusable, and dynamic.

## Including Scripts

Including scripts enables you to broaden your testing scope. You can include one or more scripts with another script, either from your local drive or from the script repository. Each time that you include scripts with another script, you create a master script. The script that contains the included scripts is the parent, and the included scripts are its children. You can edit included scripts within the master; however, any changes that you make to an included script affect the master.

## Including One Local Script with Another

Access the PeopleSoft AutoPilot form.

To include one local script with another:

1. In the Script pane, place the insertion cursor at the point where you want to insert the included script.
2. From the Tools menu, select the Include Local Script option.
3. On the Select Files to Include form, click the name of the script that you want to include.

To include more than one script, press the CTRL key or the SHIFT key and click another script. When you click the name of a script, its name appears in the File Name list.

4. Click Open.
5. In the Continue Script Playback on Include Branch Error form, click Yes or No.

PeopleSoft AutoPilot inserts the script in the Script pane of the open script at the point of the insertion cursor. PeopleSoft AutoPilot inserts a Continue Playback or Fail Playback message on the Include command line.

You can change the message from Continue Playback to Fail Playback by clicking the command line and clearing the Continue Playback on Error check box in the command pane.

## Including a Reposited Script with Another Script

Access the PeopleSoft AutoPilot form.

To include a repositited script with another script:

1. In the Script pane, position the insertion cursor at the point where you want to include a script.
2. From the Tools menu, select the Include Reposited Script option.
3. On the Select Script form, complete any of the fields on any of the tabs to narrow the search, and then click OK.
4. On the Include Repository Script form, click the name of one or more script titles, and then click Include.
5. On the Continue Script Playback on Include Branch Error form, click Yes or No.

PeopleSoft AutoPilot places the included script in the Script pane of the open script at the point of the insertion cursor.

## Editing an Included Script

Access the PeopleSoft AutoPilot form.

To edit an included script:

1. In the Script pane of the PeopleSoft AutoPilot form, select the Include command line of the script that you want to edit.
2. Right-click and select Edit for a local script, or select Check Out & Edit for a repositited script.

---

**Important!** PeopleSoft AutoPilot opens the included script. The included script might be a parent of other scripts. To edit a child of the script that you are editing, you must select it and select Edit again.

---

3. Perform the necessary edits to the included script, and then select File, Close.
4. Save the changes to the included script.

If you edited a repositited script, PeopleSoft AutoPilot prompts you to reload the script.

5. To load the changes to the included script, click Yes.

PeopleSoft AutoPilot reloads the parent script with the changes to the included script.

6. Save the changes to the master script.

## Creating Variable Links

When you create variable links, you write the commands that enable PeopleSoft AutoPilot to pass variable values between two or more scripts. You declare a variable and set its value just as you do in a standalone script. However, by declaring the variable as external, you indicate that it can be linked to a variable in another script. PeopleSoft AutoPilot uses links to pass a value that you set in one variable in one script to another script.

The process of forging variable links requires that you write two or more scripts, declare certain variables as external, set a variable value, and then create the links between the variables. You can link variables between scripts that you maintain locally or between repositited scripts. If you break links between variables, you must recreate the links before your script will run.

To declare a variable as external:

1. From the Command menu, select the Variables option.
2. Type the name of a new variable in the unpopulated New Variable list.
3. Select the External option.



4. If you are declaring the variable but not setting a value, select Unknown/None from the Source of Value list, and click the Insert button.
5. If you are setting a value for the variable, select from the Source of Value list and the value selection list.
6. Click the Insert button.

---

**Note.** PeopleSoft AutoPilot includes the word External in the Declare command line in the Script pane.

---

## Declaring a Variable as External

You declare a variable as external only if you want to link it to a variable in another script to pass a value between the scripts. You declare an external variable just as you declare a variable in a standalone script. In PeopleSoft AutoPilot, you can select an option that declares the variable as external.

---

**Note.** You can also write a script with a local variable and then, later, update the variable and make it external. You might do this when you want to include the script with another script.

---

## Assigning a Default Value to an External Variable

You can set a default value for a variable, regardless of whether you select to declare it as external. By assigning a default value, you ensure that you can run your script in standalone mode, even if you make the variable external and link it to a variable in another script. If you link the variable with the default value to a variable in a parent script, PeopleSoft AutoPilot overrides the default value during playback, either with the value of the variable in the parent script, or with a null string if the variable in the parent script has no value.

To assign a default value to an external variable:

1. From the Command menu, select the Variables option.
2. In the PeopleSoft AutoPilot command pane, type the name of a variable in the New Variable list.
3. Select the External option.
4. In the Default list, type a value for the variable.
5. Click the Insert button.



## CHAPTER 8

# Playing Back the Script

This chapter provides an overview of script playback and discusses how to:

- Configure automatic script playback.
- Use manual script playback options.
- Run script playback.

---

## Understanding Script Playback

You can play back the script at any point, regardless of whether you are finished writing commands.

PeopleSoft AutoPilot offers the following options for playing back the script:

- From the beginning to the end without interruption.
- From a selected cursor position to the end without interruption.
- From the beginning of a selected script branch (node) to the end of the branch.
- From a selected cursor position line-by-line.
- From a selected position to a designated breakpoint.

You can stop playback at any time by selecting Stop from the Play menu or by clicking the Stop button on the toolbar.

You can script wait periods during playback. When you script a wait period, the playback stops for a length of time that you specify, and then it resumes. You can also pause playback indefinitely by pressing the PAUSE key on your keyboard. Resume play by pressing PAUSE again.

You can configure playback by selecting Options from the Tools menu and selecting the Playback tab. For example, you can set PeopleSoft AutoPilot to capture and display the results of a playback session. The results are presented as an event stream: a time-stamped, chronological record of each PeopleSoft AutoPilot and PeopleSoft EnterpriseOne software event that occurs during the session.

When you play the script, PeopleSoft AutoPilot stops the playback if an error occurs in the software. If you have configured playback to save and display test results, appears in the Test Results form to indicate the playback was unsuccessful. If PeopleSoft AutoPilot encounters no errors during playback, it displays a message indicating success.

---

## Configuring Automatic Script Playback

This section provides an overview of automatic script playback and discusses how to:

- Use playback during script creation.
- Store and display playback data.
- Handle breakpoints.
- Set playback speed.
- Set the Cancel Playback on Comm Error (cancel playback on communication error) option.
- Set the Log Variables on Script Failure option.
- Set an event stream,

## Understanding Automatic Script Playback

At any time, you can set or change your script playback configuration to have PeopleSoft AutoPilot play back certain features without your intervention. To configure script playback, you use the Options form, which you access by selecting Options from the Tools menu on the PeopleSoft AutoPilot form.

## Using Playback During Script Creation

When you select the Play Back while Creating Script option, PeopleSoft AutoPilot plays each command after you insert it in the script. To write a script without any delay caused by script playback, you should clear this option. Alternatively, to observe each command that you script, select this option.

## Storing and Displaying Playback Data

PeopleSoft AutoPilot enables you to store and display the results of each script playback. If you save results data, PeopleSoft AutoPilot saves script playback results as a binary large object in table F97214. If you display test results after playback, PeopleSoft AutoPilot automatically displays the Test Results form after playback. This form contains tabs that you use to review additional information about script playback.

If you select to save script playback results, you can view a history of playback results by selecting Results from the Tools menu.

### See Also

[Chapter 10, “Storing Scripts and Test Results,” Understanding Script Reporting, page 154](#)

## Handling Breakpoints

A breakpoint is a point in the script that halts playback until you manually continue it or cancel it. To play the script uninterrupted but keep any breakpoints that you have inserted, select the Ignore Breakpoints during Playback option.

## Setting Playback Speed

If you select the Accelerated Playback option, the system notifies PeopleSoft AutoPilot as soon as processing is complete so that playback can immediately continue. In general, you should select this option only when you are running relatively simple scripts that do not require the system to perform a large amount of processing.

## Setting the Cancel Playback on Comm Error Option

If you select the Cancel Playback on Comm Error option, and you experience software communication issues, PeopleSoft AutoPilot cancels the current script. If you do not select the Cancel Playback on Comm Error option, PeopleSoft AutoPilot attempts to complete the script.

## Setting the Log Variables on Script Failure Option

If you select the Log Variables on Script Failure option, PeopleSoft AutoPilot logs the final values of all variables in the event of a script failure.

When a script fails, the system writes the current value of variable assignments to the PeopleSoft AutoPilot results output. You use this output to analyze script failures. In the results database, you can view variable contents and immediately draw conclusions about the cause of a script failure. For example, the journal date variable value of 06/03/02 causes the script to fail if the year is not 2002. This is typical of the variable data that you might view. It is recommended that you select this option.

## Setting Event Stream

The term *event stream* refers to the flow of information from the software to PeopleSoft AutoPilot that occurs during playback. You select one of the following options to configure script playback to capture this information:

- None
- PeopleSoft warning and error messages
- Level 1 API calls
- All API call levels

---

## Using Manual Script Playback Options

This section provides an overview of manual script playback options and discusses how to:

- Use the Play from Top option.
- Use the Play from Cursor option.
- Use the Play Branch option.
- Play the script from a selected line command.
- Play the script to the next line command.
- Toggle a breakpoint.
- Continue to a breakpoint.
- Use the Comment/Wait option.
- Use script comments.
- Ignore breakpoints during playback.
- Stop playback.

## Understanding Manual Script Playback Options

After you configure script playback, you run playback using the Command menu or the playback buttons on the toolbar. You can play the script from the top without interruption, play the script from any selected spot to the end, play only a selected branch of the script, play back the script one command at a time, play the script to a breakpoint, or stop playback. You can insert wait periods in the script to delay playback for a set period of time before it resumes, or you can manually pause and resume script playback.

### Using the Play from Top Option

When you select Play from Top in the Play menu, script playback begins with the first command line and continues until the end of the script unless PeopleSoft AutoPilot encounters an error.

When you use this or any of the other playback functions, you can stop the playback by clicking the Stop button on the toolbar or by pressing the PAUSE button on the keyboard. Before you play back the entire script, remove any breakpoints that you inserted in the script.

### Using Play from Cursor Option

The Play from Cursor option enables you to select any spot in the script and then play the script to completion, if PeopleSoft AutoPilot does not encounter an error or a breakpoint.

### Using the Play Branch Option

The Play Branch option enables you play only a single script node that consists of one or more context commands and a series of action commands.

## Playing the Script from a Selected Line Command

To manually control playback from a chosen point in the script, you select the Stepping On option. You can play back either from the top, from a selected cursor position, or from a chosen branch of the script. You can then select how to play back the script, either one command line at a time or to a selected breakpoint.

## Playing the Script to the Next Line Command

After you select Stepping On and select to play back either from the top, from a selected cursor position, or from a selected branch, you select the Step Next option to play the script one command line at a time. Script playback does not proceed until you click the Step Next button on the toolbar, or select Step Next from the Play menu.

## Toggling a Breakpoint

A breakpoint is a command line that you select to stop playback until you resume or cancel playback. You can insert as many breakpoints in the script as you need. This enables you to isolate areas of the script and observe the playback.

You toggle the breakpoint by placing the insertion cursor after the line in the script where you want playback to break, right-clicking, and selecting Toggle Breakpoint. You can script multiple playback breakpoints. The breakpoint itself does not stop playback. You can do so only by clicking the Stop button.

## Continuing to a Breakpoint

After you select Stepping On and select a position from which to play the script, you can play back the script to a breakpoint. PeopleSoft AutoPilot plays all commands until it reaches the breakpoint.

## Using Wait Before Proceeding

You use the Comment/Wait option to script waiting periods, or pauses, in the playback. You can insert one or more wait commands anywhere in the Script pane. After the prescribed wait period has elapsed, playback resumes without your intervention. You can specify the duration of the wait. You can insert waits in the script that are of sufficient duration to simulate the amount of time required to actually enter data in a header or detail area.

Enter comments in the Comments field. Enter the period to wait in the Time (mSec) field. Enter the time in milliseconds. For example, if you want a 30 second pause, enter *30000*.

A Wait command produces a pause of determinate length. By contrast, when you press the PAUSE button on the keyboard, PeopleSoft AutoPilot pauses playback until you press the PAUSE button again.

## Using Script Comment

Using the Comment/Wait option, you can write brief comments that, for example, explain the reason for a Wait command. If you exchange scripts with someone else, you can use the comments to explain the actions that occur at a particular point in the script or to explain what the script is designed to test. You type your comment in the Comment field in the command pane and insert it in the script.

---

**Note.** PeopleSoft AutoPilot truncates the comment in the Script pane at 54 characters, including spaces.

---

PeopleSoft AutoPilot also enables you to cut or copy comments from other scripts or from other documents and paste them into the Comment field of the command pane. For example, if a comment that you insert in one script is applicable to several other scripts, you copy that comment and paste it into other scripts.

When you select the Comment/Wait option from the Command menu, the command pane also displays the following options:

- Wait until message window is closed
- Log To Test Manager
- Fail Script

Select the *Wait until message window is closed* option if the script clicks the Delete button in a form. If you select the option, PeopleSoft AutoPilot does not proceed with script playback until it has clicked OK on the Confirm Delete form.

Select the Log To Test Manager option if you plan to include a script as part of batch testing. You use the Test Manager tool to assemble script playlists for batch playback. If you select the Log To Test Manager option, the comments that you insert in a script are sent to Test Manager and included in a report after playback.

Select the Fail Script option if a critical event in your script caused the script to fail. If you select this option, PeopleSoft AutoPilot automatically fails the script at the point where you insert the command.

If you fail the script, PeopleSoft AutoPilot inserts a comment symbol in red.

You can use the Fail Script option in conjunction with logging comments to Test Manager by selecting both options. When you run a batch test, PeopleSoft AutoPilot fails the script and generates a summary report in Test Manager that lists any comments that you include about the failure.

## See Also

[Chapter 10, “Storing Scripts and Test Results,” Understanding Script Testing, page 160](#)

## Ignoring Breakpoints During Playback

You can preserve breakpoints that you have inserted in the script, but run the script one or more times without breakpoints. Rather than toggle the breakpoints on and off, you can select the Ignore Breakpoints during Playback option. When you want to run the script to the breakpoint that you designated, clear the Ignore Breakpoints during Playback option and play back the script.

## Stopping Playback

At any point during playback, you can stop the process by selecting the Stop option in the Play menu or by clicking the Stop button on the toolbar. If the script is playing, the Stop button appears red. When you click the Stop button, PeopleSoft AutoPilot displays a Script Playback Cancelled message, and the Stop button appears gray.

---

## Running Script Playback

This section provides an overview of script playback and discusses how to:

- Play the script from the top.
- Play the script from a particular cursor position.
- Play a branch of the script.
- Play the script to the next command.
- Pause script playback after beginning playback from the top.
- Pause script playback after beginning playback from a command line.
- Ignore breakpoints in the script.
- Toggle a breakpoint.
- Play the script to a breakpoint.
- Continue playback to a breakpoint.
- Ignore breakpoints during script playback.
- Insert a Wait command in the script.
- Fail a script.
- Set transaction times in the script.
- Insert a comment in the script.

## Understanding Script Playback

You run the various script playback functions in PeopleSoft AutoPilot using the options in the Play menu or the playback buttons on the toolbar. You can also use the Comment/Wait option to script one or more pauses in the playback and to insert comments in selected command lines in the script or as standalone lines. In addition, you can right-click a command line in the Script pane to toggle a breakpoint on and off.



## Pause Playback

You can use the PAUSE button on your keyboard to control script playback.

Pressing the PAUSE button during script playback pauses the playback. Press PAUSE again to resume playback.

PeopleSoft AutoPilot must have control of the screen when you press the PAUSE button. To verify that PeopleSoft AutoPilot has control of the screen, click anywhere in the PeopleSoft AutoPilot form.

## Ignoring Breakpoints in the Script

You can ignore breakpoints that you activate in the script if you want to play back the script without interruptions but do not want to clear the breakpoints. To do so, select the Ignore Breakpoints option from the Play menu. If you need to stop playback at the breakpoints, select the Ignore Breakpoints option again to disable it.

## Inserting a Comment in the Script

You can also use the Comment/Wait option to insert into the Script pane comments about the command line that you chose or general comments about the script, including its purpose. To include the script in the batch testing that Test Manager runs, and you want the comments to appear in a summary report after batch testing, select the Log To Test Manager option.

## Playing the Script from the Top

Before you play the script from the top, close all open applications. If you do not close all applications, playback might not finish successfully. PeopleSoft AutoPilot must have control the mouse during the playback process. Do not move the mouse or attempt to open any applications or programs while playback is running.

To play the entire script from the top, in the Script pane of the PeopleSoft AutoPilot form, verify that you have completed the following steps:

1. Remove any breakpoints.
2. Disable the Stepping On button.
3. Close all PeopleSoft windows.
4. In the Play menu, select the Play from Top option.

---

**Note.** You can also start the playback process by pressing the F5 key on the keyboard.

---

## Playing the Script from a Particular Cursor Position

You use the Play From Cursor option to begin playback from the position of the cursor, rather than from the top of the script. Before you play back the script from cursor, ensure that the PeopleSoft EnterpriseOne software is open to the appropriate place. For example, if you are starting playback on a particular form, ensure that the form is open.

To play back the script from a particular cursor position:

1. In the Script pane in the PeopleSoft AutoPilot form, select a command line from which you want to play back the script.
2. Select the command line.
3. In the Play menu, select the Play From Cursor option.

## Playing a Branch of the Script

You can use the Play Branch feature to play a selected branch of a script. The script plays from the selected line to the insertion cursor.

To play a branch of the script:

1. In the Script pane in the PeopleSoft AutoPilot form, select the first command line in a branch.
2. In the Play menu, select the Play Branch option.

## Playing the Script to the Next Command

After you select the Stepping On option, you select a command line from which to play back the script—from the top, from a branch of the script, or from a particular cursor position. You can then play the script one line at a time or until the next breakpoint.

To play the script back one line at a time:

1. In the Script pane in the PeopleSoft AutoPilot form, select a command line from which you want to play back the script.
  2. In the Play menu, select the Stepping On option.
  3. In the Play menu, select one of the following buttons:
    - Play from Top
    - Play Branch
    - Play from Cursor
- PeopleSoft AutoPilot enables the Step Next, Continue to Breakpoint, and Stop buttons.
4. To proceed to the next line, click Step Next.
  5. To play to the next breakpoint in the script, click Continue to Breakpoint.
  6. To discontinue the playback, click Stop.

## Pausing Script Playback After Beginning Playback from the Top

Access the PeopleSoft AutoPilot form.

To pause script playback after beginning playback from the top:

1. In the Play menu, select the Play from Top option.
2. At the point where you want to halt playback, press the PAUSE button on the keyboard.
3. To continue playback to the next command line, press the PAUSE button again.

---

**Note.** PeopleSoft AutoPilot does not proceed to the next command line until you press the PAUSE button again.

---

4. To stop playback, click the Stop button on the toolbar.

## Pausing Script Playback After Beginning Playback from a Command Line

Access the PeopleSoft AutoPilot form.

To pause script playback after beginning playback from a command line:

1. In the Script pane in the PeopleSoft AutoPilot form, select a command line where you want to play back the script.
2. Click the Play from Cursor To End Of Script button.
3. Press the PAUSE button on the keyboard.

PeopleSoft AutoPilot runs the selected command line and advances to the next command line.

4. Continue through the script one command line at a time, pressing the PAUSE button each time that you want to run a command.

## Toggling a Breakpoint

To have the script play only to a predetermined command line in the script, you can toggle a breakpoint by highlighting it and then right-clicking the command line at which you want playback to break. You can toggle as many breakpoints as you like. For example, you might toggle a breakpoint when you have created a lengthy script and want to play back only a portion of it rather than the entire script.

When you toggle a breakpoint and then play back the script, the playback proceeds to the line on which you set the breakpoint, and then it halts until you either stop playback or continue it to another breakpoint. You can also select and right-click a command line to remove a breakpoint.

To toggle a breakpoint:

1. In the Script pane of the PeopleSoft AutoPilot form, select a playback breakpoint by selecting a line in the script.
2. Right-click.
3. Select the Toggle Breakpoint option.

