



EnterpriseOne Tools 8.94

PeopleBook: PeopleSoft Virtual AutoPilot

November 2004

EnterpriseOne Tools 8.94 PeopleBook: PeopleSoft Virtual AutoPilot
SKU E1_TOOLS8.94TVA-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	
About This PeopleBook	vii
PeopleSoft Application Prerequisites.....	vii
PeopleSoft Application Fundamentals.....	vii
Documentation Updates and Printed Documentation.....	viii
Obtaining Documentation Updates.....	viii
Ordering Printed Documentation.....	viii
Additional Resources.....	ix
Typographical Conventions and Visual Cues.....	x
Typographical Conventions.....	x
Visual Cues.....	xi
Country, Region, and Industry Identifiers.....	xi
Currency Codes.....	xii
Comments and Suggestions.....	xii
Common Elements Used in PeopleBooks.....	xii
 Preface	
PeopleSoft EnterpriseOne Virtual AutoPilot Preface.....	xv
PeopleSoft Products.....	xv
PeopleSoft Virtual AutoPilot.....	xv
 Chapter 1	
Getting Started with PeopleSoft Tools Virtual AutoPilot.....	1
PeopleSoft Tools Virtual AutoPilot Overview.....	1
Implementing PeopleSoft Virtual AutoPilot.....	1
 Chapter 2	
Understanding Data Capture for PeopleSoft Virtual AutoPilot Scripts.....	3
Overall Testing Tools Architecture.....	3
Data Capture Components.....	3
Understanding PeopleSoft AutoPilot.....	4
PeopleSoft AutoPilot Tasks.....	4
PeopleSoft Software and PeopleSoft AutoPilot Code.....	5

Event Stream.....	5
PeopleSoft AutoPilot Playback Results Detail Table (F97214).....	6

Chapter 3

Understanding PeopleSoft Virtual AutoPilot Components.....7

PeopleSoft AutoPilot Components.....	7
Virtual Script Editor.....	8
Virtual Script Editor Features.....	8
Event Pane.....	10
Event Graph.....	11
Parameter Detail Pane.....	11
Script List Pane.....	12
Parameter Value Linking.....	13
Source and Target Parameter Identification.....	13
Looping Identification.....	15
HRequest Handle Value Linking.....	16
Thread Identification.....	17
Timing Interval Maintenance.....	17
PeopleSoft Virtual AutoPilot Script Generation.....	18
Virtual Script Player.....	18
Virtual Script Player Features.....	19
Virtual Script Player Initialization File Parameters.....	19
Virtual Script Player Command Line.....	22
Environment Initialization.....	22
Modes of Operation.....	23
Preprocessing of Valid Values List Data.....	23
Valid Values List Processing.....	23
Date Formatting.....	23
Script Failure.....	24
Virtual Script Player Limitations.....	24
VSMEditor.....	24
VSM Editor Features.....	24
All Virtual Scripts List Box.....	25
Master Scripts List Box.....	25
VSM Files.....	25
Virtual Runner.....	25
Virtual Runner Features.....	25
Player Session Columns.....	26
Actions Tools.....	26

Chapter 4

Creating Virtual Scripts.....	29
Understanding Script Creation.....	29
Capturing and Importing Test Results.....	29
Understanding Test Results.....	30
Capturing Test Results.....	30
Importing Test Results.....	31
Viewing Test Results.....	31
Editing Virtual Scripts.....	32
Understanding Virtual Script Generation.....	32
Using the Find Feature.....	32
Adding Loops.....	33
Value-Linking Parameters.....	33
Linking Values in Inquiry Scripts.....	34
Linking Values in Entry Scripts.....	35
Generating PeopleSoft Virtual AutoPilot Scripts.....	35
Creating Master Scripts.....	36

Chapter 5

Running Virtual Scripts.....	37
Understanding Virtual Script Runs.....	37
Running Virtual Scripts from a Single Workstation.....	37
Understanding Virtual Script Runs from a Single Workstation.....	37
Prerequisite.....	37
Running Virtual Scripts from the Command Line.....	38
Running Virtual Scripts Using Virtual Runner.....	38
Launching and Managing Multiple Script Playback.....	40
Understanding LoadRunner.....	40
Defining Scripts.....	40
Defining the Host Machine.....	40
Defining Virtual Users.....	41
Gathering LoadRunner Results.....	41
Running Virtual Playback from the LoadRunner Controller.....	41

Chapter 6

Understanding Special Considerations for Simulated Playback.....	43
Playback Problem Simulation.....	43
Playback Timing.....	43

Event Synchronization.....	43
API Playback Timing.....	44
Interthread Timing.....	44
Call Level.....	45
Synchronous and Asynchronous Calls.....	45
Think Times.....	46

Chapter 7

Troubleshooting PeopleSoft Virtual AutoPilot.....	49
Understanding PeopleSoft Virtual AutoPilot Troubleshooting.....	49
Reviewing the PeopleSoft Virtual AutoPilot Log File to Locate Errors.....	49
Understanding the PeopleSoft Virtual AutoPilot Log File.....	49
Finding Error Entries in the PeopleSoft Virtual AutoPilot Log File.....	50
Locating the Log File in the Event of Early Script Failure.....	50
Setting the MessageLevel Parameter.....	51
Identifying Environment Problems.....	51
Understanding Environment Problems.....	51
Diagnosing Environment Problems.....	51
Investigating System Errors.....	52
Debugging PeopleSoft Virtual AutoPilot Scripts.....	52
Understanding PeopleSoft Virtual AutoPilot Debugging.....	53
Displaying Business Function Parameters.....	53
Diagnosing Business Function Failures in PeopleSoft Explorer.....	54
Researching Value-Linking Errors in the Virtual Script Editor.....	54
Verifying That Value Linking Is Functioning.....	55
Identifying and Correcting Variable Value-Linking Errors.....	56
Verifying the Validity of PeopleSoft Virtual AutoPilot Script Data.....	56
Identifying and Correcting Duplicate Key Errors.....	57
Rectifying Irregular Transaction Times.....	57
Preventing Multiple Script Playback Problems.....	58
Correcting Uninitialized User Handle Errors.....	59

Glossary of PeopleSoft Terms.....	61
--	-----------

Index	81
--------------------	-----------

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.

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
Bold	Indicates PeopleCode function names, business function names, event names, system function names, method names, language constructs, and PeopleCode reserved words that must be included literally in the function call.
<i>Italics</i>	Indicates field values, emphasis, and PeopleSoft 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.

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

Common Elements Used in PeopleBooks

As of Date

The last date for which a report or process includes data.

Business Unit

An ID that represents a high-level organization of business information. You can use a business unit to define regional or departmental units within a larger organization.

Description

Enter up to 30 characters of text.

Effective Date	The date on which a table row becomes effective; the date that an action begins. For example, to close out a ledger on June 30, the effective date for the ledger closing would be July 1. This date also determines when you can view and change the information. Pages or panels and batch processes that use the information use the current row.
Once, Always, and Don't Run	<p>Select Once to run the request the next time the batch process runs. After the batch process runs, the process frequency is automatically set to Don't Run.</p> <p>Select Always to run the request every time the batch process runs.</p> <p>Select Don't Run to ignore the request when the batch process runs.</p>
Process Monitor	Click to access the Process List page, where you can view the status of submitted process requests.
Report Manager	Click to access the Report List page, where you can view report content, check the status of a report, and see content detail messages (which show you a description of the report and the distribution list).
Request ID	An ID that represents a set of selection criteria for a report or process.
Run	Click to access the Process Scheduler request page, where you can specify the location where a process or job runs and the process output format.
SetID	<p>An ID that represents a set of control table information, or TableSets. TableSets enable you to share control table information and processing options among business units. The goal is to minimize redundant data and system maintenance tasks. When you assign a setID to a record group in a business unit, you indicate that all of the tables in the record group are shared between that business unit and any other business unit that also assigns that setID to that record group. For example, you can define a group of common job codes that are shared between several business units. Each business unit that shares the job codes is assigned the same setID for that record group.</p>
Short Description	Enter up to 15 characters of text.
User ID	An ID that represents the person who generates a transaction.

PeopleSoft EnterpriseOne Virtual AutoPilot

Preface

This preface discusses the Virtual AutoPilot PeopleBook.

PeopleSoft Products

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

PeopleSoft Virtual AutoPilot

This PeopleBook covers PeopleSoft Virtual AutoPilot, a member of the EnterpriseOne Tools suite. You use PeopleSoft Virtual AutoPilot to create a script that simulates the activities of PeopleSoft software as it handles the workload generated by many users. Its chapters describe the tool in general and defines its features and functionality.

CHAPTER 1

Getting Started with PeopleSoft Tools Virtual AutoPilot

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

PeopleSoft Tools Virtual AutoPilot Overview

Several APIs enable PeopleSoft software to interact with any database or application server. The APIs communicate with PeopleSoft software's middleware, which serves as the conduit for runtime data flowing from the client workstation to the server and back again. PeopleSoft automated testing tools capture this data, which provides you with the raw material to build a virtual script that accurately simulates PeopleSoft software processes.

Other Sources of Information

In the planning phase of your implementation, take advantage of all PeopleSoft sources of information, including the installation guides and troubleshooting information, including the installation guides and troubleshooting information. A complete list of these resources appears in the preface in *About These PeopleBooks*, with information about where to find the most current version of each.

See Also

About These PeopleBooks Preface

Implementing PeopleSoft Virtual AutoPilot

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

- You must have a valid EnterpriseOne user account.
- PeopleSoft Virtual AutoPilot must be installed and configured for you to use.

CHAPTER 2

Understanding Data Capture for PeopleSoft Virtual AutoPilot Scripts

This chapter provides an overview of automated testing tools architecture and data capture, and discusses:

- Overall testing tools architecture.
- Data capture components

Overall Testing Tools Architecture

The following components of PeopleSoft automated testing tools architecture work together to capture, record, and store data about PeopleSoft software processes, including the parameters of all API calls and all other PeopleSoft software runtime events:

- PeopleSoft AutoPilot, which enables you to write and play back a script to test PeopleSoft software applications and to configure script playback so that PeopleSoft AutoPilot captures and saves playback data
- Hooks, or code, that reside in PeopleSoft software and in PeopleSoft AutoPilot that capture and record data generated by the playback of a PeopleSoft AutoPilot script
- Event stream, which is a time-stamped, chronological record of each PeopleSoft AutoPilot and PeopleSoft software event that occurs during script playback
- PeopleSoft AutoPilot Playback Results Detail Table (F97214), which stores the event stream

The placement of PeopleSoft software code is important to the creation of PeopleSoft Virtual AutoPilot scripts. Because this code is positioned at the boundary between the PeopleSoft software runtime engine and the PeopleSoft software middleware, it captures data passing to the JDB and CallObject APIs before the APIs are routed to servers by the OCM. Therefore, you can reuse PeopleSoft Virtual AutoPilot scripts regardless of changes to OCM mappings.