PeopleSoft AutoPilot inserts the breakpoint to the script.

4. To remove the breakpoint, select a command line where you entered a breakpoint.
5. Right-click.
6. Select the Toggle Breakpoint option.

PeopleSoft AutoPilot removes the breakpoint.

## Playing the Script to a Breakpoint

After you have toggled a breakpoint, you can play back the script, either from the top or from a particular cursor position. When PeopleSoft AutoPilot reaches the command line that contains the breakpoint, playback halts. However, PeopleSoft AutoPilot does not cancel playback. To continue scripting, or to play back the script differently, you must click the Stop button to cancel playback.

To play the script to a breakpoint:

1. In the Script pane in the PeopleSoft AutoPilot form, toggle a breakpoint in the script.
2. In the Play menu, select the Play from Top option.

3. If the script plays to the breakpoint and you want to continue scripting, toggle off the breakpoint.
4. In the Play menu, select Stop.

## Continuing Playback to a Breakpoint

After you toggle one or more breakpoints and select stepping, you can play the script back from breakpoint to breakpoint by clicking the Continue to Breakpoint button on the toolbar.

To continue playback to a breakpoint:

1. In the Script pane in PeopleSoft AutoPilot, select a playback breakpoint by selecting a line in the script.
2. Right-click.
3. Select Toggle Breakpoint to insert the breakpoint in the script.
4. Select a point in the script where you want to play back the script.
5. In the Play menu, select the Stepping On option.
6. Click the Play from Cursor To End Of Script button.
7. Click Continue to Breakpoint.

---

**Note.** You can set as many breakpoints in the script as necessary and click Continue to Breakpoint each time that playback reaches one.

---

## Ignoring Breakpoints During Script Playback

Access the PeopleSoft AutoPilot form.

To ignore breakpoints during script playback:

1. In the PeopleSoft AutoPilot Script pane, toggle on one or more breakpoints in the script.
2. Click the Ignore Breakpoints button on the PeopleSoft AutoPilot toolbar.

## Inserting a Wait Command in the Script

If you insert a breakpoint in the script, playback halts when PeopleSoft AutoPilot reaches the breakpoint. Playback does not resume or stop without your intervention. Alternatively, if you select Comment/Wait in the Command menu, you can script a specified wait period, or pause, at a predetermined script command line. When the playback reaches this command line, the wait occurs, and then playback proceeds.

To insert a Wait command in the script:

1. In the Script pane in the PeopleSoft AutoPilot form, select a command line in the script.
2. In the Command menu, select the Comment/Wait option.
3. Press the TAB key or place the cursor in the unpopulated Time (msec) list.
4. Type a time, in milliseconds, for the wait.

---

**Note.** Do not use commas when you type the time of the wait.

---

5. Click the Insert button.
6. Run a playback command.

## Failing a Script

You can automatically fail a script by activating the Fail Script option. This option appears in the command pane when you click the Comment/Wait button. Activate the Fail Script option to include the script in a batch that Test Manager runs.

To fail a script:

1. In the Script pane of the PeopleSoft AutoPilot form, place the insertion cursor at the point that you want to fail the script.
2. In the Command menu, select Comment/Wait.
3. In the command pane, select the Fail Script option and click the Insert button.

PeopleSoft AutoPilot inserts a red comment symbol in the Script pane to indicate that the script fails at that point.

## Setting Transaction Times in the Script

A transaction is a series of events, bounded by a start and end point. You can also insert comments in the script to measure playback transaction time. You use the Comment/Wait option to assign a name to the transaction, to insert a starting point (such as launching an application), and to insert a finishing point (such as closing the program). Setting transaction times in the script provides important information about the time that the software requires to run a series of commands.

To set transaction times in the script:

1. In the Script pane of the PeopleSoft AutoPilot form, determine the command line that represents the start of the transaction, and then place the insertion cursor directly above it.
2. In the Command menu, select the Comment/Wait option.
3. In the unpopulated Comment list of the PeopleSoft AutoPilot command pane, enter *Start*, press SPACEBAR, and then enter a name for the transaction.
4. Click the Insert button.

PeopleSoft AutoPilot inserts a command line that marks the start of the transaction.

5. Determine the command line that represents the end of the transaction, and then place the insertion cursor after it.
6. In the Command menu, select the Comment/Wait option.
7. In the unpopulated Comment list of the PeopleSoft AutoPilot command pane, type *End*, press SPACEBAR, and type the name of the transaction.

PeopleSoft AutoPilot inserts a command line that marks the end of the transaction.

---

**Note.** The name that you assign to the end of the transaction must exactly match the name that you assign to the start of the transaction.

---

8. Click Insert.

## Inserting a Comment in the Script

Access the PeopleSoft AutoPilot form.

To insert a comment in the script:

1. In the Script pane of the PeopleSoft AutoPilot form, place the insertion cursor at the point in the script in which you want the comment to appear.
2. In the Command menu, select the Comment/Wait option.
3. In the unpopulated Comment list of the PeopleSoft AutoPilot command pane, type a comment.
4. Select the Log to Test Manager option to have PeopleSoft AutoPilot include the comment in a summary report after testing.
5. Click Insert.

**See Also**

*Virtual PeopleSoft AutoPilot Guide*

## CHAPTER 9

# Creating a Sample PeopleSoft AutoPilot Script

This chapter provides an overview of the sample PeopleSoft AutoPilot script and discusses how to create the sample PeopleSoft AutoPilot script.

---

## Understanding the Sample PeopleSoft AutoPilot Script

Although PeopleSoft AutoPilot can be used to create a script to verify any application, this sample script uses the A/P Standard Voucher Entry (accounts payable standard voucher entry) application (P0411). This section includes step-by-step instructions for developing a sample script for the application. This sample script does not provide examples of every function or feature of PeopleSoft AutoPilot. For example, this script tests an interactive application and does not launch a universal batch engine (UBE). Consult other sections of this guide if you need information about a function that is not included in the sample script.

The steps for writing a script vary from one script to another. The precise steps that you include in a script depend on your use of a particular application.

---

## Creating the Sample PeopleSoft AutoPilot Script

This section provides an overview of sample script creation and discusses how to:

- Launch an application and form.
- Declare a variable.
- Add a new form.
- Type data in a header control.
- Create a valid values list.
- Type data in a grid column.
- Update the repeat count.
- Update the Database.
- Set the value of a variable.
- Return to a previous form.
- Enter data to a Query By Example (QBE) line.
- Find records.
- Select records and delete them from the database.

- Complete the script.

## Understanding Sample Script Creation

You use PeopleSoft AutoPilot to create customized scripts that apply to the applications that you most often use. The sample script presented in this chapter illustrates how you use many of the commands that are included in PeopleSoft AutoPilot. However, the sample script does not include all context and action commands, and it does not represent a definitive method for using PeopleSoft AutoPilot. For example, this script tests functions that you perform in an interactive application. You can also test the launch and submission of a UBE, which requires that you write a different set of commands.

## Launching an Application and Form

Launching the application from PeopleSoft AutoPilot establishes one basic scripting context that you need for many other scripting commands that you insert and run. Launching an application does not have to be the first command that you script in a PeopleSoft AutoPilot session. However, the Application command is often at least one of the first commands that you script.

Selecting an application in PeopleSoft AutoPilot also requires that you select a form that is part of the application. After you launch an application from PeopleSoft AutoPilot, you can script a variety of additional context and action commands. For example, after you have established the context as a form, you can then script inputs for header controls, grid columns, and QBE lines. After you establish one of these contexts, you can script clicking buttons to access different forms or to perform functions within the active form.

To launch an application and form:

1. From the desktop or the appropriate directory, launch PeopleSoft AutoPilot.  
The PeopleSoft AutoPilot splash screen appears, followed by the PeopleSoft AutoPilot form.
2. In the File menu, select the New option.  
The command pane and Script pane are unpopulated.
3. In the Command menu, select Application.  
The Application list is populated, while the Menu list remains unpopulated.
4. Click an application code, such as P0411 for the A/P Standard Voucher Entry application.  
PeopleSoft AutoPilot populates the Menu list with items that appear under the following headings:
  - Fast Path, which corresponds to the fast path command that PeopleSoft AutoPilot will use to launch the application.
  - Menu Text; for example, Standard Voucher Entry.
  - Version, which specifies which version of the application will be launched.
5. Click a Fast Path command to launch the application, such as 3/G0411, Standard Voucher Entry, version ZJDE0001.
6. Click the Insert button in the command pane.

---

**Note.** If you the Playback button on the toolbar is activated, PeopleSoft AutoPilot launches the application, and the form specified in the Fast Path appears (in this case, the Supplier Ledger Inquiry form).

---



## Declaring a Variable

Before you can use a variable as a source of input, you must first declare it, or give it a name, to specify the place in which you store the value. You can declare a variable at any point in the script, but after you declare a variable, you can place it at the top of the script to make it global. That way, you can set its value at any point in the script. If you make the variable global, you can launch multiple applications within the script and use the stored value in any of the applications. If you declare a variable after you launch an application, you can drag the Declare command for the variable to the top of the script to make it global.

For this script, you use a previous document number to retrieve voucher entry data. You declare the variable early in the script so that you have a place already established to store the previous document number as soon as you know what it is.

To declare a variable:

1. In the Command menu, select Variables.
2. Type a name for the variable in the New Variable field.  
For the sample script, call the variable *Previous Document #*.
3. In the Source of Value list, select Unknown/None.

---

**Note.** Selecting Unknown/None indicates that you merely want to give the variable a name at this point. You do not yet set its value.

---

4. Click the Insert button.  
PeopleSoft AutoPilot inserts the Declare command for the variable after the Supplier Ledger Inquiry Form command line. At this point, you can use the variable only within that Form node because it is attached to the Form node.
5. Click the Declare command line in the Script pane to highlight it.
6. Drag the Declare command line until it is on top of the Application command line. When the indicator arrow is pointing up, release the mouse button.  
You have now attached the Declare command line to the Begin Script node, and you can use a value that you set for the variable at any point in the script.

## Adding a New Form

For this sample script, you move from the Supplier Ledger Inquiry form to the Enter Voucher - Payment Information form. To do so, you script clicking the Add button to change forms. You also select Enter Voucher - Payment Information from the Next Form list so that the Form command line in PeopleSoft AutoPilot matches the active form.

To add a new form:

1. From the Command menu, with the Playback button activated, select the Press Toolbar Button option.
2. In the Button list, select the Press Standard Button option.

---

**Note.** The options listed for Press Standard Button in PeopleSoft AutoPilot correspond to the buttons on the toolbar in the form.

---

3. Click Add.

4. In the Next Form list, select the name of the form that appears in the software when you click Add—the Enter Voucher - Payment Information form in this example.
5. Click the Insert button.

PeopleSoft AutoPilot adds the commands that you insert in the Script pane. In the playback mode, the Enter Voucher - Payment Information form appears in the software.

---

**Note.** If you select the Unknown/None option from the Next Form list (step 4), you must complete the following additional steps. You must script a Form command line that corresponds to the form that is active in the software. At all times, the most current Form command line in the PeopleSoft AutoPilot Script pane must correspond to the form that is active in the software. If the command line in the Script pane does not correspond to the active form in the software, you cannot continue scripting.

---

6. With the Playback button activated so that the software is active, note the name of the software form that is active—the Enter Voucher - Payment Information form in this example.
7. In the Command menu, select Form.  
The system displays a list of all forms within the current application.
8. Select the name of the active form from the Form list.  
In this example, select Enter Voucher - Payment Information.
9. Click Insert.  
In the Script pane, a Form command line appears that includes the name of the selected form.

## Typing Data in a Header Control

Continue the sample script by adding commands to type inputs in header controls. Like the Application and Form commands, the Header command is a context command; it establishes the context in which you take further actions.

To type data in a header control:

1. In the Command menu, select the Set Header Control Value option.
2. In the Header Control list, select the name of a header control where you want to input data—for example, Company.

---

**Note.** The header control that you select in PeopleSoft AutoPilot is highlighted with a BlueCue.

---

3. Select one of the following as the input source for the control:

- Literal Value
- Valid Values List
- Variable
- UDC Visual Assist Value
- Form Interconnect Visual Assist Value
- Clear
- Set Focus

In this sample script, select Literal Value. When you select an input source, PeopleSoft AutoPilot provides a caption for the value selection list. The caption depends on the source of input.

4. Click inside the unpopulated Literal Value list and type *I*.
5. Click the Insert button.

PeopleSoft AutoPilot types the scripted input in the Company header control in the Enter Voucher - Payment Information form. PeopleSoft AutoPilot encloses literal values in quotes in the Script pane.

6. Continue to script inputs in header controls by selecting the Set Header Control Value option in the Command menu, and then selecting a control, a source of input, and a value.

## Creating a Valid Values List

For this sample script, you script inputs in the header control Long Address Number. However, instead of entering a literal value in the control, use a valid values list as the input source.

A valid values list consists of values that you collect and store under a specified name. You use a valid values list if, for example, you need to input more than one value in a header or grid column. The following types of valid values lists exist:

- List of literal values
- Simple database query

For this script, you create a list of literal values, which contains values that you assign. With a simple database query, PeopleSoft AutoPilot draws the values from a database that you select. You can continue to enter literal values in each header control. However, you can include as many values in the valid values list as you want. Each time that you play back the script, PeopleSoft AutoPilot automatically inserts one of the values in the appropriate header control.

If the value that you want to insert in a header or grid column is constant, select a literal value. However, if the value is likely to change, you can create a valid values list so that PeopleSoft AutoPilot inserts a new value each time. In this example, the long address number is different for each vendor, so create a valid values list and give it the name Vendors.

To create a valid values list:

1. In the Tools menu, select Generate Valid Values List.
2. Select the List of Literal Values option in the Select Data File Type dialog box.
3. Click Next.
4. Type a file name in the File Properties list; for this example, name the list Vendors.
5. Enter one or more values in the Enter Values list.  
The values should appear vertically stacked in the box.
6. Click Finish.
7. In the Command menu, select Set Header Control Value.
8. In the Header Control list, select the name of a header control where you want to input data; for example, Long Address Number.
9. Select Valid Values List as the source of input.
10. Select the name Vendors, which is the name of the valid values list that you created.
11. Click the Insert button.

PeopleSoft AutoPilot types the first value in the list in the Long Address Number control in the Enter Voucher - Payment Information form. PeopleSoft AutoPilot identifies the valid values list in the Script pane with a backslash, the name that you assigned to the list, and the extension .atd.

12. Complete the following tasks:

- Script an input to the header control Business Unit. Use a literal value of *I*.
- Script an input to the header control DateForGLandVoucherJULIA. Use a literal value of *063005*.
- Script an input to header control DocVoucherInvoiceE, using a list of literal values that you named Sequential 5-Digit Numbers.

When you have inserted inputs in each of the header controls, the Script pane should contain five Type To action commands within the Header node.

## Typing Data in a Grid Column

Next, script inputs in the grid columns. For this sample script, make voucher payment inputs in the grid columns. The Grid command, like the Header command, is a context command; it establishes the grid column in the form as the environment where you take additional actions.

The names of the grid columns in the Grid Column list of the PeopleSoft AutoPilot command pane match the names of the grid columns on the form. When you select the name of a grid column from the Grid Column list, a BlueClue appears as an arrow over the corresponding grid column in the form.

For the sample script, enter vouchers in multiple rows of the grid columns of the form. You can enter literal values in each grid row, but it is easier to create a valid values list and then update the repeat count of the command line. For this script, type inputs in the Gross Amount and Remark grid columns.

To type data to a grid column:

1. Create a list of literal values that contains five values and name it Random Dollar Amounts.  
This list contains gross amounts paid to vendors.
2. From the Command menu, select Set Grid Cell Value.  
The command pane includes the following lists:
  - Grid Column
  - Source of Input
3. Select the name of a grid column from the Grid Column list; for example, select Gross Amount.
4. Select Valid Values List as the source of input.
5. Select Random Dollar Amounts.
6. Click the Insert button.
7. Repeat steps 1 through 6 to script an input in the Remarks column.

Name the list of literal values Random JE Explanations. This list contains explanations for each entry in the Gross Amount column, such as Rent.

After you insert inputs in each of the grid columns, the Script pane should contain two Type To action commands within the Detail Information (grid) node.

## Updating the Repeat Count

Because the source of input includes two lists of literal values that contain five values each, you change the repeat count for this node so that each of the five values is input in the grid during playback.

To update the repeat count :

1. If the playback mode is activated, disable it by clicking the Playback button in the toolbar.
2. In the Script pane, select Detail Information, which is the node that you need to update.

In the command pane, review the following lists:

- Define repeat count from
- Repeat Count

3. Select Literal Value from the Define repeat count list.
4. In the Repeat Count list, type a number.

In this case, you type 5, because you want to script entering five separate values that you included in your valid values lists.

5. Click Update.

---

**Note.** The repeat count in the Detail Information node is now 5. If you play back the script, PeopleSoft AutoPilot loops through this node five times. Each time it loops, PeopleSoft AutoPilot inserts in the Gross Amount and Remark grid columns a different value from the valid values lists.

---

6. Click the Playback button on the toolbar to select the playback function.

## Updating the Database

After you script the entries in the Enter Voucher - Payment Information form, you update the database with the new entries by clicking OK. For this script, you also write a new Form command because clicking OK opens a new form.

In the new form, you enter data, click OK to add it to the database, and then return to the previous form by writing a new Form command.

---

**Note.** Before performing these steps, create a list of literal values to use in Account Number column of the Enter Voucher - G/L Distribution form. Name this list Random JE Account Numbers.

---

See [Chapter 6, “Scripting Actions,” Creating a List of Literal Values, page 61](#).

To update the database and confirm a new form:

1. In the Command menu, select the Press Toolbar Button option.

Review the following lists in the command pane:

- Button
- Next Form

The Button list in the command pane has a default value of Press Standard Button. For this script, under Press Standard Button, the options match the following buttons on a form:

- OK
- Delete

- Cancel
2. Select the OK option from the Button list.
  3. From the Next Form list, select the form that follows when a user clicks OK for this application and version.  
For the sample script, you select Enter Voucher - G/L Distribution.
  4. Click Insert.  
The Enter Voucher - G/L Distribution form appears. Credit the full voucher payment amount to a particular account number. After you credit the account, update the database, and then return to the Enter Voucher - Payment Information form. To do so, complete these tasks:
    - a. Enter an account number in the appropriate grid column in the G/L Distribution form.
    - b. Click the OK button to update the database.
    - c. Confirm the Enter Voucher - Payment Information form.

---

**Note.** You can create the list of literal values before you write the script. You can also enter a valid literal value in the Account Number column.

---

To enter data, update the database, and return to a previous form:

1. Create a list of literal values.  
Name the list Random JE Account Numbers. This list contains valid G/L bank account numbers.
2. In the Command menu, select Set Grid Cell Value.
3. In the Grid Column list, select Account Number.
4. Select Valid Values List as the source of input.
5. Select Random JE Account Numbers.
6. Click Insert.  
After you enter data in the Account Number grid column in the Enter Voucher - G/L Distribution form, and distribute the amount from the Enter Voucher - Payment Information to the indicated account number, you update the database and return to the Enter Voucher - Payment Information form.
7. In the Command menu, select the Press Toolbar Button option.
8. Select the Standard Button option.
9. Select OK.
10. In the Next Form list, select Enter Voucher - Payment Information, and then click the Insert button.

## Setting the Value of a Variable

After you return to the Enter Voucher - Payment Information form, retrieve the previous document number. You will use it later to search in the Supplier Ledger Inquiry form for the data you input in the Gross Amount and Remarks grid columns in the Enter Voucher - Payment Information form. You need to retrieve and store the document number. You accomplish this by assigning its value to the variable that you declared earlier. The variable that you declared earlier has only a name, Previous Document #. Its value must be derived from a source that you select.