Data Capture Components

Creating a virtual script requires that you first capture data from a PeopleSoft software session in which you launch an application, press buttons, enter data to header controls, and so on. The automated testing tool architecture of which PeopleSoft AutoPilot is a part enables you to capture the events of a PeopleSoft software session by writing a script, configuring it for event capture, playing it back, and storing its results. You accomplish these tasks using the following three components:

Component	Function
PeopleSoft AutoPilot	Enables you to write a script, play it back in PeopleSoft software, and save the results of the playback.
Code that resides in PeopleSoft software and in PeopleSoft AutoPilot	Captures and records the data generated by PeopleSoft AutoPilot and by PeopleSoft software during script playback.
PeopleSoft AutoPilot Playback Results Detail Table (F97214)	Stores the data generated during script playback as an event stream, which is a continuous record of every PeopleSoft software and PeopleSoft AutoPilot event that occurred during script playback.

Understanding PeopleSoft AutoPilot

This section discusses:

- PeopleSoft AutoPilot tasks.
- PeopleSoft software and PeopleSoft AutoPilot code.
- Event stream.
- PeopleSoft AutoPilot Playback Results Detail Table (F97214).

PeopleSoft AutoPilot Tasks

The process of creating a virtual script begins with PeopleSoft AutoPilot, which you use to write a script that tests PeopleSoft software processes. Playing back a PeopleSoft AutoPilot script simulates PeopleSoft software activities, but only as initiated by one user. However, you can capture the results of script playback, including the processes generated, save the data, and use it to create a virtual script that you run to simulate more than one user.

PeopleSoft AutoPilot Script Creation

To begin the process of capturing data, you first write a PeopleSoft AutoPilot script to test software processes such as launching applications, pressing buttons, entering data to header controls, and so on.

PeopleSoft AutoPilot Playback Configuration

You capture data about software processes by playing back the PeopleSoft AutoPilot script, but you must configure playback for data capture. Your configuration choices establish how much data you capture and ensure that PeopleSoft AutoPilot saves the data.

You can capture data at one of two levels:

- Level 1 captures data only for initiating API calls that run alone or call other APIs. If you select this option, you capture data about only these APIs.
- All API calls capture data not only about level 1 API calls but about any API calls spawned by a level 1 API.

You also configure script playback to save and display results data after playback.

PeopleSoft AutoPilot Script Playback

After you have written a script and configured playback to capture the results, you play back the script. PeopleSoft AutoPilot captures the playback data using internal code and code placed in the software.

PeopleSoft Software and PeopleSoft AutoPilot Code

Code is strategically placed in PeopleSoft AutoPilot and in 32 JDB functions and 1 CallObject function in PeopleSoft software to gather and store during PeopleSoft AutoPilot script playback.

The placement of PeopleSoft software code provides the following advantages for creating PeopleSoft Virtual PeopleSoft AutoPilot scripts:

- Comprehensive data capture. Because code is positioned to capture both JDB and JDE CallObject API calls, you capture both database and business function activity.
- Simplified script maintenance. Because the code resides in slightly more than 30 JDB and CallObject functions combined, making changes in PeopleSoft software code is relatively easy.
- Flexibility in running scripts. Data that you capture can be run independent of platform or Object Configuration Manager (OCM) mapping considerations.

Code placed in PeopleSoft software performs the following functions that lay the groundwork for the creation of a PeopleSoft Virtual AutoPilot script:

- Captures parameter data on JDB and CallObject API calls that occur during the playback of a PeopleSoft AutoPilot script
- Writes the parameter data to a file-mapping object that PeopleSoft software shares with PeopleSoft AutoPilot
- Writes data on event rules, button presses, and event timing to the file-mapping object

Code placed in PeopleSoft AutoPilot performs the following functions that lay the groundwork for the creation of a PeopleSoft Virtual AutoPilot script:

- Writes data on PeopleSoft AutoPilot events to the file-mapping object
- Copies the PeopleSoft software and PeopleSoft AutoPilot data in the shared file space into a BLOB (Binary Large Object) field in the PeopleSoft AutoPilot Playback Results Detail Table (F97214)
- Enables the PeopleSoft AutoPilot user to access the PeopleSoft AutoPilot Playback Results Detail Table (F97214)

PeopleSoft Virtual AutoPilot scripts that you create using the data captured from an PeopleSoft AutoPilot script are platform independent and can be run on any operating system with any OCM mappings because PeopleSoft software code captures API call data before it reaches the OCM for mapping.

Event Stream

You generate an event stream when you run an PeopleSoft AutoPilot script that you have configured to capture playback results. The event stream is a time-stamped, chronological record of every PeopleSoft AutoPilot and PeopleSoft software event that occurs during playback, including:

- JDB and CallObject API calls
- Thread identification
- Event rules
- PeopleSoft software error and warning messages

- PeopleSoft AutoPilot events confirming that the script and PeopleSoft software are on the same form

PeopleSoft AutoPilot Playback Results Detail Table (F97214)

The PeopleSoft AutoPilot Playback Results Detail Table (F97214) contains the results of all PeopleSoft AutoPilot playback sessions that you have captured and saved. You can access the event stream for a script playback session through PeopleSoft AutoPilot, the Virtual Script Editor, or EnterpriseOne Analyzer Tool.

Using PeopleSoft AutoPilot, you can view a summary of every event that occurred during script playback. For example, you can view the following information for any test on the test Results form:

- Results sets
- Summary (Script, Release, Machine, and so on)
- JDE.INI and JDE and JDEDEBUG.LOG
- Screen captures
- Messages
- Results

From the Virtual Script Editor or from EnterpriseOne Analyzer Tool, you can view each event in more detail. For example, you can select an API call and view the input and output parameter values for the call. If you import an event stream record from the database table to the Virtual Script Editor, you can modify the record so that you can play it as a PeopleSoft Virtual AutoPilot script.

CHAPTER 3

Understanding PeopleSoft Virtual AutoPilot Components

This chapter discusses:

- PeopleSoft AutoPilot components.
- Virtual Script Editor.
- Virtual Script Player.
- VSM Editor.
- Virtual Runner.

PeopleSoft AutoPilot Components

You use components that are external to PeopleSoft Virtual AutoPilot (Virtual AutoPilot) to capture, record, and store data generated by an PeopleSoft AutoPilot playback session. These components include the following:

- PeopleSoft AutoPilot
- Data-capture code in PeopleSoft AutoPilot and in PeopleSoft software
- Event stream
- PeopleSoft AutoPilot Playback Results Detail Table (F97214), where the event stream from the playback session is stored

The internal components of PeopleSoft Virtual AutoPilot enable you to complete the PeopleSoft Virtual AutoPilot scripting process. These components are:

- Virtual Script Editor
- Virtual Script Player
- VSMEditor
- Virtual Runner

You use the internal components of Virtual User Tool to complete the following tasks:

- Import an event stream into the Virtual Script Editor
- Modify the event stream by adding rules that govern the passing of parameters and looping (repeated JDB Fetch calls to complete a database inquiry)
- Use the Virtual Script Editor to automatically add rules to handle thread identification and hRequest handles
- Generate a PeopleSoft Virtual AutoPilot script

- Run a PeopleSoft Virtual AutoPilot script on the Virtual Script Player
 - Use the VSMEditor to concatenate a series of individual PeopleSoft Virtual AutoPilot scripts into one master script
 - Use Virtual Runner to manage script playback, either from a single workstation or from multiple workstations
- You also can manage script playback using LoadRunner from Mercury Interactive.

Virtual Script Editor

This section discusses:

- Virtual Script Editor features.
- Event pane.
- Event graph.
- Parameter detail pane.
- Script list pane.
- Parameter value linking.
- Source and target parameter identification.
- Looping identification.
- HRequest handle value linking.
- Thread identification.
- Timing interval maintenance.
- PeopleSoft Virtual AutoPilot script generation.

Virtual Script Editor Features

The Virtual Script Editor enables you to create and generate a PeopleSoft Virtual AutoPilot (Virtual AutoPilot) script that you can use to simulate the activity of many concurrent users. Working with the Virtual Script Editor represents the second step in a three-step process of producing a PeopleSoft Virtual AutoPilot script playback session. You create a PeopleSoft Virtual AutoPilot script playback session by completing the following steps in the following order:

1. Capture data generated by PeopleSoft AutoPilot script playback and store the event stream in a results repository.
2. Use the Virtual Script Editor to modify an event stream and to generate a PeopleSoft Virtual AutoPilot script that contains all the information required by the Virtual Script Player to simulate the activities of the system's runtime engine.
3. Play back a modified event stream (the PeopleSoft Virtual AutoPilot script) using the Virtual Script Player.

The event stream is a chronological, time-stamped record of every event that occurs during the playback of an PeopleSoft AutoPilot script, including the following:

- User input
- Processing performed by the run-time engine, such as thread creation

- Event rules; informative messages
- API calls to the PeopleSoft middleware

PeopleSoft AutoPilot performs no editing during the process. The event stream represents a record of the events that have already occurred. You cannot edit it by adding, deleting, or reordering data. To change it, you must generate a new one by modifying an existing PeopleSoft AutoPilot script, or by creating a new script and then replaying it.

Using the Virtual Script Editor, you can do the following:

- View the titles of all the scripts whose results you stored in the PeopleSoft AutoPilot Playback Results Detail Table (F97214)
- Import an event stream
- View an event stream as a single, continuous record
- View the timing of events by category, represented in a horizontal bar graph
- Select an individual API call and view the input values sent to the server and the output values returned to the client workstation
- Create links between parameters of API calls so that parameter values can be passed between calls during virtual script playback
- Identify and designate loops so that the virtual script can handle repetitive processing tasks, such as database retrieval

The Virtual Script Editor helps you address problems that you encounter when trying to create a script that you can run to simulate activities in a dynamic client/server system. Problems that you might encounter include the following:

- Identifying API parameters that require dynamic values
- Providing a way to pass values dynamically between API call parameters to avoid data conflict and record contention
- Making the values of hRequest handle parameter values dynamic to simulate concurrent user activity
- Synchronizing timing between events during script playback to keep processing running regardless of network stress placed on the server
- Synchronizing timing between data-dependent APIs in threads running asynchronously to avoid one API starting before another has finished processing
- Identifying repetitive processing tasks, such as database inquiry, so that the Virtual Script Player can efficiently simulate the work of the software

The Virtual Script Editor handles the following virtual script creation tasks automatically:

- Linking values of parameters in separate API calls so that values can be passed, provided that the calls meet certain criteria
- Storing the values of hRequest handles as variables
- Storing identification of thread IDs
- Storing information about time gaps between events in a single thread and between interthread-dependent events

You perform the following virtual script creation tasks manually:

- Linking values of parameters in separate API calls that do not meet all the criteria required for automatic value linking
- Identifying repetitive processes, such as database inquiry, as loops

After you have completed these manual tasks, you use the Virtual Script Editor to generate the virtual script. The Virtual Script Player receives from the Virtual Script Editor all of the information necessary to run the virtual script.