To set the value of a declared variable:

1. In the Command menu, select Variables.

2. In the Existing Variable list, select the name of the variable that you declared earlier in the script.
3. In the Source of Value list, select one of the following:

- Literal Value
- Valid Values List
- Variable
- Header Control Data
- Grid Cell Data

In this sample script, retrieve and store the previous Enter Voucher - Payment Information document number. Because the previous document number appears in a header control, select Header Control Data as the source of value.

4. Select or type a value.

In this case, PeopleSoft AutoPilot populates the value selection list with the names of the header controls in the Enter Voucher - Payment Information form. The value selection list, therefore, is called Header Control. Select Previous Document.

5. Click Insert.

You have now assigned a value to the variable that you named Previous Document #. You have told PeopleSoft AutoPilot to derive that value from the header control Previous Document in the form Enter Voucher - Payment Information. PeopleSoft AutoPilot stores that value in your declared variable. Because you previously made the variable global, you can use this value at any point that you select in the script.

## Returning to a Previous Form

When you return to the previous form, enter the value of the variable in the QBE line on the Supplier Ledger Inquiry form. In the sample script, click Cancel to return to the Supplier Ledger Inquiry form. Select the Press Toolbar Button option from the Command menu and select Cancel from the Press Standard Button options in the Button list. You then select the next form that appears when you click the Cancel button. Therefore, you select Supplier Ledger Inquiry from the Next Form list.

To return to a previous form:

1. In the Command menu, select Press Toolbar Button.
2. Select the Cancel option from Standard Button in the Button list.
3. Select the Supplier Ledger Inquiry form from the Next Form list.
4. Click the Insert button.

In the software, the Supplier Ledger Inquiry form becomes active. In the PeopleSoft AutoPilot Script pane, the Form command line shows Supplier Ledger Inquiry.

## Entering Data to a QBE Line

Because you have assigned a value to the declared variable, Previous Document #, you can now use the value that you stored.

You decide to enter the stored value, the previous document number, to the QBE line of the Supplier Ledger Inquiry form, then script clicking the Find button to retrieve the values that you entered to the grid in the Enter Voucher - Payment Information form.

To enter data in a QBE line:

1. In the Command menu, select Set QBE Cell Value.
2. Select the grid column Document Voucher Invoice Entry from the Grid Column list.
3. Select Variable from the Source of Input list.
4. In the value selection list, select the name of the variable that you declared; remember, it also now contains the value that you set.
5. Click the Insert button.

PeopleSoft AutoPilot inputs the previous Enter Voucher - Payment Information document number, which you stored in the variable that you named Previous Document #, in the QBE line of the grid, in the Document Voucher Invoice Entry column.

## Finding Records

To find the values that you entered in the Enter Voucher - Payment Information form, click Find. Because you do not move to a new form, do not select another form from the Next Form list before clicking the Insert button.

To find records:

1. In the Command menu, select the Press Toolbar Button option.
2. Select Standard Button in the Button list, and then select Find.

You do not need to select an option from the Next Form list because you are remaining on the Supplier Ledger Inquiry form.

3. Click the Insert button.

The grid contains the voucher entries that relate to the document number that you input in the Document Voucher Invoice Entry grid column in the QBE line.

## Selecting Records and Deleting Them from the Database

Next, script the deletion of records from the database. These records appear in the detail area of the Supplier Ledger Inquiry form because you scripted clicking the Find button in the previous task. To delete these records, you must select them, and then script clicking Delete.

When the script plays, the records that you select to delete in this task are actually deleted, just as they would be in a live session. Before you perform this task, verify that you are in a test environment and do not click the Insert button until you are sure that you have selected the correct records to delete. PeopleSoft AutoPilot automatically clicks OK in the Confirm Delete dialog box that appears when you have selected a grid line for deletion. You cannot click OK or Cancel.

To select records and delete them from the database:

1. In the Command menu, select the Select Grid Row option.
2. In the command pane, select from the Source of Row Number list.
3. Type a literal value of 1 in the value selection list.

By typing this value, you select the row that contains the record that you want to delete.

4. Select the single-click option in the Action on grid row list.

This command selects the row in the detail area of the grid.

5. Click the Insert button.
6. In the Command menu, select the Press Toolbar Button option.



7. Click Select Standard Button in the Button list, and then select Delete.
8. Click the Insert button.

PeopleSoft AutoPilot clicks OK on the Confirm Delete form that appears in the software and deletes the record that you selected. You can repeat this command as many times as necessary to delete any and all records that you need to delete.

## Completing the Script

At the conclusion of the scripting, exit all open applications. For the sample script, click the Cancel button. Because you started on the Supplier Ledger Inquiry form, this command enables you to exit the A/P Standard Voucher Entry program (P0411) and complete the script.

To complete the script:

1. In the Command menu, select the Press Toolbar Button option.
2. Select Standard Button, and then select Cancel.

Because you are exiting the application, do not select an option from the Next Form list.

3. Click Insert.

---

**Note.** By clicking Cancel, you return to the starting point of PeopleSoft EnterpriseOne software. You should end each scripting session by clicking Cancel or Close, especially if you intend to create additional scripts. Leaving windows open can impede playback of scripts and make it difficult to write additional scripts.

---

4. In the File menu, select Save.
5. Type a file name in the File Name field.
6. Click Save.

You have completed the sample script. While creating the script, you launched an application, selected a form and version associated with the application, and accessed forms by clicking buttons. You typed data in header controls, grid columns, and QBE lines. You derived that data from literal values that you typed in the value selection list and from valid values lists that you created and then selected from the value selection list. You added the data to the database, retrieved it, and deleted it. You declared a variable, set its value, and used it as a source of input in a QBE line. At the end, you canceled the application and saved the script.



## CHAPTER 10

# Storing Scripts and Test Results

This chapter provides an overview of storing scripts and test results and discusses how to:

- Work with the script repository.
- Work with script reporting.
- Work with PeopleSoft AutoPilot Test Manager.
- Manage script testing.

---

## Understanding Storing Scripts and Test Results

This section provides an overview of scripts and discusses:

- Script overview.
- Script repository.
- Script categorization.
- Property pages for scripts.
- Naming conventions for saved scripts.
- Add to Repository command.
- Browse Repository Scripts command.
- Deletion of scripts.
- Get Copy command.
- Checkout command.
- Undo Checkout command.
- My Checkouts form.
- Check In command.
- Where Included command.

### Script Overview

Scripts that you write are reusable, dynamic objects that continue to be useful after you complete them. PeopleSoft AutoPilot enables you to write and run scripts. You can also include those scripts in a base of knowledge about the software and manage batch testing of scripts.

The script repository is a key component of the PeopleSoft AutoPilot knowledge base. The repository is a database of scripts. It is stable because repositored scripts are controlled copies that can be changed only by the owner or an administrator who has permissions. The database is varied because many people with different areas expertise can contribute to it. Finally, the database is organized because you can assign defining properties to each script that you reposit. These properties enable you to categorize scripts by application, for example.

Capturing and storing test results is another important way in which PeopleSoft AutoPilot enables you to build a knowledge base about the software. If you configure PeopleSoft AutoPilot to capture playback results, it generates an event stream during playback. The event stream is a chronological, time-stamped record of PeopleSoft AutoPilot and PeopleSoft EnterpriseOne events that occur during playback. PeopleSoft AutoPilot stores these test results locally and in a repository, the F97214 table. You can use these results to troubleshoot PeopleSoft EnterpriseOne processes. For example, you might identify a processing error or isolate an error message.

The results repository is an important part of the automated testing process. To analyze playback events in detail, you can import an event stream from the repository to the Analyzer Tool. You can also import an event stream to Virtual Script Editor, which is part of Virtual PeopleSoft AutoPilot. Using Virtual Script Editor, you can generate from the event stream a virtual script. You can use the virtual script on a single workstation to simulate multiple users.

PeopleSoft AutoPilot also enables you to manage the testing of scripts. Using Test Manager, you can create playlists of locally saved and repositored scripts and conduct batch testing. This frees you from the time-consuming task of running one test at a time. You can use Test Manager to conduct testing of an entire suite of applications in one session.

## Script Repository

The script repository is similar to a library. It is a centralized location where you can find and retrieve scripts. You can search for scripts in the repository by browsing through all of the scripts that are available or by focusing your search on, for example, scripts that test a particular suite of applications.

In addition, like a library, the script repository acquires new materials. Each time that you create a script, you can assign distinguishing properties to it and then add it to the repository, from which it can be retrieved and viewed by others. Scripts can circulate freely among those who have access to the system, or you can assign levels of security to scripts to restrict their use.

Unlike a library, you can modify the materials that you remove from the repository. For example, you might check out a script from the repository to modify it. The PeopleSoft AutoPilot script repository enables a dynamic interchange between the people who use it.

You can use the script repository to build a database of scripts. You use PeopleSoft AutoPilot to:

- Categorize scripts according to a set of user-defined criteria.
- Identify scripts with unique names.
- Add scripts to the repository.
- Browse for scripts.
- Check scripts in or out of the repository.
- Retrieve copies of scripts.
- Modify scripts.
- Assign security to scripts.
- Track changes that you make to scripts.

- Identify scripts that are included in other scripts.

### See Also

*EnterpriseOne Tools 8.94 PeopleBook: PeopleSoft Analyzer Tool*

*EnterpriseOne Tools 8.94 PeopleBook: PeopleSoft Virtual AutoPilot*

## Script Categorization

Like a librarian who adds books to a library collection, when you add a script to the repository, you assign it properties, such as title, description, the application that the script tests, the purpose of the test, and so on. The properties pages that are attached to each repository script provide important summary information for users who check out a script, and they provide a way for you to categorize scripts and make them easier to find.

Categorizing scripts also facilitates running batch test scripts using Test Manager. Using Test Manager, you can browse the repository for scripts in a particular category, add them to a playlist, and automatically play them back.

Access the Script Properties form by selecting Properties from the File menu. The Script Properties form contains controls with scroll buttons.

The combo boxes contain user-defined values. You select from these values to categorize a script. To ensure that the information in the database is consistent, reliable, and easy to access by browsing, use a consistent set of user-defined values, which you maintain in table F0004 and table F0005, rather than using individual user text entries.

You can add to the values that appear in the combo boxes by using the User Defined Codes program (P0004A). The following table lists the relevant user-defined codes (UDCs) that PeopleSoft AutoPilot uses to populate the combo boxes in the Script Properties form:

Product Code in P0004A	User-Defined Code in P0004A	Combo Box in Script Properties Form in PeopleSoft AutoPilot
98 (technical tools)	SY (system code)	System Code field on General tab
H97 (benchmarking/performance)	DN (department name)	Department field on General tab
H97	GU (general usage)	General Usage field on General tab
H97	DU (detail usage)	Detail Usage control on General tab
H97	OT (other)	Test Case control on Details tab

**Note.** The user-defined combo box values also appear in the Select Script form and the Add Script to Repository form.

## Property Pages for Scripts

As you add a script to the repository, you should complete property pages that provide fundamental information about it, such as title, description, owner, the application that the script tests, and so on. Completing property pages also enables you to classify a script as part of a large-scale testing effort. For example, you can designate the following script properties to include the script in a suite of scripts:

- System code.
- Department.
- General use, such as benchmarking.
- Detailed use, such as batch applications.

You document the properties of a script by entering information in the Script Properties form. When you save the script, PeopleSoft AutoPilot saves the information along with the script.

When you add the script to the repository, PeopleSoft AutoPilot saves the property page data in the database. This data loads when you check out a script from the repository, and it overwrites any subsequent property page changes that you might have made in the local script.

## General Tab

The General tab of the Script Properties form contains a series of fields in which you enter data that defines the script.

The information that you enter on this tab provides baseline information about the script and its origins. You enter data in the following fields on the General tab of the Script Properties form:

Field	Description
Title	Script title that PeopleSoft AutoPilot automatically enters when you save the script. You can change the title after you check in the script.
Description	Brief description of the script, such as the function that it tests.
Main Application	The primary application that the script tests.
Owner	Script owner, automatically identified as the person who adds the script to the repository. You can change the owner after you check in the script.
System Code	A user-defined reporting system code.
Department	A user-defined department or group name.
General	The general testing purpose of the script, such as benchmarking. The values for this parameter are user-defined.
Detail	The particular testing purpose of the script, such as testing batch applications. The values for this parameter are user-defined.
Reference Number	A code that identifies the script. For example, you might use this code to enter a SAR number that the script tests, or a regression test that the script runs to verify that an error has been corrected.

---

**Note.** After you add the script to the repository, retrieve it from the repository, and view its properties, fields on the General tab display the repository date and the last person to open the script, as well as the time and date that the script was opened.

---

## Details Tab

On the Details tab, you can enter information that defines how the script fits into a test management scheme, as well as validation information and information about resetting data.

The following fields enable you to enter quality-assurance-related data that is useful in large-scale testing:

- Test Script.

You enter the collection of related scripts in which yours belongs, such as Tools Applications.

- Test Case.

You enter the specific function that you are testing, such as activating the Address by Effective Date feature. The values for this parameter are user defined.

Select one of the following options to indicate whether script validation occurs automatically or if you need to review the script output manually to determine if it ran successfully:

- Automatic
- Manual

Select the automatic validation option if simply running the script successfully means that the functions worked, and you do not need to further test the results. However, in some cases—such as when you test universal batch engines (UBEs)—you must manually review the output of the script to determine whether it was successful. For example, you might need to verify that a UBE report generated successfully.

If you select the Manual option, PeopleSoft AutoPilot enables the following additional options:

- Report
- Screen Prints
- Log Comments

Selecting one or more of these options reminds people who run the script to manually review the output after the script runs.

In addition, you can indicate whether the script resets changes that you made to constants or to master file data, such as additions to or deletions from the F4101 table.

## Comments Tab

On the Comments tab, you can enter additional descriptive information about the test, the purpose of the script, and any other information that is relevant.

## Categories Tab

The Categories tab enables you to specifically define the object of that you are testing. A system administrator creates user-defined testing categories, which appear in the form as options. For example, you might define a category of testing as package verification. Options on the Categories tabs—such as Daily Build or Weekly Package—can indicate the type of verification testing that the script performs.

## Naming Conventions for Saved Scripts

No predetermined rules exist for naming scripts that you add to the repository. Use a particular naming convention for all scripts. For example, you might give the script a title that identifies the application, release, or function that it tests. In addition, you might specify whether the test is new or a retest. Following a naming convention enables other users to identify the purpose of a script. Assigning properties to the script enables you to accurately subcategorize it; adhering to a naming convention furthers that goal.

### See Also

Chapter 7, “Working with the Script pane,” Understanding Script Retention, page 108

## Add to Repository Command

After you create a script, establish its properties, give it a unique title, and save it locally, you can add it to the repository using the Add Script to Repository form. You can use this form to change the properties of the script before you add it to the repository. However, you cannot specify the owner because PeopleSoft AutoPilot automatically assigns the owner ID, closes the script to prevent further changes, and adds it to the repository.

## Browse Repository Scripts Command

The Browse Repository Scripts command enables you to search for scripts in the repository. You can make the search as narrow or as broad as necessary. After you have found the script that you need, you can create a copy of it or check it out of the repository.

You use the following two forms with the Browse Repository Scripts command:

- Select Script
- Browse Scripts

You use the Select Script form to establish search criteria. The Browse Scripts form contains information about the scripts in the repository that have properties that match the criteria specified in the Select Script form.

### Select Script Form

The Select Script form contains the same tabs and fields as the Script Properties form. However, it is a query form rather than a form for entering script properties.

You can select any of the tabs, except Comments, and enter data in the fields or select options to establish search criteria for a particular type of script. Alternatively, you can enter the exact title of a script. PeopleSoft AutoPilot matches the criteria that you set and the entries in the Script Properties pages. If you do not enter any field information, PeopleSoft AutoPilot includes all scripts in the repository.

All of the fields on the General and Details tabs on the Select Script form—with the exception of the Reference Number field—enable you to use wildcards (asterisks) to have PeopleSoft AutoPilot include all scripts in the search.

You can use the asterisks alone or with an entry in a field. For example, to find all scripts that tested the 8.10 release, you might enter *\*810\** in the Title field. PeopleSoft AutoPilot includes in its search all scripts that contain 810 in the title, regardless of any text that comes before or after 810.

If you enter information in a field, PeopleSoft AutoPilot automatically appends the wildcard to the text string. For example, if you enter *P0911* in the Main Application field, PeopleSoft AutoPilot returns all scripts with a main application property that includes P0911, such as P0911A, P0911B, and so on.



The Reference Number field contains a 0, which indicates that you have not entered a SAR number. If you leave the 0 in this field, PeopleSoft AutoPilot does not use a reference number as a search criterion and includes all reference numbers, such as SARs, in its search.

If you enter a reference number, you limit your search to those scripts that tested a particular SAR or to tests that are defined by another reference number.

## Browse Scripts Form

When you click OK on the Select Script form, the Browse Scripts form appears. This form contains summaries of all scripts that match the data specified in the Select Script form. The Browse Scripts form contains the following column headings, which identify a script:

- Title
- Description
- Owner
- User
- Machine
- Security

The script owner is the person designated in the Script Properties form. The user is the last person who checked out or checked in the script. The Machine field identifies the workstation that the user used to check out the script. The Security field indicates the level of security that the owner attached to the script. The security levels and their meanings are as follows:

Security Level	Meaning
No restrictions	Anyone can check out and change the script; all properties can be changed except security level.
Owner Locked	Anyone can check out and change the script, but the owner and security level cannot be changed.
No Checkout/in	People who do not own the script can only create a copy of it and save changes locally.
No Access	People who do not own the script can only see that it resides in the repository.

The form also contains the following buttons:

- Get Copy, which enables you to create a copy of a script.
- Checkout, which enables you to check out a script from the repository.
- Undo Checkout, which enables you to undo a script checkout.
- Delete, which enables you to delete a script from the repository if you are authorized to do so.
- Close, which enables you to exit from the Browse Scripts form.

PeopleSoft AutoPilot disables the Get Copy and Checkout buttons until you select the title of a script. After you select a title, these two buttons are enabled.

You can use the Repository Script Properties form to review the properties of any script that you want to check out by right-clicking any script and selecting Properties. This form contains the four tabs that appear on the Select Properties and Script Properties forms. You cannot change script properties using the Repository Script Properties form.

You can create a copy of or check out a single script. You can also create multiple copies or check out more than one script. You can open the checked-out copy, check it in, or close the form. If you close the form, the script remains checked out. If you select a combination of scripts that you checked out and scripts that someone else checked out, you cannot open a script or check it in.

## Deletion of Scripts

You can delete from the repository any script that you own. However, the following restrictions exist:

- You cannot delete a script that you do not own unless you have authorization.
- You must enter a password that changes daily to delete a script that you did not add but that you are authorized to delete.
- You cannot delete a script that is included in another script.

## Get Copy Command

While you are in the Browse Scripts form, you can create a copy of a script that resides in the repository. PeopleSoft AutoPilot enables you to view and run a copy of a script that you obtain from the repository, but you cannot make any permanent changes to it without first saving it as a local copy in your script directory.

PeopleSoft AutoPilot enables the Get Copy button in the Browse Scripts form when you select a script that is not checked out. If you select a checked-out script, the Get Copy command is not available. Instead, you can use the Open command to access a copy of the checked-out script.

When you create a copy of a repositored script, the copy opens in PeopleSoft AutoPilot. The form caption bar contains the word Repository, and the title and description of the repositored script.

If you modify the copy and then click Save, PeopleSoft AutoPilot displays an error message indicating that the script is not checked out and that you must either check it out or save it as a local file before the changes take effect.

## Checkout Command

You might retrieve a script from the repository to modify it. To do so, you must check out the script, make and save the changes, and then check it back into the repository. PeopleSoft AutoPilot returns the script to the repository with the changes intact. The repository then contains a new version of the script.

When you check out a script from the repository, you check out the latest version of the script. A script cannot be checked out to more than one person at a time. This prevents two people from making changes to the script simultaneously.

You can also run the Checkout command when you are browsing scripts. In the Select Script form, you can limit the script search by selecting options and entering information on the various tabs.