Event Pane

You click the Import button on the Virtual Script Editor form to import the results from an PeopleSoft AutoPilot script playback session. The Virtual Script Editor populates the event pane with the event stream. The event stream contains a time stamping of each event. Therefore, you can review PeopleSoft AutoPilot events or API calls during playback that might have taken an unusual amount of time to run.

You use the event pane to view data about the following kind of playback events:

- CallObject APIs
- JDB APIs
- PeopleSoft AutoPilot events
- Event rules
- Informative messages, including system errors and warnings
- Thread creation

The event pane also contains the following columnar information about each event:

- Timing information, such as the start, end, and elapsed time of an event
- Thread identification
- hUser handle identification
- hRequest handle identification
- Call level
- Message entry identifying the event
- Message information about an event, such as JDB call to open a table from memory cache

The message entry for each event includes an abbreviation that identifies the type of event that occurred. The following table summarizes the abbreviations and the type of event that each represents:

Abbreviation in Message Column of Event Stream	Type of Script Event
JDB	Database API call
RTE	CallObject API call
EVR	Event rule
LOG	System warning message
ERR	System error message

Abbreviation in Message Column of Event Stream	Type of Script Event
MSG	PeopleSoft AutoPilot message
AUT	Action in PeopleSoft AutoPilot (for example, typing to control)
THR	Thread action

You use the following buttons to change the view in the event pane:

Generate	Enables you to generate a PeopleSoft Virtual AutoPilot script. Click this button only after you have finished editing the event stream in the Virtual Script Editor.
Filter	Enables you to remove unwanted events from the list by applying criteria found in the Filter form.
View Log	Enables you to look at the log produced when you generate the PeopleSoft Virtual AutoPilot script. The log includes the number of lines in the script and the number of errors, if any.

Event Graph

You can view playback events by category in a horizontal bar graph by choosing the Graph option in the Virtual Script Editor. While the event stream pane presents the events of an PeopleSoft AutoPilot script playback vertically, in a single chronological stream, the event graph presents the events horizontally across a timeline.

You can break up the chronology by message type, such as JDB API calls or event rules.

The event graph provides you with another detailed snapshot of activity that occurred during PeopleSoft AutoPilot script playback. You can focus on events of unusual duration, which can be helpful in debugging applications, analyzing network activity, or rewriting and rerunning the original PeopleSoft AutoPilot script.

Parameter Detail Pane

You can view the parameters that make up an API call by clicking an API call event line in the event pane. The pane that appears shows the name of each parameter in the call and its value, if any. For example, the detail pane might display the value of the user handle parameter that a JDB call passes to the database.

This detail pane provides a complete snapshot of each API call at a given point in time. For example, the pane shows arrows that indicate the flow of data that occurred during the call. An arrow on the left side of the box next to the name of a parameter indicates that the call passed the value from the client workstation to the JDENET or database driver. An arrow on the right side of the box indicates that the call returned data from the server. In some cases, a box contains both arrows, indicating that data flowed in both directions.

The parameter detail pane offers a before-and-after snapshot of script playback. Before playback, parameters for a CallObject API, such as BatchOpenOnInitialization, contain no batch number or batch date parameter values. After playback, these parameters contain returned values.

The parameter detail pane also displays the parameters of API calls that pass an environment handle to the database.

Finally, many API calls contain a request handle that points to a particular place in memory that the run-time engine has allocated for the call. The parameter for the request handle appears in the parameter detail pane if the API call used a request handle.

The ability of the PeopleSoft AutoPilot and PeopleSoft software hooks to capture data at this level of detail is critically important to PeopleSoft Virtual AutoPilot because the goal of PeopleSoft Virtual AutoPilot is to simulate, as closely as possible, the actual activities of the system. If the PeopleSoft Virtual AutoPilot script does not have the complete parameters of an API call, it cannot accurately model the activities of the system and its interaction with the client workstation, the database server, and the application server.

Script List Pane

The script list pane on the Virtual Script Editor form displays in chronological order the PeopleSoft AutoPilot script playback results that you saved.

The script list pane displays script result information in the following columns:

Column	Description
Test	Database ID number assigned to each PeopleSoft AutoPilot script playback session
Client	ID of the workstation on which you ran the test
Start Time	Date and time at which you ran the test
Elapsed	Time it took the test to run to a successful conclusion, failure, or cancellation
Environment	System environment against which you ran the test
Release	PeopleSoft software release against which you ran the test
Script	Name that you assigned to the test
Status	Result of the test-success, failure, or cancellation

After you select a script, you select one of the following buttons to manipulate the form view:

Filter	Enables you to remove PeopleSoft AutoPilot script playback results that you do not want, using criteria on the Filter form.
Import	Imports into the Virtual Script Editor the event stream from a test result that you select.
Refresh	Refreshes the script list pane from the database.
Delete	Removes one or more tests from the database.

See Also

[Chapter 6, “Understanding Special Considerations for Simulated Playback,” Call Level, page 45](#)

Parameter Value Linking

After you import an event stream into the Virtual Script Editor, you are ready to create the virtual script. Using the Target Parameters and Source Parameters panes, you complete the task of value linking. Value linking ensures that the virtual script can pass parameter values from one API to another. You identify a value-containing parameter in a source parameter API call and link the value to a target parameter in another API call. This process ensures the passing of a parameter value from one API to another API that requires the value.

In addition, the values contained in many API call parameters must be dynamic. For example, each time a user performs voucher entry, the PeopleSoft software creates a new batch number, a function that is essential to prevent the creation of duplicate keys. Value linking ensures that the Virtual Script Player can simulate this function. When you link the parameter value of two API calls, the Virtual Script Editor stores the value as a variable, and the value changes each time you run a virtual script.

For example, a script might call the business function `BatchOpenOnInitialization`. For the parameter `ICU`, which is the batch number, suppose the API returns the value 5056. In turn, the script might call the business function `BeginDoc`, which uses the value 5056 as an input to the `ICU` parameter. To simulate multiple script playback, the value 5056 must change in order to reflect the new batch numbers returned each time people using the system make these API calls. As long as you have linked the parameters, the batch number parameter value will change each time you run a virtual script.

Value linking simulates the application logic that is used to run PeopleSoft software operations. It codifies the relationship between one API call and another. When you run the virtual script, the Virtual Script Editor passes to the Virtual Script Player the ID number of the source parameter that you link to the target parameter. The Virtual Script Player uses this information to pass parameter values between API calls.

Several types of data necessary to run a virtual script are candidates for value linking:

- The client host name, which could change any time a script is played back.
- Next numbers, which must change each time a script is run in order to avoid producing duplicate data that would break the script.
- Valid values lists used in PeopleSoft AutoPilot scripts, which must be designated as such in a PeopleSoft Virtual AutoPilot script so that, during run time, the Virtual Script Player draws new values from the list rather than using the same value repeatedly.

Source and Target Parameter Identification

The Virtual Script Editor provides detailed information about API calls in the event stream when you click the Link Parameters button in the toolbar. The Virtual Script Editor identifies the API calls made during script playback as source parameters or target parameters. A source parameter contains a value that the system passes to a parameter in another API call. The parameter receiving the value is the target parameter.

Information about the source and target parameters appears in separate panes of the Virtual Script Editor form. Each of the panes contains the following information about API calls made during script playback:

Column Heading in Target Parameters Pane and in Source Parameters Pane	Information Displayed
ID	Displays in the source parameter pane a value that identifies that parameter. If you value-link the source parameter to a target parameter, the source parameter ID value appears next to the target parameter, along with a chain-link symbol indicating that you have linked the parameters.
Start Time	Specifies the time the event occurred during playback.
Thread	Identifies the thread generated by the system's run-time engine in which the event occurred.
Label	Identifies the data dictionary alias of the parameter.
Value	Shows the value of the parameter contained within the JDB or CallObject API call.
Comment	Contains the variable name of a business function parameter and the type of data that it contains.
Event	Identifies the specific JDB or CallObject API called or PeopleSoft AutoPilot event in which a value was entered.

To see the complete set of parameters for an API call that occurred during PeopleSoft AutoPilot playback, click an item in either pane. The Virtual Script Editor displays the parameter names and values for the selected call. Arrows indicate the direction of the flow of data.

The detail panes provide a snapshot of the API calls that the applications generated during PeopleSoft AutoPilot playback. You can examine the parameter values and the flow of the data to help determine, for example, a parameter value used in one API call that the system passed to another API call later in the script.

To find the parameter of an API call in an event line, you might have to click a node in the detail pane to expand a tree. For example, for a JDB call, find the Value node in the detail pane and expand the tree to expose all of the column parameters in the database table. You then can search the column parameters for the source parameter for which you are looking.

Automatic Value Linking of API Call Parameters

When you click the Link Parameters button on the Virtual Script Editor, the tool automatically links some source parameters of API calls to target parameters of other API calls. The Virtual Script Editor accomplishes this automatic linking according to a set of rules. The following rules govern the automatic linking of API calls:

- Data must have been entered in PeopleSoft AutoPilot.
- The value of the target parameter must exactly match the value of the source parameter.
- The data dictionary ID of the target parameter must exactly match the data dictionary ID of the source parameter.

The Virtual Script Editor finds those parameters that meet each of these conditions and automatically links them.

Looping Identification

PeopleSoft Virtual AutoPilot requires a method to handle repetitive processing, such as that which occurs when you click the Find button on a Find/Browse form to perform a database inquiry. In this situation, a JDB Fetch call might return any number of values from the database. Looping rules provide a way to identify these repetitive retrievals. Having identified all of them in a single step, you can more easily link the values of source and target parameters.

You can specify the number of times you want the API to return to the database to retrieve values to be used as parameters in the script. This capability enables you to more accurately simulate the load placed on the system. You can increase or decrease this number, but the actual number of matches that the API returns, based on the inquiry command that you write in the PeopleSoft AutoPilot script, will likely determine the number of loops that you specify.