## Undo Checkout Command

You can undo the Checkout command if, for example, you change an existing script but decide to cancel the changes. If you check in the script, the changes take effect, and the repository contains a new version of the script. If you use the Undo Checkout command, none of the changes that you make to the script take effect. To undo the checkout, select File, Repository, Check In/Check Out, Undo Check Out. A form confirms the undo checkout.

## My Checkouts Form

You can find scripts that you have checked out by using the My Checkouts form. The My Checkouts form enables you to keep track of the scripts that you check out to ensure that you check them back in. The My Checkouts form contains the same headings as the Browse Scripts form. Each document icon next to the script title displays a green check mark, which indicates that you have checked out the script on the machine that you are currently using.

You cannot check out a script on one machine and check it in on another. The My Checkouts form shows all of your checkouts on the current machine. If you check out a script on a machine other than the one on which you are currently working, the document icon next to the script title contains a red *X*, which indicates that you cannot check in the script from the current machine.

## Check In Command

After you check out a script, you can make any necessary changes to the repositied version. When you are finished with the changes, you must check in the script for them to take effect. When the script is checked in to the repository, the new version is available to other people. When you check in a script, it automatically closes and enters the repository, just as when you add a script to the repository.

## Where Included Command

You sometimes need to determine if a script is included with another script, because you cannot delete an included script. You can use the Where Included command to search the repository for all scripts that include another script.

When you enter the title of a script, the Where Included form displays the titles and descriptions of any repositied scripts that include the script title.

From the Where Included form, you can select a script and create a copy of it or check it out. The script that you select from this form is a master script; it is the parent of the script for which you initially searched, as well as for any other scripts that might be included with it. If no scripts appear in the Where Included form, you can delete the script.

---

## Working with the Script Repository

This section provides an overview of the script repository and discusses how to:

- Assign properties to a script.
- Add a script to the repository.
- Browse for repository scripts.
- Delete a script from the repository.

- Assign security to a repositied script.
- Get a copy of a script.
- Check out a script.
- Undo script checkout.
- Check in a script.
- Query for included scripts.
- Use a Command Line to Load a Repository Script.

## Understanding the Script Repository

The PeopleSoft AutoPilot script repository enables you to retrieve scripts for study, playback, and modification. You can add scripts that you create, and others can retrieve copies to review the functions that you tested or to use a script as a template for another script. In turn, you can retrieve other people's scripts for the same purpose. You can also check out scripts from the repository, change them, and store the new versions by checking them in.

The script repository works in conjunction with the other components of PeopleSoft AutoPilot. After you have written a script, you can assign properties to it and save it locally. When you add it to the repository, you make it available to others in a centralized storage location, and it becomes a controlled version that can be changed only by following prescribed procedures.

## Assigning Properties to a Script

Before you add a script to the repository, you can assign properties that remain with it when you save the script. Properties include the script title, description, main application tested, and other parameter values that you assign to it. These identifying features facilitate searches of the repository for scripts of a specified type. You save the property pages locally along with the script.

To assign properties to a script:

1. From the PeopleSoft AutoPilot form, open a script.
2. In the File menu, select the Properties option.  
The Script Properties form appears.
3. In the Script Properties form, enter information or select options in the fields on each of the following tabs:
  - General
  - Details
  - Comments
  - Categories
4. Click OK.
5. In the File menu, select Save or Save As and assign a title that follows your predetermined naming convention.

## Adding a Script to the Repository

After you save a script (and any assigned properties), you can add it to the repository. The script must be open before you can add it. When you select the Add to Repository command, PeopleSoft AutoPilot enables you to assign script properties for the first time or to modify previously assigned properties. After you add the script to the repository, PeopleSoft AutoPilot identifies the script by the title and description that you enter.

To add a script to the repository:

1. On the PeopleSoft AutoPilot form, open the script that you want to add to the repository.
2. In the File menu, select Repository, Add to Repository.

The Add Script to Repository form appears.

3. On the Add Script to Repository form, enter any script properties that you want to assign.
4. Click OK.

PeopleSoft AutoPilot closes the script and checks it into the repository. A controlled copy of the script now exists in the repository. You can still change the local copy.

## Browsing for Repository Scripts

You can browse the repository for scripts to run, use as a template, or modify. You can view all of the scripts in the repository or enter search criteria in the Select Script form to search for scripts of a certain type. You enter information and select options to establish search criteria. PeopleSoft AutoPilot uses the criteria to display any scripts that have matching properties.

To browse for scripts:

1. In the File menu of the PeopleSoft AutoPilot form, select Repository, and then Browse Repository Scripts.

The Select Script form appears. The form contains the same four tabs that appear on the Script Properties form and the Add Script to Repository form.

2. On the Select Script form, establish criteria for the scripts to retrieve.
3. Click OK.

The Browse Scripts form appears and displays the titles of the scripts that match the specified criteria.

## Deleting a Script from the Repository

You can delete a script from the repository if you are its owner or you have the proper authorization. As a precaution, PeopleSoft AutoPilot prompts you to first confirm the delete.

---

**Note.** You cannot delete a script if it is included in another script.

---

To delete a script from the repository:

1. In the File menu of the PeopleSoft AutoPilot form, select Repository, and then Browse Repository Scripts.
2. On the Select Script form, enter your user ID in the Owner field on the General tab and click OK.

---

**Note.** If you are authorized to delete scripts other than those that you own, you can use other selection criteria by completing additional controls and options on the tabs of the Select Script form.

---

3. On the Browse Scripts form, select the title of at least one script that you want to delete from the repository and click the Delete button.

An PeopleSoft AutoPilot form prompts you to confirm the deletion of the script. If you click Yes when the script is open, PeopleSoft AutoPilot closes the script.

4. Click OK.

---

**Note.** The deletion fails if the script is included in another script.

---

## Assigning Security to a Reposited Script

After you add a script to the repository, you can assign security to it, or you can leave it unsecured. Assigning security to a script restricts other people's to access it.

You can assign security to a script if you are not its original owner, but only if the original script has no restrictions. However, you must make yourself the owner of the script before you can change its security.

---

**Note.** You can also assign security to a script by using the My Checkouts form if you have checked out the script from the repository.

---

To assign security to a repositied script:

1. In the File menu of the PeopleSoft AutoPilot form, select Repository, and then Browse Repository Scripts.
2. On the Select Script form, enter your user ID in the Owner field on the General tab and click OK.
3. On the Browse Scripts form, select one or more scripts and right-click.  
The system displays a pop-up menu that contains four security-level options.
4. From the pop-up menu, select one of the following security levels:
  - No Restrictions
  - Owner Locked
  - No Checkout/in
  - No Access

## Creating a Copy of a Script

You can use the Browse Scripts form to create a copy of a script to play back or to use as a template for creating another script. Note these points about script copies:

- You can change the copy, but you can save the changes only to a separate local copy of the script, not to the repositied script.
- You cannot create copies of more than one script if one of the scripts that you select is checked out.
- If you select only the checked-out script, you create a copy of the script, including the changes made since the last checkout.

To create a copy of a script:

1. From the File menu of the PeopleSoft AutoPilot form, select Repository, and then select Browse Repository Scripts.
2. On the Select Script form, establish criteria for the types of scripts to retrieve.

3. Click OK.
4. In the Browse Scripts form, select at least one script.
5. Click the Get Copy button.

## Checking Out a Script

You can check out scripts that are added to the repository. Only one person at a time can check out a script. You can make changes to a script that you check out, and then check it back in. PeopleSoft AutoPilot saves all changes and creates a new version without prompting you to add the script to the repository.

If someone has checked out a script, the document icon next to the script title in the Browse Scripts form contains a check mark or an *X*. An *X* appears if a script has been checked out to another computer. If you attempt to select multiple scripts and one of them has already been checked out, PeopleSoft AutoPilot disables the Get Copy and Checkout commands. If you select the checked-out script only, you can open the script to check it into the repository or to undo the checkout.

To check out a script:

1. From the File menu of the PeopleSoft AutoPilot form, select Repository and Browse Repository Scripts.
2. On the Select Script form, establish criteria for the types of scripts to retrieve.
3. Click OK.
4. On Browse Scripts, select one or more scripts; and then click Check Out.

## Undoing Script Checkout

You can undo a script checkout if, for example, you make changes to the script but then decide not to save the changes.

To undo a script checkout:

1. From the File menu of the PeopleSoft AutoPilot form with a checked-out script open, select Repository, Check In/Check Out, Undo Check Out.  
An PeopleSoft AutoPilot form appears prompting you to confirm the checkout cancellation.
2. Click Yes.

## Checking in a Script

If you check out a script and make changes to it, you can check it back in to save the changes in the repository.

From the File menu of the PeopleSoft AutoPilot form, with a checked-out script open, select Repository, Check In/Check Out, Check In.

---

**Note.** You can identify all of the scripts that you have checked out by selecting File, Repository, My Checkouts.

---

The script closes. PeopleSoft AutoPilot checks in the new version of the script.

## Querying for Included Scripts

You can use the Where Included form to search the repository for any scripts in which a particular script is included. If a script is included in another repositored script, you cannot delete it from the repository.

To query for included scripts:

1. From the File menu of the PeopleSoft AutoPilot form, select Repository, Where Included.
2. On the Where Included form, enter the title of a script and click OK.

PeopleSoft AutoPilot displays the titles of all scripts that include the script for which you entered the title. You can check out or create a copy of these scripts.

## Using a Command Line to Load a Repository Script

You can load any repository script into PeopleSoft AutoPilot from a command line. The script command line parameter is passed in with an .ATR extension. This extension is used to designate repository scripts.

The command line calls the PeopleSoft AutoPilot executable file and identifies the specific repository script to load into PeopleSoft AutoPilot. The .ATR extension indicates to PeopleSoft AutoPilot that it must access the repository to retrieve and load the script that you entered in the command line. Enter the command as follows:

```
C:\PeopleSoft AutoPilot.exe MyRepositoryScript.ATR
```

where *MyRepositoryScript* is the title of the repository script to be loaded.

---

## Working with Script Reporting

This section provides an overview of script reporting and discusses how to:

- Use the event stream.
- Use the Test Results form.

## Understanding Script Reporting

The script repository contains information about scripts that test particular applications and processes. The architecture of PeopleSoft AutoPilot also contains a results repository, the F97214 table. This repository contains information about the actual PeopleSoft AutoPilot and PeopleSoft EnterpriseOne events that occur during script playback.

If you configure PeopleSoft AutoPilot to capture and store playback events, it records each event using internally placed code and code in PeopleSoft EnterpriseOne software. When playback finishes, PeopleSoft AutoPilot sends the record of events, called the *event stream*, to the repository. You can view each event stream on the Test Results form.

## Using the Event Stream

The event stream provides a snapshot of the events that occurred during script playback. For example, you can see which tables were opened, which business functions were called, which event rules were invoked, and the time required to complete each event. You can also identify error and warning messages that appeared. This information enables you to troubleshoot problems that occur during playback.

You also capture an event stream when you want to use the Analyzer Tool to analyze script playback, or use Virtual PeopleSoft AutoPilot to generate a virtual script that you can use to simulate multiple users on a single workstation.



## Using the Test Results Form

If you have configured script playback to capture, save, and display results, the Test Results form appears when playback finishes. The Test Results form displays playback data such as the event stream, which is a chronological listing of each event that occurs during playback. You can filter the list for test time, type, or text. You can also view previous test results, and you can review details about the results.

The Test Results form contains the following tabs:

- Browse Result Sets
- Summary
- JDE.INI
- JDE.LOG
- JDEBUG.LOG
- Screen Captures
- Messages
- Results

### Browse Result Sets Tab

The Browse Result Sets tab contains summaries of all the tests for which you have saved results. You can also view the events in an individual test. A check mark beside a test indicates that it was successful; an *X* indicates a test that failed or was cancelled.

The Test Results form for saved tests also permits you to print results and to export them to a spreadsheet using buttons at the bottom of the form. The Filter button enables you to filter the saved test results using the columns in the form as criteria. You use the Filter form to select a filter criterion.

### Summary Tab

Clicking the Summary tab displays the following properties for each test that you run:

- Script
- Machine
- Release
- Environment
- User
- Start time
- End time
- Elapsed playback time
- Status of the playback

### JDE.INI Tab

The JDE.INI tab enables you to view the initialized settings for PeopleSoft EnterpriseOne software that existed before PeopleSoft AutoPilot played the script. PeopleSoft AutoPilot captures the file from C:\Winnt\JDE.INI, and then displays its contents on the tab. You can troubleshoot the file to see, for example, whether paths in the JDE.INI setting point to the correct database or drive. You can also use data on the JDE.INI tab to duplicate the results of one test in another.

## jde.log Tab

After script playback, PeopleSoft AutoPilot captures the jde.log file from C:\jde.log and displays it on the jde.log tab of the Test Results form. You can view the contents of the file to track errors that occur during processing.

## Screen Captures Tab

If a script fails, PeopleSoft AutoPilot captures the screen that is active when the script fails and lists the screen shot on the Screen Captures tab.

## jddebug.log Tab

The jddebug.log tab displays the jddebug.log file that PeopleSoft AutoPilot captures from the C:\jddebug.log file after it completes script playback. You can troubleshoot the file to determine, for example, when normal execution of the script stopped. You can also review the timing of all processes that occur during script playback.

## Messages Tab

Data that appears on the Messages tab of the Test Results form summarizes the script that PeopleSoft AutoPilot played back.

You can review each context command and action command that you write in the script, as well as any error messages that PeopleSoft AutoPilot generates. In addition, you can review any error messages that PeopleSoft EnterpriseOne software generates during playback.

On the Messages tab, you can filter, print, and export test results. You can filter test results for the following:

- A particular point during playback that an event occurred
- A particular type of event, such as a message or an action in PeopleSoft AutoPilot
- A text description of the event in the Test Results form

You can print the test results, provided that you have set up a default printer to do so. Using the Export button, you can export test results to a spreadsheet.

## Results Tab

When a script finishes, you can see the results on the Messages tab in a grid format. The Results tab contains those same results in an enhanced tree control format that enables you to view individual events. You can use this tree control format to assist in troubleshooting. For example, if you are searching for errors in a script, you can identify them by the red exclamation marks next to the events.

## See Also

*Options for Configuring PeopleSoft AutoPilot in the PeopleSoft AutoPilot Guide*

*Results in the PeopleSoft AutoPilot Guide*

---

# Working with PeopleSoft AutoPilot Test Manager

This section provides an overview of Test Manager and discusses how to:

- Use the script display pane.

- Use the script storage pane.
- Use the test results pane.
- Use the Test Manager toolbar.

## Understanding Test Manager

PeopleSoft AutoPilot Test Manager enables you to test multiple PeopleSoft AutoPilot scripts in a batch, so that you can review results quickly and gather test results for archiving. You use Test Manager to create a playlist that contains scripts that you save on your local drive, scripts that you retrieve from the script repository, or a combination of both.

After you assemble a playlist, you run it. Test Manager launches PeopleSoft AutoPilot, which in turn launches EnterpriseOne software. Test Manager runs the playlist to completion, closing PeopleSoft AutoPilot each time a script completes and opening PeopleSoft AutoPilot immediately after the next script appears in the queue. Test Manager displays a test status of Failure, Success, or Incomplete for each script that runs.

You can view the results of each script playback. These results include messages that enable you to analyze the cause of a script failure. In addition, you can save the playlists that you create, edit them, and replay them.

## Using the Script Display Pane

The script display pane in Test Manager is the area where you select scripts to assemble a playlist. The pane contains two tabs, Local and Repository. When you select the Local tab, Test Manager displays all the scripts that you have stored on the local drive.

When you select the Repository tab and then click the Repository Filter button, Test Manager launches the Select Script form, which you can use to enter search criteria for scripts that have been checked in to the repository.

Test Manager populates the script display pane with the names of the scripts that match the property criteria that you enter in the Select Script form. You can add these scripts to the playlist in the script storage pane.

---

**Note.** Test Manager creates copies of the repository scripts. It does not check out scripts from the repository. Adding a repositored script to the Test Manager playlist does not prevent other users from checking out the script from the repository and changing it.

---

### See Also

[Chapter 10, “Storing Scripts and Test Results,” Property Pages for Scripts, page 143](#)

[Chapter 10, “Storing Scripts and Test Results,” Assigning Properties to a Script, page 150](#)

## Using the Script Storage Pane

You add scripts from the script display pane to the script storage pane to create a playlist. When you initially add scripts to the script storage pane, Test Manager displays the state of the script as Idle, which means that you have added it to the playlist but have not yet run it.

You can remove scripts from the script storage pane. When you remove a script from the playlist, Test Manager prompts you to confirm the action.

After you assemble the playlist, you run it by clicking the Run button on the toolbar. Test Manager launches PeopleSoft AutoPilot and runs the scripts in the order that they appear in the script storage pane. After the script runs, Test Manager displays one of the following states, depending on the results of the test: Success, Failure, Cancellation, or Incomplete.

## Using the Test Results Pane

After you have assembled and run a playlist, Test Manager summarizes the results in the test results pane. You can review the summary by clicking the Report button on the toolbar. The test result summary displays the following information about the test:

- Total number of tests generated.
- Status breakdown, including the number of scripts that failed, succeeded, were canceled, or did not finish.
- Name of the client machine.
- Environment in which the test was run.
- Release in which the test was run.
- Name of the script.
- Number of the test.
- Status of each script run.
- Time elapsed for each script run.
- Comments that you added to the script and designated for logging in Test Manager.

If a script fails, click the Report button to display message types in the test results pane, along with the time of the message. These message types provide information about why the script failed, as well as information about warning messages that might have appeared during playback.

In addition, Test Manager provides information about warning messages. The following table lists some of the message types that can appear in the test results pane and summarizes their meanings:

Message Type	Description
110	Failure status with text ###FAILURE&&&.
138	Warning status. Each message type 138 includes the path for a screen capture that is included in the PeopleSoft AutoPilot script.
2607	Failure status. No data returned.
2608	Failure status. Unexpected records found during validation.
2609	Failure status. Database validation failed.
3000	Failure status. Status bar message that contains warnings or error text. Message text that includes STB: Error, which indicates that the script failed. Warning messages do not indicate that the script failed. However, to help the tester, Test Manager summarizes all warning messages.

Message Type	Description
6016	Failure status. Variable not found.
6301	Warning status. PeopleSoft AutoPilot failed to set processing option text, which might cause a failure later in the script.

## Using the PeopleSoft AutoPilot Test Manager Toolbar

The Test Manager toolbar enables you to control a test session and view its results. The Test Manager toolbar contains the following buttons:

Close	Close a test session. If you have not saved the playlist, you are prompted to do so.
Stop	Stop a script playback session.
Reset	Reset script status and test results.
Run	Initiate a test session, which launches PeopleSoft AutoPilot.
Log	Display the Test Results form, which contains detailed summaries of each test that you ran and saved in PeopleSoft AutoPilot.
Report	Populate the test results pane with summary information about the playback.
Remove	Remove a script from the playlist in the script storage pane.
Add	Add a script from the script display pane to the script storage pane.
Up	Move a script up in the playlist.
Down	Move a script down in the playlist.

---

## Managing Script Testing

This section provides an overview of script testing and discusses how to:

- Create a playlist.
- Save a playlist.
- Run a test.
- View test results.
- Reset a test.

## Understanding Script Testing

You use PeopleSoft AutoPilot Test Manager to create a playlist from scripts that reside on your local drive or in the script repository. Test Manager enables you to run a playlist multiple times, without intervention. The playlist can contain a combination of local and repositored scripts. You can save a playlist, or you can reset it and play it again from the top. You can view the summarized results of each playback in the test results pane, collected playback results, or the events of an individual test.

## Creating a Playlist

The first task in Test Manager is to create a playlist. You retrieve the scripts for your playlist from your local drive, the script repository, or both. The Add button on the toolbar enables you to move scripts from the script display pane to the script storage pane, where the playlist resides.

To create a playlist:

1. On the desktop or in the directory where you store Test Manager, click the Test Manager executable file. The Test Manager splash screen appears, followed by the PeopleSoft AutoPilot Test Manager form.

---

**Note.** When the Test Manager form appears, the script display pane and the script storage pane might not appear. Drag the pane that contains them by using the grabber, which is represented by a pair of vertical bars.