Manual Value Linking of API Call Parameters

If the source and target parameter do not meet all three of the following you must link them manually by clicking a parameter in each pane, and then clicking the Link Parameters button in the Virtual Script Editor tool bar.

- Source parameter must come from an Autopilot event (type to grid cell, for example)
- Data dictionary items must (called *label* in Virtual Script Editor) match in source and target
- Values must match in source and target

In deciding the target parameter value to link to a source parameter value, you:

- Match data dictionary aliases
- Match parameter values
- Select, in general, the event in the Source Parameter pane whose start time most closely matches the start time of the event in the Target Parameters pane

You do not have to link the values of source and target parameters when:

- APIs do not contain data dictionary items
- An API call returns a zero or null value for the source parameter that might be value-linked to a target
- The data flow of the source parameter is indicated as bi-directional, but the input value and the return values are the same

You code as literal the values of any parameters that meet at least one of these criteria by clicking the Mark Literal button on the Virtual Script Editor form.

The content of the PeopleSoft AutoPilot script also plays an important role in your decisions on value linking. If the PeopleSoft AutoPilot script that you write contains a literal value that the script writes to a grid column or header control, you cannot make that literal value dynamic by linking. The Virtual Script Player will be forced to use that literal value repeatedly during PeopleSoft Virtual AutoPilot script playback.

Because you cannot make literal values dynamic, avoid using them often in a PeopleSoft Virtual AutoPilot script. The entry of the same value to a grid column or header control by multiple users does not accurately simulate the way people use the system. To set up a more realistic scenario when you write the PeopleSoft AutoPilot script, create valid values lists containing more than one value. During PeopleSoft Virtual AutoPilot script playback, the Virtual Script Player goes to the `.atd` directory on your hard drive to retrieve the list's values, and then it cycles through them, entering a different value in each simulated playback session until it reaches the end of the list, when it returns to the top of the list and repeats the cycle.

Dynamic Loop Creation

You create a dynamic loop in the PeopleSoft AutoPilot script by writing a command to press the Find button on a Find/Browse form. This command triggers a string of JDB API calls, culminating in a Fetch call.

PeopleSoft AutoPilot and PeopleSoft software code records all of these events during playback; however, thousands of events might exist in an PeopleSoft AutoPilot script. The Virtual Script Editor offers an easy way to locate the PeopleSoft AutoPilot press Find button event and the repetitive processing that occurred because of the event. When you type the word Find in the locator space in the Virtual Script Editor form, the Virtual Script Editor highlights the first line in the event stream pane that contains the word Find. Then, you can scroll down the pane to the Fetch call.

Following the Fetch call, you can scroll through the event stream to locate the series of calls that resulted from the Fetch. This series of calls constitutes the loop.

Dynamic Loop Designation

The Virtual Script Editor enables you to designate dynamic loops in the PeopleSoft Virtual AutoPilot script. By doing so, you add looping rules to the script. These rules enable the Virtual Script Player to perform the repetitive processing that the system performs.

When you right-click the Fetch line in the event stream and select Add Loop, the Virtual Script Editor produces a Dynamic Loop Manager form, which you use to apply a dynamic loop and to establish the rules by which the Virtual Script Player manages the loop at virtual run time.

You can instruct the Virtual Script Player to run the inquiry loop until the data runs out, or you can instruct it to loop a specific number of times.

When you click the Apply Loop button in the Dynamic Loop Manager, you establish the looping rule. The Virtual Script Editor indicates the loop in the event stream, in the Source Parameters pane, and in the Target Parameters pane by graying the sequence of API calls that are part of the loop.

Dynamic Loop Editing

You can edit established looping rules before generating a PeopleSoft Virtual AutoPilot script. You also can undo the loop if, for example, a series of calls does not constitute a loop. The ability to edit dynamic loops provides an added measure of control over the creation of PeopleSoft Virtual AutoPilot scripts.

To edit the loop, you right-click the Fetch command line in the event pane and select Edit Loop Details. On the Dynamic Loop Manager form, you can change the number of times you want the script to loop, or you can select the Undo Loop option to remove the loop.

HRequest Handle Value Linking

The Virtual Script Editor automatically stores hRequest handle parameter values for JDB API calls. This value represents the address of a memory block that the system allocates for storing information about an open table. The address provides you with entry to the database each time you need to open a table to perform a Fetch, FetchKeyed, SelectKeyed, FetchMatchingKey, or CloseTable function. However, when you create a PeopleSoft Virtual AutoPilot script and play it back, the hRequest handle parameter value probably changes. Playback could not continue if this value were constant.

The Virtual Script Editor handles the problem by storing the hRequest handle parameter value as a variable and passing the variable to the Virtual Script Player during playback. The value of the hRequest handle variable changes to reflect the new address of a database table opened during script playback.

You can view the hRequest handle returned from the original API database call by clicking a call in the Source Parameters pane and viewing the details of the call. The Virtual Script Editor displays in the detail pane the request handle returned from the OpenTable API call.

If a database API call, such as OpenTable, leads to additional API calls, such as FetchKeyed and CloseTable, the Virtual Script Player passes the new memory address of the opened table to these subsequent calls. During virtual playback, the subsequent APIs use the new handle to run SQL statements and to close the table.

Thread Identification

The Virtual Script Editor also stores the idThread numbers that PeopleSoft AutoPilot gathers into the event stream during script playback. These identifier numbers represent the synchronous and asynchronous threads generated by the runtime engine. The runtime engine assigns each event to a thread and tags each thread with a number.