---

2. In the script display pane of the PeopleSoft AutoPilot Test Manager form, select either the Local or the Repository tab.
3. If you select the Local tab, select a local script from the script display pane and click the Add button on the toolbar.

Test Manager adds the selected test to the script storage pane.

---

**Note.** You can select more than one test by selecting a script in the script display pane, holding down either the CTRL or the SHIFT key, and selecting additional scripts.

---

4. Select the Repository tab.
5. Click the Repository Filter button.

Test Manager displays the Select Script form.

6. In the Select Script form, select script criteria to narrow the number of scripts that you copy from the repository, and then click OK.

Test Manager displays in the script display pane any repository scripts that match the search criteria.

7. Select one or more repository scripts from the script display pane, and then click the Add button on the toolbar.
8. Continue adding local and repository scripts until you have created the playlist.
9. To change the sequence of the scripts, click a script in the script storage pane and click the Up or Down button on the toolbar.
10. To remove a script from the playlist, select it in the script storage pane and click the Remove button on the toolbar.

## Saving a Playlist

After you create a playlist, you can save it, or you can run the test before you save it. Remember, if you do not save the playlist, Test Manager prompts you to do so when you exit the form.

To save a playlist:

1. In the File menu of the PeopleSoft AutoPilot Test Manager form, select Save or Save As.
2. Assign the playlist a filename and save it to a drive and directory, and click Save.

---

**Note.** Test Manager assigns to all playlists the default extension of .apl.

---

## Running a Test

After you create a playlist, you can run the test. Test Manager launches PeopleSoft AutoPilot, and then runs each script in the queue in the order that you set up in the script storage pane.

Test Manager launches PeopleSoft AutoPilot, minimizes the PeopleSoft AutoPilot form, and then begins running the first script in the queue. As each test finishes, Test Manager displays its result in the script storage pane. When Test Manager finishes running a script, it closes PeopleSoft AutoPilot, and then relaunched it with the beginning of the next script in the queue.

As each script finishes running, Test Manager displays the result of Success, Failure, Canceled, or Incomplete.

To run a test:

1. In the File menu of the PeopleSoft AutoPilot Test Manager form, select Open.
2. Open the drive, directory, and file in which you store your playlists, select one or more playlists, and click Open.
3. On the toolbar of the PeopleSoft AutoPilot Test Manager form, click Run.

---

**Note.** During playback, EnterpriseOne software remains open, unless a script contains an Exit PeopleSoft command, in which case the software closes. In this case, with the beginning of the next script in the queue, PeopleSoft AutoPilot Test Manager launches PeopleSoft AutoPilot, which launches EnterpriseOne software.

---

4. To stop the test, click the Stop button on the toolbar.

## Viewing Test Results

After Test Manager completes the playlist, you can review the playback results in one of two ways. Click the Report button to review in the test results pane a summary of the results from the current playlist. To view summaries of the tests from all the playlists that you run and save, click the Log button.

To view test results:

1. On the PeopleSoft AutoPilot Test Manager form, open a saved playlist.
2. After PeopleSoft AutoPilot Test Manager completes the playlist, click the Report button on the toolbar.  
Test Manager populates the test results pane with a summary of the results for each script in the playlist.
3. To view summaries of all scripts that you have played back, click the Log button on the toolbar.  
Test Manager displays the Test Results form, which contains summary information about the results of all played-back scripts.

4. To view in detail all the events for the playback of an individual script, select the script on the Test Results form, and then select the Results tab or the Messages tab.

## Resetting a Test

After you assemble a playlist and run it, you can reset the test, which overwrites the previous results. Resetting might be appropriate if scripts in the original test fails and you make changes to correct the failure.

To reset a test:

1. On the PeopleSoft AutoPilot Test Manager form, open a playlist that you have already run.
2. On the toolbar, click the Reset button.  
If Test Manager displays a form warning you that resetting the test overwrites the existing results, click Yes.
3. On the toolbar, click the Run button to rerun the test



# Glossary of PeopleSoft Terms

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

	PeopleSoft Stock Administration, and the COBRA Administration feature of the Base Benefits business process.
<b>action template</b>	In PeopleSoft Receivables, outlines a set of escalating actions that the system or user performs based on the period of time that a customer or item has been in an action plan for a specific condition.
<b>activity</b>	<p>In PeopleSoft Enterprise Learning Management, an instance of a catalog item (sometimes called a class) that is available for enrollment. The activity defines such things as the costs that are associated with the offering, enrollment limits and deadlines, and waitlisting capacities.</p> <p>In PeopleSoft Enterprise Performance Management, the work of an organization and the aggregation of actions that are used for activity-based costing.</p> <p>In PeopleSoft Project Costing, the unit of work that provides a further breakdown of projects—usually into specific tasks.</p> <p>In PeopleSoft Workflow, a specific transaction that you might need to perform in a business process. Because it consists of the steps that are used to perform a transaction, it is also known as a step map.</p>
<b>address usage</b>	In PeopleSoft Enterprise Campus Solutions, a grouping of address types defining the order in which the address types are used. For example, you might define an address usage code to process addresses in the following order: billing address, dormitory address, home address, and then work address.
<b>adjustment calendar</b>	In PeopleSoft Enterprise Campus Solutions, the adjustment calendar controls how a particular charge is adjusted on a student's account when the student drops classes or withdraws from a term. The charge adjustment is based on how much time has elapsed from a predetermined date, and it is determined as a percentage of the original charge amount.
<b>administrative function</b>	In PeopleSoft Enterprise Campus Solutions, a particular functional area that processes checklists, communication, and comments. The administrative function identifies which variable data is added to a person's checklist or communication record when a specific checklist code, communication category, or comment is assigned to the student. This key data enables you to trace that checklist, communication, or comment back to a specific processing event in a functional area.
<b>admit type</b>	In PeopleSoft Enterprise Campus Solutions, a designation used to distinguish first-year applications from transfer applications.
<b>agreement</b>	In PeopleSoft eSettlements, provides a way to group and specify processing options, such as payment terms, pay from a bank, and notifications by a buyer and supplier location combination.
<b>allocation rule</b>	In PeopleSoft Enterprise Incentive Management, an expression within compensation plans that enables the system to assign transactions to nodes and participants. During transaction allocation, the allocation engine traverses the compensation structure from the current node to the root node, checking each node for plans that contain allocation rules.
<b>alternate account</b>	A feature in PeopleSoft General Ledger that enables you to create a statutory chart of accounts and enter statutory account transactions at the detail transaction level, as required for recording and reporting by some national governments.
<b>analysis database</b>	In PeopleSoft Enterprise Campus Solutions, database tables that store large amounts of student information that may not appear in standard report formats. The analysis database tables contain keys for all objects in a report that an application program can use to reference other student-record objects that are not contained in the printed report. For instance, the analysis database contains data on courses that are considered for satisfying a requirement but that are rejected. It also contains information on

	courses captured by global limits. An analysis database is used in PeopleSoft Enterprise Academic Advisement.
<b>AR specialist</b>	Abbreviation for <i>receivables specialist</i> . In PeopleSoft Receivables, an individual in who tracks and resolves deductions and disputed items.
<b>arbitration plan</b>	In PeopleSoft Enterprise Pricer, defines how price rules are to be applied to the base price when the transaction is priced.
<b>assessment rule</b>	In PeopleSoft Receivables, a user-defined rule that the system uses to evaluate the condition of a customer's account or of individual items to determine whether to generate a follow-up action.
<b>asset class</b>	An asset group used for reporting purposes. It can be used in conjunction with the asset category to refine asset classification.
<b>attribute/value pair</b>	In PeopleSoft Directory Interface, relates the data that makes up an entry in the directory information tree.
<b>audience</b>	In PeopleSoft Enterprise Campus Solutions, a segment of the database that relates to an initiative, or a membership organization that is based on constituent attributes rather than a dues-paying structure. Examples of audiences include the Class of '65 and Undergraduate Arts & Sciences.
<b>authentication server</b>	A server that is set up to verify users of the system.
<b>base time period</b>	In PeopleSoft Business Planning, the lowest level time period in a calendar.
<b>benchmark job</b>	In PeopleSoft Workforce Analytics, a benchmark job is a job code for which there is corresponding salary survey data from published, third-party sources.
<b>billing career</b>	In PeopleSoft Enterprise Campus Solutions, the one career under which other careers are grouped for billing purposes if a student is active simultaneously in multiple careers.
<b>bio bit or bio brief</b>	In PeopleSoft Enterprise Campus Solutions, a report that summarizes information stored in the system about a particular constituent. You can generate standard or specialized reports.
<b>book</b>	In PeopleSoft Asset Management, used for storing financial and tax information, such as costs, depreciation attributes, and retirement information on assets.
<b>branch</b>	A tree node that rolls up to nodes above it in the hierarchy, as defined in PeopleSoft Tree Manager.
<b>budgetary account only</b>	An account used by the system only and not by users; this type of account does not accept transactions. You can only budget with this account. Formerly called "system-maintained account."
<b>budget check</b>	In commitment control, the processing of source transactions against control budget ledgers, to see if they pass, fail, or pass with a warning.
<b>budget control</b>	In commitment control, budget control ensures that commitments and expenditures don't exceed budgets. It enables you to track transactions against corresponding budgets and terminate a document's cycle if the defined budget conditions are not met. For example, you can prevent a purchase order from being dispatched to a vendor if there are insufficient funds in the related budget to support it.
<b>budget period</b>	The interval of time (such as 12 months or 4 quarters) into which a period is divided for budgetary and reporting purposes. The ChartField allows maximum flexibility to define operational accounting time periods without restriction to only one calendar.

<b>business event</b>	<p>In PeopleSoft Receivables, defines the processing characteristics for the Receivable Update process for a draft activity.</p> <p>In PeopleSoft Sales Incentive Management, an original business transaction or activity that may justify the creation of a PeopleSoft Enterprise Incentive Management event (a sale, for example).</p>
<b>business unit</b>	A corporation or a subset of a corporation that is independent with regard to one or more operational or accounting functions.
<b>buyer</b>	In PeopleSoft eSettlements, an organization (or business unit, as opposed to an individual) that transacts with suppliers (vendors) within the system. A buyer creates payments for purchases that are made in the system.
<b>campus</b>	In PeopleSoft Enterprise Campus Solutions, an entity that is usually associated with a distinct physical administrative unit, that belongs to a single academic institution, that uses a unique course catalog, and that produces a common transcript for students within the same academic career.
<b>catalog item</b>	In PeopleSoft Enterprise Learning Management, a specific topic that a learner can study and have tracked. For example, "Introduction to Microsoft Word." A catalog item contains general information about the topic and includes a course code, description, categorization, keywords, and delivery methods. A catalog item can have one or more learning activities.
<b>catalog map</b>	In PeopleSoft Catalog Management, translates values from the catalog source data to the format of the company's catalog.
<b>catalog partner</b>	In PeopleSoft Catalog Management, shares responsibility with the enterprise catalog manager for maintaining catalog content.
<b>categorization</b>	Associates partner offerings with catalog offerings and groups them into enterprise catalog categories.
<b>category</b>	In PeopleSoft Enterprise Campus Solutions, a broad grouping to which specific comments or communications (contexts) are assigned. Category codes are also linked to 3C access groups so that you can assign data-entry or view-only privileges across functions.
<b>channel</b>	In PeopleSoft MultiChannel Framework, email, chat, voice (computer telephone integration [CTI]), or a generic event.
<b>ChartField</b>	A field that stores a chart of accounts, resources, and so on, depending on the PeopleSoft application. ChartField values represent individual account numbers, department codes, and so forth.
<b>ChartField balancing</b>	You can require specific ChartFields to match up (balance) on the debit and the credit side of a transaction.
<b>ChartField combination edit</b>	The process of editing journal lines for valid ChartField combinations based on user-defined rules.
<b>ChartKey</b>	One or more fields that uniquely identify each row in a table. Some tables contain only one field as the key, while others require a combination.
<b>checkbook</b>	In PeopleSoft Promotions Management, enables you to view financial data (such as planned, incurred, and actual amounts) that is related to funds and trade promotions.
<b>checklist code</b>	In PeopleSoft Enterprise Campus Solutions, a code that represents a list of planned or completed action items that can be assigned to a staff member, volunteer, or unit. Checklists enable you to view all action assignments on one page.

<b>class</b>	In PeopleSoft Enterprise Campus Solutions, a specific offering of a course component within an academic term.  See also <i>course</i> .
<b>Class ChartField</b>	A ChartField value that identifies a unique appropriation budget key when you combine it with a fund, department ID, and program code, as well as a budget period. Formerly called <i>sub-classification</i> .
<b>clearance</b>	In PeopleSoft Enterprise Campus Solutions, the period of time during which a constituent in PeopleSoft Contributor Relations is approved for involvement in an initiative or an action. Clearances are used to prevent development officers from making multiple requests to a constituent during the same time period.
<b>clone</b>	In PeopleCode, to make a unique copy. In contrast, to <i>copy</i> may mean making a new reference to an object, so if the underlying object is changed, both the copy and the original change.
<b>cohort</b>	In PeopleSoft Enterprise Campus Solutions, the highest level of the three-level classification structure that you define for enrollment management. You can define a cohort level, link it to other levels, and set enrollment target numbers for it.  See also <i>population</i> and <i>division</i> .
<b>collection</b>	To make a set of documents available for searching in Verity, you must first create at least one collection. A collection is set of directories and files that allow search application users to use the Verity search engine to quickly find and display source documents that match search criteria. A collection is a set of statistics and pointers to the source documents, stored in a proprietary format on a file server. Because a collection can only store information for a single location, PeopleSoft maintains a set of collections (one per language code) for each search index object.
<b>collection rule</b>	In PeopleSoft Receivables, a user-defined rule that defines actions to take for a customer based on both the amount and the number of days past due for outstanding balances.
<b>comm key</b>	See <i>communication key</i> .
<b>communication key</b>	In PeopleSoft Enterprise Campus Solutions, a single code for entering a combination of communication category, communication context, communication method, communication direction, and standard letter code. Communication keys (also called <i>comm keys</i> or <i>speed keys</i> ) can be created for background processes as well as for specific users.
<b>compensation object</b>	In PeopleSoft Enterprise Incentive Management, a node within a compensation structure. Compensation objects are the building blocks that make up a compensation structure's hierarchical representation.
<b>compensation structure</b>	In PeopleSoft Enterprise Incentive Management, a hierarchical relationship of compensation objects that represents the compensation-related relationship between the objects.
<b>condition</b>	In PeopleSoft Receivables, occurs when there is a change of status for a customer's account, such as reaching a credit limit or exceeding a user-defined balance due.
<b>configuration parameter catalog</b>	Used to configure an external system with PeopleSoft. For example, a configuration parameter catalog might set up configuration and communication parameters for an external server.
<b>configuration plan</b>	In PeopleSoft Enterprise Incentive Management, configuration plans hold allocation information for common variables (not incentive rules) and are attached to a node without a participant. Configuration plans are not processed by transactions.

<b>constituents</b>	In PeopleSoft Enterprise Campus Solutions, friends, alumni, organizations, foundations, or other entities affiliated with the institution, and about which the institution maintains information. The constituent types delivered with PeopleSoft Enterprise Contributor Relations Solutions are based on those defined by the Council for the Advancement and Support of Education (CASE).
<b>content reference</b>	Content references are pointers to content registered in the portal registry. These are typically either URLs or iScripts. Content references fall into three categories: target content, templates, and template pagelets.
<b>context</b>	<p>In PeopleCode, determines which buffer fields can be contextually referenced and which is the current row of data on each scroll level when a PeopleCode program is running.</p> <p>In PeopleSoft Enterprise Campus Solutions, a specific instance of a comment or communication. One or more contexts are assigned to a category, which you link to 3C access groups so that you can assign data-entry or view-only privileges across functions.</p> <p>In PeopleSoft Enterprise Incentive Management, a mechanism that is used to determine the scope of a processing run. PeopleSoft Enterprise Incentive Management uses three types of context: plan, period, and run-level.</p>
<b>control table</b>	Stores information that controls the processing of an application. This type of processing might be consistent throughout an organization, or it might be used only by portions of the organization for more limited sharing of data.
<b>cost profile</b>	A combination of a receipt cost method, a cost flow, and a deplete cost method. A profile is associated with a cost book and determines how items in that book are valued, as well as how the material movement of the item is valued for the book.
<b>cost row</b>	A cost transaction and amount for a set of ChartFields.
<b>course</b>	<p>In PeopleSoft Enterprise Campus Solutions, a course that is offered by a school and that is typically described in a course catalog. A course has a standard syllabus and credit level; however, these may be modified at the class level. Courses can contain multiple components such as lecture, discussion, and lab.</p> <p>See also <i>class</i>.</p>
<b>course share set</b>	In PeopleSoft Enterprise Campus Solutions, a tag that defines a set of requirement groups that can share courses. Course share sets are used in PeopleSoft Enterprise Academic Advisement.
<b>current learning</b>	In PeopleSoft Enterprise Learning Management, a self-service repository for all of a learner's in-progress learning activities and programs.
<b>data acquisition</b>	In PeopleSoft Enterprise Incentive Management, the process during which raw business transactions are acquired from external source systems and fed into the operational data store (ODS).
<b>data elements</b>	<p>Data elements, at their simplest level, define a subset of data and the rules by which to group them.</p> <p>For Workforce Analytics, data elements are rules that tell the system what measures to retrieve about your workforce groups.</p>
<b>dataset</b>	A data grouping that enables role-based filtering and distribution of data. You can limit the range and quantity of data that is displayed for a user by associating dataset rules with user roles. The result of dataset rules is a set of data that is appropriate for the user's roles.
<b>delivery method</b>	In PeopleSoft Enterprise Learning Management, identifies the primary type of delivery method in which a particular learning activity is offered. Also provides

	<p>default values for the learning activity, such as cost and language. This is primarily used to help learners search the catalog for the type of delivery from which they learn best. Because PeopleSoft Enterprise Learning Management is a blended learning system, it does not enforce the delivery method.</p> <p>In PeopleSoft Supply Chain Management, identifies the method by which goods are shipped to their destinations (such as truck, air, rail, and so on). The delivery method is specified when creating shipment schedules.</p>
<b>delivery method type</b>	In PeopleSoft Enterprise Learning Management, identifies how learning activities can be delivered—for example, through online learning, classroom instruction, seminars, books, and so forth—in an organization. The type determines whether the delivery method includes scheduled components.
<b>directory information tree</b>	In PeopleSoft Directory Interface, the representation of a directory's hierarchical structure.
<b>division</b>	<p>In PeopleSoft Enterprise Campus Solutions, the lowest level of the three-level classification structure that you define in PeopleSoft Enterprise Recruiting and Admissions for enrollment management. You can define a division level, link it to other levels, and set enrollment target numbers for it.</p> <p>See also <i>population</i> and <i>cohort</i>.</p>
<b>document sequencing</b>	A flexible method that sequentially numbers the financial transactions (for example, bills, purchase orders, invoices, and payments) in the system for statutory reporting and for tracking commercial transaction activity.
<b>dynamic detail tree</b>	A tree that takes its detail values—dynamic details—directly from a table in the database, rather than from a range of values that are entered by the user.
<b>edit table</b>	A table in the database that has its own record definition, such as the Department table. As fields are entered into a PeopleSoft application, they can be validated against an edit table to ensure data integrity throughout the system.
<b>effective date</b>	A method of dating information in PeopleSoft applications. You can predate information to add historical data to your system, or postdate information in order to enter it before it actually goes into effect. By using effective dates, you don't delete values; you enter a new value with a current effective date.
<b>EIM ledger</b>	Abbreviation for <i>Enterprise Incentive Management ledger</i> . In PeopleSoft Enterprise Incentive Management, an object to handle incremental result gathering within the scope of a participant. The ledger captures a result set with all of the appropriate traces to the data origin and to the processing steps of which it is a result.
<b>elimination set</b>	In PeopleSoft General Ledger, a related group of intercompany accounts that is processed during consolidations.
<b>entry event</b>	In PeopleSoft General Ledger, Receivables, Payables, Purchasing, and Billing, a business process that generates multiple debits and credits resulting from single transactions to produce standard, supplemental accounting entries.
<b>equitization</b>	In PeopleSoft General Ledger, a business process that enables parent companies to calculate the net income of subsidiaries on a monthly basis and adjust that amount to increase the investment amount and equity income amount before performing consolidations.
<b>equity item limit</b>	In PeopleSoft Enterprise Campus Solutions, the amounts of funds set by the institution to be awarded with discretionary or gift funds. The limit could be reduced by amounts equal to such things as expected family contribution (EFC) or parent contribution. Students are packaged by Equity Item Type Groups and Related Equity Item Types. This limit can be used to assure that similar student populations are packaged equally.

<b>event</b>	<p>A predefined point either in the Component Processor flow or in the program flow. As each point is encountered, the event activates each component, triggering any PeopleCode program that is associated with that component and that event. Examples of events are FieldChange, SavePreChange, and RowDelete.</p> <p>In PeopleSoft Human Resources, also refers to an incident that affects benefits eligibility.</p>
<b>event propagation process</b>	<p>In PeopleSoft Sales Incentive Management, a process that determines, through logic, the propagation of an original PeopleSoft Enterprise Incentive Management event and creates a derivative (duplicate) of the original event to be processed by other objects. Sales Incentive Management uses this mechanism to implement splits, roll-ups, and so on. Event propagation determines who receives the credit.</p>
<b>exception</b>	<p>In PeopleSoft Receivables, an item that either is a deduction or is in dispute.</p>
<b>exclusive pricing</b>	<p>In PeopleSoft Order Management, a type of arbitration plan that is associated with a price rule. Exclusive pricing is used to price sales order transactions.</p>
<b>fact</b>	<p>In PeopleSoft applications, facts are numeric data values from fields from a source database as well as an analytic application. A fact can be anything you want to measure your business by, for example, revenue, actual, budget data, or sales numbers. A fact is stored on a fact table.</p>
<b>financial aid term</b>	<p>In PeopleSoft Enterprise Campus Solutions, a combination of a period of time that the school determines as an instructional accounting period and an academic career. It is created and defined during the setup process. Only terms eligible for financial aid are set up for each financial aid career.</p>
<b>forecast item</b>	<p>A logical entity with a unique set of descriptive demand and forecast data that is used as the basis to forecast demand. You create forecast items for a wide range of uses, but they ultimately represent things that you buy, sell, or use in your organization and for which you require a predictable usage.</p>
<b>fund</b>	<p>In PeopleSoft Promotions Management, a budget that can be used to fund promotional activity. There are four funding methods: top down, fixed accrual, rolling accrual, and zero-based accrual.</p>
<b>gap</b>	<p>In PeopleSoft Enterprise Campus Solutions, an artificial figure that sets aside an amount of unmet financial aid need that is not funded with Title IV funds. A gap can be used to prevent fully funding any student to conserve funds, or it can be used to preserve unmet financial aid need so that institutional funds can be awarded.</p>
<b>generic process type</b>	<p>In PeopleSoft Process Scheduler, process types are identified by a generic process type. For example, the generic process type SQR includes all SQR process types, such as SQR process and SQR report.</p>
<b>gift table</b>	<p>In PeopleSoft Enterprise Campus Solutions, a table or so-called <i>donor pyramid</i> describing the number and size of gifts that you expect will be needed to successfully complete the campaign in PeopleSoft Contributor Relations. The gift table enables you to estimate the number of donors and prospects that you need at each gift level to reach the campaign goal.</p>
<b>GL business unit</b>	<p>Abbreviation for <i>general ledger business unit</i>. A unit in an organization that is an independent entity for accounting purposes. It maintains its own set of accounting books.</p> <p>See also <i>business unit</i>.</p>
<b>GL entry template</b>	<p>Abbreviation for <i>general ledger entry template</i>. In PeopleSoft Enterprise Campus Solutions, a template that defines how a particular item is sent to the general ledger. An item-type maps to the general ledger, and the GL entry template can involve multiple general ledger accounts. The entry to the general ledger is further controlled</p>



by high-level flags that control the summarization and the type of accounting—that is, accrual or cash.

<b>GL Interface process</b>	Abbreviation for <i>General Ledger Interface process</i> . In PeopleSoft Enterprise Campus Solutions, a process that is used to send transactions from PeopleSoft Enterprise Student Financials to the general ledger. Item types are mapped to specific general ledger accounts, enabling transactions to move to the general ledger when the GL Interface process is run.
<b>group</b>	<p>In PeopleSoft Billing and Receivables, a posting entity that comprises one or more transactions (items, deposits, payments, transfers, matches, or write-offs).</p> <p>In PeopleSoft Human Resources Management and Supply Chain Management, any set of records that are associated under a single name or variable to run calculations in PeopleSoft business processes. In PeopleSoft Time and Labor, for example, employees are placed in groups for time reporting purposes.</p>
<b>incentive object</b>	In PeopleSoft Enterprise Incentive Management, the incentive-related objects that define and support the PeopleSoft Enterprise Incentive Management calculation process and results, such as plan templates, plans, results data, user interaction objects, and so on.
<b>incentive rule</b>	In PeopleSoft Sales Incentive Management, the commands that act on transactions and turn them into compensation. A rule is one part in the process of turning a transaction into compensation.
<b>incur</b>	In PeopleSoft Promotions Management, to become liable for a promotional payment. In other words, you owe that amount to a customer for promotional activities.
<b>initiative</b>	In PeopleSoft Enterprise Campus Solutions, the basis from which all advancement plans are executed. It is an organized effort targeting a specific constituency, and it can occur over a specified period of time with specific purposes and goals. An initiative can be a campaign, an event, an organized volunteer effort, a membership drive, or any other type of effort defined by the institution. Initiatives can be multipart, and they can be related to other initiatives. This enables you to track individual parts of an initiative, as well as entire initiatives.
<b>inquiry access</b>	<p>In PeopleSoft Enterprise Campus Solutions, a type of security access that permits the user only to view data.</p> <p>See also <i>update access</i>.</p>
<b>institution</b>	In PeopleSoft Enterprise Campus Solutions, an entity (such as a university or college) that is independent of other similar entities and that has its own set of rules and business processes.
<b>item</b>	<p>In PeopleSoft Inventory, a tangible commodity that is stored in a business unit (shipped from a warehouse).</p> <p>In PeopleSoft Demand Planning, Inventory Policy Planning, and Supply Planning, a noninventory item that is designated as being used for planning purposes only. It can represent a family or group of inventory items. It can have a planning bill of material (BOM) or planning routing, and it can exist as a component on a planning BOM. A planning item cannot be specified on a production or engineering BOM or routing, and it cannot be used as a component in a production. The quantity on hand will never be maintained.</p> <p>In PeopleSoft Receivables, an individual receivable. An item can be an invoice, a credit memo, a debit memo, a write-off, or an adjustment.</p>
<b>item shuffle</b>	In PeopleSoft Enterprise Campus Solutions, a process that enables you to change a payment allocation without having to reverse the payment.

<b>joint communication</b>	In PeopleSoft Enterprise Campus Solutions, one letter that is addressed jointly to two people. For example, a letter might be addressed to both Mr. Sudhir Awat and Ms. Samantha Mortelli. A relationship must be established between the two individuals in the database, and at least one of the individuals must have an ID in the database.
<b>keyword</b>	In PeopleSoft Enterprise Campus Solutions, a term that you link to particular elements within PeopleSoft Student Financials, Financial Aid, and Contributor Relations. You can use keywords as search criteria that enable you to locate specific records in a search dialog box.
<b>KPI</b>	An abbreviation for <i>key performance indicator</i> . A high-level measurement of how well an organization is doing in achieving critical success factors. This defines the data value or calculation upon which an assessment is determined.
<b>LDIF file</b>	Abbreviation for <i>Lightweight Directory Access Protocol (LDAP) Data Interchange Format file</i> . Contains discrepancies between PeopleSoft data and directory data.
<b>learner group</b>	In PeopleSoft Enterprise Learning Management, a group of learners who are linked to the same learning environment. Members of the learner group can share the same attributes, such as the same department or job code. Learner groups are used to control access to and enrollment in learning activities and programs. They are also used to perform group enrollments and mass enrollments in the back office.
<b>learning components</b>	In PeopleSoft Enterprise Learning Management, the foundational building blocks of learning activities. PeopleSoft Enterprise Learning Management supports six basic types of learning components: web-based, session, webcast, test, survey, and assignment. One or more of these learning component types compose a single learning activity.
<b>learning environment</b>	In PeopleSoft Enterprise Learning Management, identifies a set of categories and catalog items that can be made available to learner groups. Also defines the default values that are assigned to the learning activities and programs that are created within a particular learning environment. Learning environments provide a way to partition the catalog so that learners see only those items that are relevant to them.
<b>learning history</b>	In PeopleSoft Enterprise Learning Management, a self-service repository for all of a learner's completed learning activities and programs.
<b>ledger mapping</b>	You use ledger mapping to relate expense data from general ledger accounts to resource objects. Multiple ledger line items can be mapped to one or more resource IDs. You can also use ledger mapping to map dollar amounts (referred to as <i>rates</i> ) to business units. You can map the amounts in two different ways: an actual amount that represents actual costs of the accounting period, or a budgeted amount that can be used to calculate the capacity rates as well as budgeted model results. In PeopleSoft Enterprise Warehouse, you can map general ledger accounts to the EW Ledger table.
<b>library section</b>	In PeopleSoft Enterprise Incentive Management, a section that is defined in a plan (or template) and that is available for other plans to share. Changes to a library section are reflected in all plans that use it.
<b>linked section</b>	In PeopleSoft Enterprise Incentive Management, a section that is defined in a plan template but appears in a plan. Changes to linked sections propagate to plans using that section.
<b>linked variable</b>	In PeopleSoft Enterprise Incentive Management, a variable that is defined and maintained in a plan template and that also appears in a plan. Changes to linked variables propagate to plans using that variable.
<b>LMS</b>	Abbreviation for <i>learning management system</i> . In PeopleSoft Enterprise Campus Solutions, LMS is a PeopleSoft Student Records feature that provides a common set of interoperability standards that enable the sharing of instructional content and data between learning and administrative environments.

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

<b>national allowance</b>	In PeopleSoft Promotions Management, a promotion at the corporate level that is funded by nondiscretionary dollars. In the industry, you may know this as a national promotion, a corporate promotion, or a corporate discount.
<b>need</b>	In PeopleSoft Enterprise Campus Solutions, the difference between the cost of attendance (COA) and the expected family contribution (EFC). It is the gap between the cost of attending the school and the student's resources. The financial aid package is based on the amount of financial need. The process of determining a student's need is called <i>need analysis</i> .
<b>node-oriented tree</b>	A tree that is based on a detail structure, but the detail values are not used.
<b>pagelet</b>	Each block of content on the home page is called a pagelet. These pagelets display summary information within a small rectangular area on the page. The pagelet provide users with a snapshot of their most relevant PeopleSoft and non-PeopleSoft content.
<b>participant</b>	In PeopleSoft Enterprise Incentive Management, participants are recipients of the incentive compensation calculation process.
<b>participant object</b>	Each participant object may be related to one or more compensation objects. See also <i>compensation object</i> .
<b>partner</b>	A company that supplies products or services that are resold or purchased by the enterprise.
<b>pay cycle</b>	In PeopleSoft Payables, a set of rules that define the criteria by which it should select scheduled payments for payment creation.
<b>payment shuffle</b>	In PeopleSoft Enterprise Campus Solutions, a process allowing payments that have been previously posted to a student's account to be automatically reapplied when a higher priority payment is posted or the payment allocation definition is changed.
<b>pending item</b>	In PeopleSoft Receivables, an individual receivable (such as an invoice, a credit memo, or a write-off) that has been entered in or created by the system, but hasn't been posted.
<b>PeopleCode</b>	PeopleCode is a proprietary language, executed by the PeopleSoft application processor. PeopleCode generates results based upon existing data or user actions. By using business interlink objects, external services are available to all PeopleSoft applications wherever PeopleCode can be executed.
<b>PeopleCode event</b>	An action that a user takes upon an object, usually a record field, that is referenced within a PeopleSoft page.
<b>PeopleSoft Internet Architecture</b>	The fundamental architecture on which PeopleSoft 8 applications are constructed, consisting of a relational database management system (RDBMS), an application server, a web server, and a browser.
<b>performance measurement</b>	In PeopleSoft Enterprise Incentive Management, a variable used to store data (similar to an aggregator, but without a predefined formula) within the scope of an incentive plan. Performance measures are associated with a plan calendar, territory, and participant. Performance measurements are used for quota calculation and reporting.
<b>period context</b>	In PeopleSoft Enterprise Incentive Management, because a participant typically uses the same compensation plan for multiple periods, the period context associates a plan context with a specific calendar period and fiscal year. The period context references the associated plan context, thus forming a chain. Each plan context has a corresponding set of period contexts.
<b>person of interest</b>	A person about whom the organization maintains information but who is not part of the workforce.

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

	number to determine the primary student attribute value that is used when you extract data to report on cohorts. The lowest number takes precedence.
<b>primary name type</b>	In PeopleSoft Enterprise Campus Solutions, the name type that is used to link the name stored at the highest level within the system to the lower-level set of names that an individual provides.
<b>process category</b>	In PeopleSoft Process Scheduler, processes that are grouped for server load balancing and prioritization.
<b>process group</b>	In PeopleSoft Financials, a group of application processes (performed in a defined order) that users can initiate in real time, directly from a transaction entry page.
<b>process definition</b>	Process definitions define each run request.
<b>process instance</b>	A unique number that identifies each process request. This value is automatically incremented and assigned to each requested process when the process is submitted to run.
<b>process job</b>	You can link process definitions into a job request and process each request serially or in parallel. You can also initiate subsequent processes based on the return code from each prior request.
<b>process request</b>	A single run request, such as a Structured Query Report (SQR), a COBOL or Application Engine program, or a Crystal report that you run through PeopleSoft Process Scheduler.
<b>process run control</b>	A PeopleTools variable used to retain PeopleSoft Process Scheduler values needed at runtime for all requests that reference a run control ID. Do not confuse these with application run controls, which may be defined with the same run control ID, but only contain information specific to a given application process request.
<b>product category</b>	In PeopleSoft Enterprise Incentive Management, indicates an application in the Enterprise Incentive Management suite of products. Each transaction in the PeopleSoft Enterprise Incentive Management system is associated with a product category.
<b>programs</b>	In PeopleSoft Enterprise Learning Management, a high-level grouping that guides the learner along a specific learning path through sections of catalog items. PeopleSoft Enterprise Learning Systems provides two types of programs—curricula and certifications.
<b>progress log</b>	In PeopleSoft Services Procurement, tracks deliverable-based projects. This is similar to the time sheet in function and process. The service provider contact uses the progress log to record and submit progress on deliverables. The progress can be logged by the activity that is performed, by the percentage of work that is completed, or by the completion of milestone activities that are defined for the project.
<b>project transaction</b>	In PeopleSoft Project Costing, an individual transaction line that represents a cost, time, budget, or other transaction row.
<b>promotion</b>	In PeopleSoft Promotions Management, a trade promotion, which is typically funded from trade dollars and used by consumer products manufacturers to increase sales volume.
<b>prospects</b>	In PeopleSoft Enterprise Campus Solutions, students who are interested in applying to the institution.  In PeopleSoft Enterprise Contributor Relations, individuals and organizations that are most likely to make substantial financial commitments or other types of commitments to the institution.
<b>publishing</b>	In PeopleSoft Enterprise Incentive Management, a stage in processing that makes incentive-related results available to participants.

<b>rating components</b>	In PeopleSoft Enterprise Campus Solutions, variables used with the Equation Editor to retrieve specified populations.
<b>record group</b>	A set of logically and functionally related control tables and views. Record groups help enable TableSet sharing, which eliminates redundant data entry. Record groups ensure that TableSet sharing is applied consistently across all related tables and views.
<b>record input VAT flag</b>	Abbreviation for <i>record input value-added tax flag</i> . Within PeopleSoft Purchasing, Payables, and General Ledger, this flag indicates that you are recording input VAT on the transaction. This flag, in conjunction with the record output VAT flag, is used to determine the accounting entries created for a transaction and to determine how a transaction is reported on the VAT return. For all cases within Purchasing and Payables where VAT information is tracked on a transaction, this flag is set to Yes. This flag is not used in PeopleSoft Order Management, Billing, or Receivables, where it is assumed that you are always recording only output VAT, or in PeopleSoft Expenses, where it is assumed that you are always recording only input VAT.
<b>record output VAT flag</b>	Abbreviation for <i>record output value-added tax flag</i> . See <i>record input VAT flag</i> .
<b>recname</b>	The name of a record that is used to determine the associated field to match a value or set of values.
<b>recognition</b>	In PeopleSoft Enterprise Campus Solutions, the recognition type indicates whether the PeopleSoft Enterprise Contributor Relations donor is the primary donor of a commitment or shares the credit for a donation. Primary donors receive hard credit that must total 100 percent. Donors that share the credit are given soft credit. Institutions can also define other share recognition-type values such as memo credit or vehicle credit.
<b>reference data</b>	In PeopleSoft Sales Incentive Management, system objects that represent the sales organization, such as territories, participants, products, customers, channels, and so on.
<b>reference object</b>	In PeopleSoft Enterprise Incentive Management, this dimension-type object further defines the business. Reference objects can have their own hierarchy (for example, product tree, customer tree, industry tree, and geography tree).
<b>reference transaction</b>	In commitment control, a reference transaction is a source transaction that is referenced by a higher-level (and usually later) source transaction, in order to automatically reverse all or part of the referenced transaction's budget-checked amount. This avoids duplicate postings during the sequential entry of the transaction at different commitment levels. For example, the amount of an encumbrance transaction (such as a purchase order) will, when checked and recorded against a budget, cause the system to concurrently reference and relieve all or part of the amount of a corresponding pre-encumbrance transaction, such as a purchase requisition.
<b>regional sourcing</b>	In PeopleSoft Purchasing, provides the infrastructure to maintain, display, and select an appropriate vendor and vendor pricing structure that is based on a regional sourcing model where the multiple ship to locations are grouped. Sourcing may occur at a level higher than the ship to location.
<b>relationship object</b>	In PeopleSoft Enterprise Incentive Management, these objects further define a compensation structure to resolve transactions by establishing associations between compensation objects and business objects.
<b>remote data source data</b>	Data that is extracted from a separate database and migrated into the local database.
<b>REN server</b>	Abbreviation for <i>real-time event notification server</i> in PeopleSoft MultiChannel Framework.
<b>requester</b>	In PeopleSoft eSettlements, an individual who requests goods or services and whose ID appears on the various procurement pages that reference purchase orders.

<b>reversal indicator</b>	In PeopleSoft Enterprise Campus Solutions, an indicator that denotes when a particular payment has been reversed, usually because of insufficient funds.
<b>role</b>	Describes how people fit into PeopleSoft Workflow. A role is a class of users who perform the same type of work, such as clerks or managers. Your business rules typically specify what user role needs to do an activity.
<b>role user</b>	A PeopleSoft Workflow user. A person's role user ID serves much the same purpose as a user ID does in other parts of the system. PeopleSoft Workflow uses role user IDs to determine how to route worklist items to users (through an email address, for example) and to track the roles that users play in the workflow. Role users do not need PeopleSoft user IDs.
<b>roll up</b>	In a tree, to roll up is to total sums based on the information hierarchy.
<b>run control</b>	A run control is a type of online page that is used to begin a process, such as the batch processing of a payroll run. Run control pages generally start a program that manipulates data.
<b>run control ID</b>	A unique ID to associate each user with his or her own run control table entries.
<b>run-level context</b>	In PeopleSoft Enterprise Incentive Management, associates a particular run (and batch ID) with a period context and plan context. Every plan context that participates in a run has a separate run-level context. Because a run cannot span periods, only one run-level context is associated with each plan context.
<b>search query</b>	You use this set of objects to pass a query string and operators to the search engine. The search index returns a set of matching results with keys to the source documents.
<b>search/match</b>	In PeopleSoft Enterprise Campus Solutions and PeopleSoft Enterprise Human Resources Management Solutions, a feature that enables you to search for and identify duplicate records in the database.
<b>seasonal address</b>	In PeopleSoft Enterprise Campus Solutions, an address that recurs for the same length of time at the same time of year each year until adjusted or deleted.
<b>section</b>	In PeopleSoft Enterprise Incentive Management, a collection of incentive rules that operate on transactions of a specific type. Sections enable plans to be segmented to process logical events in different sections.
<b>security event</b>	In commitment control, security events trigger security authorization checking, such as budget entries, transfers, and adjustments; exception overrides and notifications; and inquiries.
<b>serial genealogy</b>	In PeopleSoft Manufacturing, the ability to track the composition of a specific, serial-controlled item.
<b>serial in production</b>	In PeopleSoft Manufacturing, enables the tracing of serial information for manufactured items. This is maintained in the Item Master record.
<b>service impact</b>	In PeopleSoft Enterprise Campus Solutions, the resulting action triggered by a service indicator. For example, a service indicator that reflects nonpayment of account balances by a student might result in a service impact that prohibits registration for classes.
<b>service indicator</b>	In PeopleSoft Enterprise Campus Solutions, indicates services that may be either withheld or provided to an individual. Negative service indicators indicate holds that prevent the individual from receiving specified services, such as check-cashing privileges or registration for classes. Positive service indicators designate special services that are provided to the individual, such as front-of-line service or special services for disabled students.



<b>session</b>	<p>In PeopleSoft Enterprise Campus Solutions, time elements that subdivide a term into multiple time periods during which classes are offered. In PeopleSoft Contributor Relations, a session is the means of validating gift, pledge, membership, or adjustment data entry. It controls access to the data entered by a specific user ID. Sessions are balanced, queued, and then posted to the institution's financial system. Sessions must be posted to enter a matching gift or pledge payment, to make an adjustment, or to process giving clubs or acknowledgements.</p> <p>In PeopleSoft Enterprise Learning Management, a single meeting day of an activity (that is, the period of time between start and finish times within a day). The session stores the specific date, location, meeting time, and instructor. Sessions are used for scheduled training.</p>
<b>session template</b>	In PeopleSoft Enterprise Learning Management, enables you to set up common activity characteristics that may be reused while scheduling a PeopleSoft Enterprise Learning Management activity—characteristics such as days of the week, start and end times, facility and room assignments, instructors, and equipment. A session pattern template can be attached to an activity that is being scheduled. Attaching a template to an activity causes all of the default template information to populate the activity session pattern.
<b>setup relationship</b>	In PeopleSoft Enterprise Incentive Management, a relationship object type that associates a configuration plan with any structure node.
<b>share driver expression</b>	In PeopleSoft Business Planning, a named planning method similar to a driver expression, but which you can set up globally for shared use within a single planning application or to be shared between multiple planning applications through PeopleSoft Enterprise Warehouse.
<b>single signon</b>	With single signon, users can, after being authenticated by a PeopleSoft application server, access a second PeopleSoft application server without entering a user ID or password.
<b>source key process</b>	In PeopleSoft Enterprise Campus Solutions, a process that relates a particular transaction to the source of the charge or financial aid. On selected pages, you can drill down into particular charges.
<b>source transaction</b>	In commitment control, any transaction generated in a PeopleSoft or third-party application that is integrated with commitment control and which can be checked against commitment control budgets. For example, a pre-encumbrance, encumbrance, expenditure, recognized revenue, or collected revenue transaction.
<b>speed key</b>	See <i>communication key</i> .
<b>SpeedChart</b>	A user-defined shorthand key that designates several ChartKeys to be used for voucher entry. Percentages can optionally be related to each ChartKey in a SpeedChart definition.
<b>SpeedType</b>	A code representing a combination of ChartField values. SpeedTypes simplify the entry of ChartFields commonly used together.
<b>staging</b>	A method of consolidating selected partner offerings with the offerings from the enterprise's other partners.
<b>standard letter code</b>	In PeopleSoft Enterprise Campus Solutions, a standard letter code used to identify each letter template available for use in mail merge functions. Every letter generated in the system must have a standard letter code identification.
<b>statutory account</b>	Account required by a regulatory authority for recording and reporting financial results. In PeopleSoft, this is equivalent to the Alternate Account (ALTACCT) ChartField.

<b>step</b>	In PeopleSoft Sales Incentive Management, a collection of sections in a plan. Each step corresponds to a step in the job run.
<b>storage level</b>	In PeopleSoft Inventory, identifies the level of a material storage location. Material storage locations are made up of a business unit, a storage area, and a storage level. You can set up to four storage levels.
<b>subcustomer qualifier</b>	A value that groups customers into a division for which you can generate detailed history, aging, events, and profiles.
<b>Summary ChartField</b>	You use summary ChartFields to create summary ledgers that roll up detail amounts based on specific detail values or on selected tree nodes. When detail values are summarized using tree nodes, summary ChartFields must be used in the summary ledger data record to accommodate the maximum length of a node name (20 characters).
<b>summary ledger</b>	An accounting feature used primarily in allocations, inquiries, and PS/nVision reporting to store combined account balances from detail ledgers. Summary ledgers increase speed and efficiency of reporting by eliminating the need to summarize detail ledger balances each time a report is requested. Instead, detail balances are summarized in a background process according to user-specified criteria and stored on summary ledgers. The summary ledgers are then accessed directly for reporting.
<b>summary time period</b>	In PeopleSoft Business Planning, any time period (other than a base time period) that is an aggregate of other time periods, including other summary time periods and base time periods, such as quarter and year total.
<b>summary tree</b>	A tree used to roll up accounts for each type of report in summary ledgers. Summary trees enable you to define trees on trees. In a summary tree, the detail values are really nodes on a detail tree or another summary tree (known as the <i>basis</i> tree). A summary tree structure specifies the details on which the summary trees are to be built.
<b>syndicate</b>	To distribute a production version of the enterprise catalog to partners.
<b>system function</b>	In PeopleSoft Receivables, an activity that defines how the system generates accounting entries for the general ledger.
<b>TableSet</b>	A means of sharing similar sets of values in control tables, where the actual data values are different but the structure of the tables is the same.
<b>TableSet sharing</b>	Shared data that is stored in many tables that are based on the same TableSets. Tables that use TableSet sharing contain the SETID field as an additional key or unique identifier.
<b>target currency</b>	The value of the entry currency or currencies converted to a single currency for budget viewing and inquiry purposes.
<b>tax authority</b>	In PeopleSoft Enterprise Campus Solutions, a user-defined element that combines a description and percentage of a tax with an account type, an item type, and a service impact.
<b>template</b>	A template is HTML code associated with a web page. It defines the layout of the page and also where to get HTML for each part of the page. In PeopleSoft, you use templates to build a page by combining HTML from a number of sources. For a PeopleSoft portal, all templates must be registered in the portal registry, and each content reference must be assigned a template.
<b>territory</b>	In PeopleSoft Sales Incentive Management, hierarchical relationships of business objects, including regions, products, customers, industries, and participants.
<b>3C engine</b>	Abbreviation for <i>Communications, Checklists, and Comments engine</i> . In PeopleSoft Enterprise Campus Solutions, the 3C engine enables you to automate business processes that involve additions, deletions, and updates to communications, checklists,

and comments. You define events and triggers to engage the engine, which runs the mass change and processes the 3C records (for individuals or organizations) immediately and automatically from within business processes.

<b>3C group</b>	Abbreviation for <i>Communications, Checklists, and Comments group</i> . In PeopleSoft Enterprise Campus Solutions, a method of assigning or restricting access privileges. A 3C group enables you to group specific communication categories, checklist codes, and comment categories. You can then assign the group inquiry-only access or update access, as appropriate.
<b>TimeSpan</b>	A relative period, such as year-to-date or current period, that can be used in various PeopleSoft General Ledger functions and reports when a rolling time frame, rather than a specific date, is required. TimeSpans can also be used with flexible formulas in PeopleSoft Projects.
<b>trace usage</b>	In PeopleSoft Manufacturing, enables the control of which components will be traced during the manufacturing process. Serial- and lot-controlled components can be traced. This is maintained in the Item Master record.
<b>transaction allocation</b>	In PeopleSoft Enterprise Incentive Management, the process of identifying the owner of a transaction. When a raw transaction from a batch is allocated to a plan context, the transaction is duplicated in the PeopleSoft Enterprise Incentive Management transaction tables.
<b>transaction state</b>	In PeopleSoft Enterprise Incentive Management, a value assigned by an incentive rule to a transaction. Transaction states enable sections to process only transactions that are at a specific stage in system processing. After being successfully processed, transactions may be promoted to the next transaction state and “picked up” by a different section for further processing.
<b>Translate table</b>	A system edit table that stores codes and translate values for the miscellaneous fields in the database that do not warrant individual edit tables of their own.
<b>tree</b>	The graphical hierarchy in PeopleSoft systems that displays the relationship between all accounting units (for example, corporate divisions, projects, reporting groups, account numbers) and determines roll-up hierarchies.
<b>tuition lock</b>	In PeopleSoft Enterprise Campus Solutions, a feature in the Tuition Calculation process that enables you to specify a point in a term after which students are charged a minimum (or <i>locked</i> ) fee amount. Students are charged the locked fee amount even if they later drop classes and take less than the normal load level for that tuition charge.
<b>unclaimed transaction</b>	In PeopleSoft Enterprise Incentive Management, a transaction that is not claimed by a node or participant after the allocation process has completed, usually due to missing or incomplete data. Unclaimed transactions may be manually assigned to the appropriate node or participant by a compensation administrator.
<b>universal navigation header</b>	Every PeopleSoft portal includes the universal navigation header, intended to appear at the top of every page as long as the user is signed on to the portal. In addition to providing access to the standard navigation buttons (like Home, Favorites, and signoff) the universal navigation header can also display a welcome message for each user.
<b>update access</b>	In PeopleSoft Enterprise Campus Solutions, a type of security access that permits the user to edit and update data.  See also <i>inquiry access</i> .
<b>user interaction object</b>	In PeopleSoft Sales Incentive Management, used to define the reporting components and reports that a participant can access in his or her context. All Sales Incentive Management user interface objects and reports are registered as user interaction objects. User interaction objects can be linked to a compensation structure node through a compensation relationship object (individually or as groups).

<b>variable</b>	In PeopleSoft Sales Incentive Management, the intermediate results of calculations. Variables hold the calculation results and are then inputs to other calculations. Variables can be plan variables that persist beyond the run of an engine or local variables that exist only during the processing of a section.
<b>VAT exception</b>	Abbreviation for <i>value-added tax exception</i> . A temporary or permanent exemption from paying VAT that is granted to an organization. This terms refers to both VAT exoneration and VAT suspension.
<b>VAT exempt</b>	Abbreviation for <i>value-added tax exempt</i> . Describes goods and services that are not subject to VAT. Organizations that supply exempt goods or services are unable to recover the related input VAT. This is also referred to as exempt without recovery.
<b>VAT exoneration</b>	Abbreviation for <i>value-added tax exoneration</i> . An organization that has been granted a permanent exemption from paying VAT due to the nature of that organization.
<b>VAT suspension</b>	Abbreviation for <i>value-added tax suspension</i> . An organization that has been granted a temporary exemption from paying VAT.
<b>warehouse</b>	A PeopleSoft data warehouse that consists of predefined ETL maps, data warehouse tools, and DataMart definitions.
<b>work order</b>	In PeopleSoft Services Procurement, enables an enterprise to create resource-based and deliverable-based transactions that specify the basic terms and conditions for hiring a specific service provider. When a service provider is hired, the service provider logs time or progress against the work order.
<b>worker</b>	A person who is part of the workforce; an employee or a contingent worker.
<b>workset</b>	A group of people and organizations that are linked together as a set. You can use worksets to simultaneously retrieve the data for a group of people and organizations and work with the information on a single page.
<b>worksheet</b>	A way of presenting data through a PeopleSoft Business Analysis Modeler interface that enables users to do in-depth analysis using pivoting tables, charts, notes, and history information.
<b>worklist</b>	The automated to-do list that PeopleSoft Workflow creates. From the worklist, you can directly access the pages you need to perform the next action, and then return to the worklist for another item.
<b>XML schema</b>	An XML definition that standardizes the representation of application messages, component interfaces, or business interlinks.
<b>yield by operation</b>	In PeopleSoft Manufacturing, the ability to plan the loss of a manufactured item on an operation-by-operation basis.
<b>zero-rated VAT</b>	Abbreviation for <i>zero-rated value-added tax</i> . A VAT transaction with a VAT code that has a tax percent of zero. Used to track taxable VAT activity where no actual VAT amount is charged. Organizations that supply zero-rated goods and services can still recover the related input VAT. This is also referred to as exempt with recovery.

# Index

## A

### Action commands

Build Tree Path command 90

Command Line 98

Database Validation command 93, 94, 95, 96

*See Also* Expect No Matching Records option; validation association; validation declaration; validation definition; validation execution

Press Push Button command 87

*See Also* clickable bitmap options; press push button options

Press Toolbar Button command 81, 83

*See Also* Custom Button option; Grid Scroll Button option; Select Grid Tab option; Standard Button option

scripting a Command Line

command 98, 99

*See Also* capturing a current form; sending a command line command

scripting the Build Tree Path

command 91, 92, 93

*See Also* adding a parent node or child; removing a parent node or child; using literal values; using variable values

scripting the Database Validation

command 96, 98

*See Also* associating a validation; declaring a validation; executing a validation

scripting the Press Push Button

command 88

*See Also* clicking a bitmap; pressing a push button

scripting the Press Toolbar Button

command 83, 85, 86

*See Also* pressing a custom button; pressing a standard button; pressing the grid scroll button; selecting a grid tab

scripting the Select ComboBox Item

command 90

scripting the Select Grid Row

command 77, 78, 79

*See Also* clicking by cell content; clicking by row number; performing grid row operations

scripting the Type To command 60, 61, 70, 72

*See Also* using the Header Control or Grid Column list; using the Source of Input list; using the value selection list

Select ComboBox Item command 89

select Grid Line command 76

*See Also* Operation Type list

Select Grid Row command 75, 76, 77

*See Also* Action on Grid Row list; Grid Columns list; Source of Row Number list

Type To command 52, 53, 59

*See Also* clear source of input; Header Control/Grid Column list; value selection list

Type To Command 53

*See Also* Source of Input list

Action on Grid Row list 76

double-click grid row 76

double-click grid row button 77

position grid row for add/edit 77

single-click on grid row 76

single-click on grid row button 76

Add to Repository command 146

adding a parent node or child to a tree path 92

adding a script to the repository 151

adding a value to a variable 66

adding command lines 105

additional documentation xiv

Application command 26, 38

application fundamentals xiii

Application Interconnect command 30, 44

arranging multiple PeopleSoft AutoPilot forms 20

Assigning a default value to an external variable 115

- assigning a form interconnect visual assist value to a header control or grid column 71
- assigning a literal value to a header control or grid column 70
- assigning a UDC visual assist value to a header control or grid column 74
- assigning a valid values list value to a header control or grid column 71
- assigning a variable value to a header control or grid column 71
- Assigning properties to a script 150
- associating a validation 96
- Associating a Validation
  - zero decimals 96
- Automatic script playback
  - configuration 118
  - breakpoints 118
  - event stream 119
  - playback during script creation 118
  - playback speed 118
  - storage and display of playback data 118

## B

- Breakpoints 118
- broken links 112
- Browse Repository Scripts command 146
  - Browse Scripts form 147
  - Select Script form 146
- Browse Result Sets tab of Test Results form 155
- Browse Scripts form 147
- Browsing for repository scripts 151
- Build Tree Path command 90
- building a tree path using literal values 92
- Building a tree path using variable values 91

## C

- Caption bar 13
- capturing a current form 99
- Categories tab for script property pages 145
- change the scope of a variable 63
- changing the size of the PeopleSoft AutoPilot form 20
- Check In command 149
- Checking in a script 153

- Checking out a script 153
- Checkout command 148
- clear source of input 59
- clearing an input from a header control or grid column 70
- Click by Cell Content option 76
- Click by Row Number option 76
- Clickable bitmap options 87
- Clicking a bitmap 88
- Clicking by cell content 78
- Clicking by row number 78
- clicking options in a header 73
- Command Line 98
- command lines 102
- Command option 14
- command pane 8, 9
  - insert button 10
  - lists 9
- command pane lists 9
- Comments tab for script property pages 145
- comments, submitting xviii
- common elements xviii
- Completing the script 139
- concatenating a variable 67
- conditional statements 56
- Configure tab 17
- contact information xviii
- context commands 26
  - Application command 38
  - Application Interconnect command 30, 44
  - Form command 32, 47
    - See Also* Form list; Next Form list
  - Grid Column command 33, 48
  - Header command 32, 48
  - processing options command 31
  - QBE command 33, 48
  - scripting context commands 37
  - UBE command 27, 28, 29, 30, 37, 38, 41, 42, 43, 44
    - See Also* data selection; launching a UBE; options; Print command; printing; processing options; selecting data; setting processing options; submission; submitting
- Context commands
  - Application command 26
- continue playback to a breakpoint 126
- Continue to Breakpoint command 121

- create a list of literal values 61
- create a sample script
  - declaring a variable 131
- Create Work With Batch Versions
  - Commands option 27
- Creating a playlist 160
- Creating a sample script 129
  - completing the script 139
  - creating a valid values list 133
  - entering data to a QBE line 137
  - finding records in the database 138
  - launching a new form 131
  - launching an application and form 130
  - returning to a previous form 137
  - selecting records and deleting them from the database 138
  - setting the value of a variable 136
  - typing data to a grid column 134, 135
    - See Also* updating the repeat count
  - typing data to a header control 132
  - updating the database 135
  - updating the repeat count 135
- Creating a valid values list 133
- creating a Valid Values list from a simple database query 62
- creating a variable to confirm validation success 68
- creating a variable to store a valid values list count 68
- Creating variable links 114
- cross-references xvii
- Custom Button option 81
  - form exit 82
  - row exit 82
- Customer Connection website xiv

## D

- Database Validation command 93
  - Expect No Matching Records option 96
    - validation association 94, 95
      - See Also* key selection value;
      - validation value
    - validation declaration 94
    - validation definition 94
    - validation execution 95
- declare a variable 63
- declaring a validation 96
- declaring a variable 131
- Declaring a variable as external 115
- default values for external variables 110

- default values for variables 56
- delete command lines 107
- Deleting a script from the repository 151
- Details tab for script property pages 145
- Directories tab 17
- documentation
  - printed xiv
  - related xiv
  - updates xiv
- Double-click grid row 76
- double-click grid row button 77
- drag and drop 104

## E

- edit a context command line 108
- edit an action command line 107
- edit command lines 105
  - editing a context command line 108
  - editing an action command line 107
- Entering data to a QBE line 137
- Event stream 119, 154
- Execute FASTPATH option for UBE
  - command 27
- executing a validation 98
- Exit Work With Batch Versions command option 30
- expand and collapsing a node 105
- Expand/Collapse button 103
- Expect No Matching Records option 96
- Expect No Printer Selection Window option 30
- external variables 56, 110

## F

- fail a script 127
- File option 13
- Finding records in the database 138
- floating the toolbar 22
- Form command 32, 47
  - Form list 32
  - Next Form list 32
- Form exit 82
- form interconnect visual assist value as a source of input 58
- Form list 32

## G

- General tab for script property pages 144
- Generate Valid Values List option 15

- Get Copy command 148
- Getting a copy of a script 152
- global variables 55
- glossary 163
- Grid Column command 33, 48
- Grid Columns list 77
- Grid Scroll Button option 83

**H**

- Header command 32, 48
- Header Control/Grid Column list 53
- Help option 19

**I**

- Ignore Breakpoints during Playback 122
- Ignoring breakpoints in the script 123
- Include Local Script option 16
- Include Reposited Script option 16
- including scripts 113
- Indented nodes 103
- insert a Wait command 126
- Insert button 10
- Inserting a comment in the script 123
- Insertion cursor 103

**J**

- Jde.ini tab of Test Results form 155
- Jde.log tab of Test Results form 156
- Jdedebug.log tab of Test Results form 156

**K**

- Key Selection value 95

**L**

- launch a UBE 38
  - automatic submission 41
  - from a menu 38
  - from a row menu 39
  - from another UBE 41
- launche a UBE
  - from a report menu 39
- Launching a new form 131
- Launching an application and form 130
- Link Variable form 111
- literal value as a source of input 53
- local variables 55

**M**

- MAF (Multiple Application Framework)

- HTML 14
- Tools options 14
- Managing script testing
  - creating a playlist 160
  - resetting a test 162
  - running a test 161
  - saving a playlist 161
  - viewing test results 161
- manipulating the PeopleSoft AutoPilot form 20
  - arranging multiple PeopleSoft AutoPilot forms 20
  - changing the form size 20
  - sizing panes 21
- manipulating the PeopleSoft AutoPilot toolbar 21
  - floating the toolbar 22
  - resizing the toolbar 22
- Manual script playback options 120
  - Continue to Breakpoint command 121
  - Ignore Breakpoints during Playback 122
  - play a chosen branch of the script command 120
  - play from a chosen script line command 120
  - Play from Cursor command 120
  - Play from Top command 120
  - play to the next script line command 120
  - playback pause 123
  - script comment 121
  - Stop Playback command 122
  - Toggle a Breakpoint command 120
  - Wait Before Proceeding command 121
- Menu bar 13
  - Command option 14
  - File option 13
  - Help option 19
  - Play option 14
  - Tools option 14, 15, 16
    - See Also* Generate Valid Values List; Include Local Script; Include Reposited Script; options for configuring PeopleSoft AutoPilot; Results; Select PeopleSoft Client
  - View option 14
  - Window option 19
- Messages tab of Test Results form 156
- MMA Partners xiv



- modify scripts 105
  - deleting command lines 107
  - editing command lines 105, 107, 108
    - See Also* editing a context command line; editing an action command line
  - expanding and collapsing a node 105
- modifying scripts
  - adding command lines 105
  - moving command lines 105
- moving command lines 105
- My Checkouts 149

## N

- Naming conventions for saved scripts 146
- Next Form list 32
- nodes 106
  - drag and drop 104
  - Expand/Collapse button 103
  - indented 103
  - repeat count 104
- Nodes
  - parallel 103
- notes xvii

## O

- Operation Type list 76
  - Click by Cell Content option 76
  - Click by Row Number option 76
- option combinations for UBEs 28
- options for configuring PeopleSoft
  - AutoPilot 16
    - Configure tab 17
    - Directories tab 17
    - PeopleSoft HTML tab 17
    - PeopleSoft Java tab 17
    - Playback tab 18
    - Playback tab options 17
    - Sign-on tab 17
    - Speed tab 17

## P

- panes in the PeopleSoft AutoPilot form 8
- parallel nodes 103
- Pausing playback 124
- PeopleBooks
  - ordering xiv
- PeopleCode, typographical
  - conventions xvi

- PeopleSoft application fundamentals xiii
- PeopleSoft AutoPilot
  - action commands 52, 75, 77, 87, 89, 90, 91, 93, 96, 98
    - See Also* Build Tree Path command; Command Line; Database Validation command; Press Push Button command; scripting a Command Line command; scripting the Build Tree Path command; scripting the Database Validation command; scripting the Select ComboBox Item command; scripting the Select Grid Row command; Select ComboBox Item command; Select Grid Row command; Type To command
- caption bar 13
- command pane 8, 9, 10
  - See Also* insert button; lists
- context commands 26, 27, 30, 31, 32, 33, 37, 38, 44, 45, 47, 48
  - See Also* Application command; Application Interconnect command; Form command; Grid Column command; Header command; Processing Options command; Processing Options command context commands; Processing Options command Processing Options command ; QBE command; scripting context commands; UBE command
- creating a sample script 129
- defined 3
- form panes 8
- menu bar 13, 19
  - See Also* Help option
- Menu bar 13, 14, 19
  - See Also* Command option; File option; Play option; Tools option; Window option
- playing back the script 117, 122
  - See Also* running script playback
- Script pane 8, 10, 101, 105, 108, 113
  - See Also* modifying scripts; reusing scripts; script retention and reuse; structure
- scripting action commands 51
- scripting context commands 25

- Status bar 19
  - storing scripts and test results 141, 142, 150, 154
    - See Also* script reporting; script repository; working with the script repository
  - Tool bar 19
  - toolbar buttons 8
  - user interface 7
  - PeopleSoft AutoPilot form
    - manipulating the form 20, 21
      - See Also* arranging multiple PeopleSoft AutoPilot forms; changing the form size; sizing panes
    - manipulating the toolbar 21, 22
      - See Also* floating the toolbar; resizing the toolbar
  - PeopleSoft AutoPilot Test Manager
    - managing script testing 160, 161, 162
      - See Also* creating a playlist; resetting a test; running a test; saving a playlist; viewing test results
    - script display pane 157
    - script storage pane 157
    - test results pane 158
    - toolbar 159
  - PeopleSoft AutoPilot Test Manager toolbar 159
  - PeopleSoft HTML tab 17
  - PeopleSoft Java tab 17
  - Performing grid row operations 79
  - Play a chosen branch of the script command 120
  - play back the script
    - running script playback 125, 126, 127
      - See Also* failing a script; inserting a Wait command; playing the script to a breakpoint; setting transaction times in the script; toggling a breakpoint
  - Play from a chosen script line command 120
  - Play from Cursor command 120
  - Play from Top command 120
  - Play option 14
  - play the script to a breakpoint 125
  - Play to the next script line command 120
  - Playback during script creation 118
  - Playback pause 123
  - Playback speed 118
  - Playback tab 18
  - Playback tab options 17
  - playback the script
    - running script playback 126
      - See Also* continuing playback to a breakpoint
  - Playing a branch of the script 124
  - Playing back the script 117
    - running script playback 122, 123, 124
      - See Also* ignoring breakpoints in the script; inserting a comment in the script; pausing playback; playing a branch of the script; playing the script from a chosen cursor position; playing the script from the top
    - script playback functions 118, 120
      - See Also* automatic script playback configuration; manual script playback options
  - Playing the script from a chosen cursor position 123
  - Playing the script from the top 123
  - Position grid row for add/edit 77
  - prerequisites xiii
  - Press Push Button command 87
    - clickable bitmap options 87
    - press push button options 87
  - Press push button options 87
  - Press Toolbar Button command
    - Custom Button option 81, 82
      - See Also* form exit; row exit
    - Grid Scroll Button option 83
    - Select Grid Tab option 83
    - Standard Button option 81
  - Pressing a custom button 85
  - Pressing a push button 88
  - Pressing a standard button 83
  - Pressing the grid scroll button 86
  - printed documentation xiv
  - printing a UBE 44
  - Processing Options command 31
  - Property pages for scripts 143
    - Categories tab 145
    - Comments tab 145
    - Details tab 145
    - General tab 144
- Q**
- QBE command 33, 48

Query by included scripts 149  
 Querying for included scripts in the repository 154

## R

recursive value searching 111  
 related documentation xiv  
 removing a parent node or child from a tree path 93  
 repeat count 104  
 Resetting a test 162  
 resizing the toolbar 22  
 Results option 16  
 Returning to a previous form 137  
 reusing scripts 113  
   including scripts 113  
 Reusing scripts  
   assigning a default value to an external variable 115  
   creating variable links 114  
   declaring a variable as external 115  
 Row exit 82  
 run script playback  
   continuing playback to a breakpoint 126  
   failing a script 127  
   inserting a Wait command 126  
   playing the script to a breakpoint 125  
   setting transaction times in the script 127  
   toggling a breakpoint 125  
 Running a test 161  
 Running script playback  
   ignoring breakpoints in the script 123  
   inserting a comment in the script 123  
   pausing playback 124  
   playing a branch of the script 124  
   playing the script from a chosen cursor position 123  
   playing the script from the top 123  
 running the Type To command  
   typing data in a QBE line 75

## S

Saving a playlist 161  
 Script comment 121  
 Script deletion from repository 148  
 Script display pane 157  
 script includes 109

Script pane 8, 10, 101  
   modifying scripts 105, 107  
     *See Also* adding command lines; deleting command lines; editing command lines; expanding and collapsing a node; moving command lines  
 reusing scripts 113, 114, 115  
   *See Also* assigning a default value to an external variable; creating variable links; declaring a variable as external; including scripts  
 script retention and reuse 108, 109, 112  
   *See Also* script includes; script saving; script sharing; variable linking between scripts  
 structure 101, 102, 103, 106  
   *See Also* command lines; insertion cursor; nodes  
 Script pane structure 101  
   command lines 102  
   insertion cursor 103  
   nodes 103, 104, 106  
     *See Also* drag and drop; expand/collapse button; indented; parallel; repeat count  
 Script playback functions  
   automatic script playback configuration 118, 119  
     *See Also* breakpoints; event stream; playback during script creation; playback speed; storage and display of playback data  
 manual script playback options 120, 121, 122, 123  
   *See Also* Continue to Breakpoint command; Ignore Breakpoints during Playback; play a chosen branch of the script command; play from a chosen script line command; Play from Cursor command; Play from Top command; play to the next script line command; playback pause; script comment; Stop Playback command; Toggle a Breakpoint command; Wait Before Proceeding command  
 Script reporting 154  
   event stream 154  
   Test Results form 155, 156

- See Also* Browse Result Sets
    - tab; JDE.INI tab; jde.log tab;
    - jdidebug.log tab; Messages tab;
    - Summary tab
- Script repository 142
  - Add to Repository command 146
  - Browse Repository Scripts
    - command 146, 147
    - See Also* Browse Scripts form; Select Script form
  - Check In command 149
  - Checkout command 148
  - Get Copy command 148
  - My Checkouts 149
  - naming conventions for saved scripts 146
  - property pages for scripts 143, 144, 145
    - See Also* Categories tab; Comments tab; Details tab; General tab
  - query by included scripts 149
  - script deletion 148
  - Undo Checkout command 149
- script retention and reuse 108
  - script includes 109
  - script saving 109
  - script sharing 112
  - variable linking between scripts 109, 110, 111
    - See Also* default values for external variables; external variables; variable links
- Script saving 109
- script sharing 112
- Script storage pane 157
- script the Type To command
  - using the Source of Input list 61
- Script the Type To command
  - using the Source of Input list 61, 63
  - See Also* valid values list; variable
- Scripting a Command Line command
  - capturing a current form 99
  - sending a command line command 99
- scripting action commands 51
- scripting context commands 25, 37
- scripting the Build Tree Path command
  - adding a parent node or child 92
  - using literal values 92
- Scripting the Build Tree Path command
  - removing a parent node or child 93
  - using variable values 91
- scripting the Database Validation command
  - declaring a validation 96
  - executing a validation 98
- Scripting the Database Validation command
  - associating a validation 96
- Scripting the Press Push Button command
  - clicking a bitmap 88
  - pressing a push button 88
- Scripting the Press Toolbar Button
  - command
    - pressing a custom button 85
    - pressing a standard button 83
    - pressing the grid scroll button 86
    - selecting a grid tab 86
- Scripting the Select ComboBox Item
  - command 90
- Scripting the Select Grid Row
  - command 77
    - clicking by cell content 78
    - clicking by row number 78
    - performing grid row operations 79
- scripting the Type To command 72
  - clicking options in a header 73
  - running 75
    - See Also* typing data in a QBE line
  - typing data in a grid column 73
  - typing data in a header control 73
  - using the Source of Input list 69, 70
    - See Also* clearing an input from a header control or grid column; UDC visual assist value
  - using the value selection list 71, 74
    - See Also* assigning a form interconnect visual assist value; assigning a UDC visual assist value; assigning a variable value
- Scripting the Type To command
  - using the Header Control or Grid Column list 60
  - using the Source of Input list 61, 69
    - See Also* form interconnect visual assist value; literal value
  - using the value selection list 70, 71
    - See Also* assigning a literal value; assigning a valid values list value
- Security for repositored scripts 152
- Select ComboBox Item command 89
- select data for a UBE 42
- Select Grid Line command

- Action on Grid Row list 76
  - See Also* single-click on grid row
- Operation Type list 76
- Select Grid Row command 75
  - Action on Grid Row list 76, 77
    - See Also* double-click grid row;
      - double-click grid row button;
      - position grid row for add/edit;
      - single-click on grid row button
- Grid Columns list 77
- Operation Type list 76
  - See Also* Click by Cell Content option;
    - Click by Row Number option
- Source of Row Number list 77
- Select Grid Tab option 83
- Select PeopleSoft Client option 16
- Select Script form 146
- Selecting a grid tab 86
- Selecting records and deleting them from the database 138
- sending a command line command 99
- set conditional statements 65
- set the value of a variable 64
- setting processing options for a UBE 43
- Setting the value of a variable 136
- Setting transaction times in the script 127
- Sign-on tab 17
- Single click on grid row 76
- single-click on grid row button 76
- sizing panes in the PeopleSoft AutoPilot form 21
- Source of Input list 53
  - form interconnect visual assist value 58
  - literal value 53
  - UDC visual assist value 58
  - valid values list 53
  - variable 54, 55, 56, 57, 58
    - See Also* addition; concatenation;
      - conditional statements; default values; external; global; local; scope; subtraction; system; valid values count; validation success; watch list
- Source of Row Number list 77
- Speed tab 17
- Standard Button option 81
- Status bar 19
- Stop Playback command 122
- Storage and display of playback data 118
- Storing scripts and test results 141

- assigning properties to a script 150
- PeopleSoft AutoPilot Test Manager 157, 158, 159, 160
  - See Also* managing script testing;
    - script display pane; script storage pane; test results pane; toolbar
- script reporting 154, 155
  - See Also* event stream; Test Results form
- script repository 142, 143, 146, 148, 149
  - See Also* Add to Repository command;
    - Browse Repository Scripts command; Check In command; Checkout command; Get Copy command; My Checkouts; naming conventions for saved scripts; property pages for scripts; query by included scripts; script deletion; Undo Checkout command
- working with the script repository 150, 151, 152, 153, 154
  - See Also* adding a script to the repository; assigning properties to a script; assigning security to a repositied script; browsing for repository scripts; checking in a script; checking out a script; deleting a script from the repository; getting a copy of a script; querying for included scripts; undoing script checkout
- submit a UBE 41
- subtracting a value from a variable 67
- suggestions, submitting xviii
- Summary tab of Test Results form 155
- system variables 57

## T

- terms 163
- Test Results form 155
  - Browse Result Sets tab 155
  - JDE.INI tab 155
  - jde.log tab 156
  - jdidebug.log tab 156
  - Messages tab 156
  - Summary tab 155
- Test results pane 158
- toggle a breakpoint 125
- Toggle a Breakpoint command 120

- Tool bar 19
  - Tool bar buttons 8
  - Tools option 14
    - Generate Valid Values List 15
    - Include Local Script 16
    - Include Reposited Script 16
    - options for configuring PeopleSoft
      - AutoPilot 16, 17, 18
        - See Also* Configure tab; Directories tab; PeopleSoft HTML tab; PeopleSoft Java tab; Playback tab; Playback tab options; Sign-on tab; Speed tab
    - Results 16
    - Select PeopleSoft Client 16
  - Type To command 52
    - clear source of input 59
    - Header Control/Grid Column list 53
    - Source of Input list 53, 54, 58
      - See Also* form interconnect visual assist value; literal value; UDC visual assist value; valid values list; variable
    - value selection list 59
  - Type To Command
    - Source of Input list 53
  - typing data in a grid column 73
  - typing data in a header control 73
  - typing data in a QBE line 75
  - Typing data to a grid column 134
    - updating the repeat count 135
  - Typing data to a header control 132
  - typographical conventions xvi
- U**
- UBE command 27, 37
    - data selection 28
    - launching a UBE 38, 39, 41
      - See Also* automatic submission; from a menu; from a report menu; from a row menu; from another UBE
    - options 27, 28
      - See Also* combinations; Create Work With Batch Versions Commands; Execute FASTPATH
    - Print command 30
      - See Also* Exit Work With Batch Versions command option; Expect No Printer Selection Window
    - option; Wait for UBE to complete print option
    - printing 44
    - processing options 29
    - selecting data 42
    - setting processing options 43
    - submission 28
    - submitting 41
  - UBE data selection 28
  - UBE options 27
    - combinations 28
    - Create Work With Batch Versions Commands 27
    - Execute FASTPATH 27
  - UBE Print command 30
    - Exit Work With Batch Versions command option 30
    - Expect No Printer Selection Window option 30
    - Wait for UBE to complete print option 30
  - UBE processing options 29
  - UBE submission 28
  - UDC visual assist value as a source of input 58
  - Undo Checkout command 149
  - Undoing script checkout 153
  - update the repeat count 63
  - update the value of an existing variable 65
  - Updating the database 135
  - Updating the repeat count 135
  - use a literal value as a source of input 61
  - use a valid values list as a source of input 61
  - use a variable in the script 65
  - use a variable value as a source of input 63
  - use the Source of Input list
    - literal value 61
    - valid values list 61, 62
      - See Also* creating a list of literal values; creating a valid values list from a simple database query
  - Valid Values list 63
    - See Also* updating the repeat count
  - variable 63, 64, 65
    - See Also* changing the scope; declaring; setting conditional statements; setting the value; updating the value; using in the script
  - Use the Source of Input list

- variable 63
- using a form interconnect visual assist value as a source of input 69
- using a UDC visual assist value as a source of input 69
- using a Valid Values list in the script 62
- using the Header Control or Grid Column list 60
- using the Source of Input list 61
  - clearing an input to a header control or grid column 70
  - form interconnect visual assist value 69
  - UDC visual assist value 69
  - Valid Values list 62
    - See Also* using in the script
- variable 66, 67, 68
  - See Also* adding a value; concatenating a variable; confirming validation success; storing a valid values list count; subtracting a value
- using the value selection list 70
  - assigning a form interconnect visual assist value 71
  - assigning a literal value 70
  - assigning a UDC visual assist value 74
  - assigning a valid values list value 71
  - assigning a variable value 71

## V

- valid values count 58
- Valid values list
  - creating a list of literal values 61
- Valid Values list
  - creating a Valid Values list from a simple database query 62
  - updating the repeat count 63
  - using in the script 62
- valid values list as a source of input 53
- validating
  - zero decimals 96
- validation
  - zero decimals 96
- validation association
  - Key Selection value 95
  - validation value 95
- Validation association 94
- Validation declaration 94
- Validation definition 94
- validation execution 95

- Validation Success variable 58
- validation value 95
- Value Selection list 59
- variable
  - adding a value 66
  - addition 56
  - changing the scope 63
  - concatenating a variable 67
  - concatenation 57
  - conditional statements 56
  - confirming validation success 68
  - declaring 63
  - default values 56
  - external 56
  - global 55
  - local 55
  - scope 55
  - setting the value 64
  - storing a valid values list count 68
  - subtracting a value 67
  - subtraction 57
  - system 57
  - updating the value 65
  - using in the script 65
  - valid values count 58
  - validation success 58
  - watch list 58

## Variable

- setting conditional statements 65
- variable addition 56
- variable as a source of input 54
- variable concatenation 57
- variable linking between scripts 109
  - default values for external variables 110
  - external variables 110
  - variable links 111, 112
    - See Also* broken links; link variable form; recursive value searching
- variable links 111
  - broken links 112
  - Link Variable form 111
  - recursive value searching 111
- variable scope 55
- variable subtraction 57
- Variable Watch list 58
- View option 14
- Viewing test results 161
- visual cues xvii

## W

- Wait Before Proceeding command 121
- wait for UBE to complete print option 30
- warnings xvii
- Window option 19
- Working with the script repository 150
  - adding a script to the repository 151
  - assigning security to a repositied script 152
  - browsing for repository scripts 151
  - checking in a script 153
  - checking out a script 153
  - deleting a script from the repository 151
  - getting a copy of a script 152
  - querying for included scripts 154
  - undoing script checkout 153