During virtual script playback, the Virtual Script Editor passes idThread parameters to the Virtual Script Player, which assigns different idThreads to each event and associates each script event with its new identifier.

Note. During Virtual User Tool script playback, the Virtual Script Player rennumbers the original threads generated during PeopleSoft AutoPilot script playback. The Virtual Script Editor's role is to store the thread identification information and to pass it on through the virtual script.

Timing Interval Maintenance

The Virtual Script Editor also automatically handles problems of timing that might emerge in the creation of a Virtual User Tool script. The time-stamped event stream log of events captures the length of time elapsed between each event. However, after you create a Virtual User Tool script, you do not know the different scenarios in which the Virtual User Tool script runs. For example:

- The workstation on which the script runs might be simulating 50 users.
- The power of the workstation might differ from the one on which the original script data was captured.
- The server against which the Virtual User Tool script runs might be more or less powerful than the server against which the original script ran.

These factors combine to make it likely that the time required by a Virtual User Tool script to run will differ from the time that the original script required to run.

The Virtual Script Editor handles this problem by preserving in the virtual script the time intervals that existed between events when you ran the original script. The time intervals represent the length of time required to carry out the processing between events. Thus, even if an API call during virtual script playback takes longer to carry out than the API call in the original script, the Virtual Script Player preserves the original time difference between one API call and the next.

The Virtual Script Player initialization file also contains timing parameters that govern the playback of the Virtual User Tool script. You can adjust, to a limited extent, some of these parameters; for example, you can adjust how fast the Virtual User Tool script plays back.

See Also

[Chapter 3, “Understanding PeopleSoft Virtual AutoPilot Components,” Virtual Script Player Initialization File Parameters, page 19](#)

[Chapter 6, “Understanding Special Considerations for Simulated Playback,” Playback Timing, page 43](#)

PeopleSoft Virtual AutoPilot Script Generation

The PeopleSoft Virtual AutoPilot script is the output from the Virtual Script Editor and the input to the Virtual Script Player. PeopleSoft Virtual AutoPilot scripts appear in text file form with a header and the edited list of events that you captured during script playback, imported into the Virtual Script Editor, and edited, both manually and automatically.

For ease and consistency of interpretation, each event in the script is structured in a particular way. For example, each event begins with the letter *e* and is followed by a unique identifying number. In addition, each script identifies the environment and the network user, and contains an open table handle. However, it is not necessary that you look at a PeopleSoft Virtual AutoPilot script in order to run it.

PeopleSoft Virtual AutoPilot classifies the following three types of events and identifies them as such in the script:

Event	Description
Functions	Includes JDB and CallObject APIs
Assignment statements	Refers to values typed in PeopleSoft AutoPilot
Conditional	Tests/branches (if/then statements)

PeopleSoft Virtual AutoPilot divides each event into parts and, in turn, identifies each of the parts based on an assigned format and a unique value. In short, the PeopleSoft Virtual AutoPilot script contains the details necessary for the Virtual Script Player to simulate running the PeopleSoft software kernel.

PeopleSoft Virtual AutoPilot identifies transaction boundaries, which you can set in the original script by designating a script command as the start of the transaction and another script command as the end of the transaction. Setting transaction boundaries can help you to analyze system performance when running a series of tasks.

Virtual Script Player

This section discusses:

- Virtual Script Player features.
- Virtual Script Player initialization file parameters.
- Virtual Script Player command line.
- Environment initialization.
- Modes of operation.
- Preprocessing of valid values list data.
- Valid values processing.
- Date formatting.
- Script failure.
- Virtual Script Player limitations.

Virtual Script Player Features

The Virtual Script Player uses the PeopleSoft Virtual AutoPilot script that you generate in the Virtual Script Editor to simulate the concurrent activities of one or more PeopleSoft software users. It bypasses the presentation layer of PeopleSoft software and reproduces the PeopleSoft application calls to the JDB and CallObject middleware. This reproduction is based on the timing and the sequencing of data in the event stream that you generate with PeopleSoft AutoPilot, manipulate in the Virtual Script Editor, and generate in modified form in the PeopleSoft Virtual AutoPilot script. In essence, the Virtual Script Player assumes the role of the PeopleSoft software runtime engine.

Virtual Script Player Initialization File Parameters

The `thevap.ini` file is a text file that contains the parameters that define the way that the Virtual Script Player runs. These parameters govern the paths that the Virtual Script Player follows to find files, synchronize playback timing, and set playback speed.

You can change the parameters, within established limits, to set the way the PeopleSoft Virtual AutoPilot scripts play.

Command

The Command section of the `thevap.ini` file contains the parameters that are necessary for interaction between Virtual Runner, which manages script playback, and the Virtual Script Player, which runs playback. These parameters specify the following:

- User ID and password
- Environment
- Script name
- Log file of summary playback statistics
- Location of the Virtual Script Player executable

The following table summarizes the [COMMAND] parameters and the meaning of each one:

Parameter	Meaning
UserID=	PeopleSoft Virtual AutoPilot user ID. Override on command line by entering -u and a user ID.
Password=	Password for PeopleSoft Virtual AutoPilot user. Override on command line by entering -p and a password.
Environment=	Environment for PeopleSoft Virtual AutoPilot script playback. Override on command line by entering -e and an environment.
Script=playscript.vsx	Name of PeopleSoft Virtual AutoPilot script (user can specify full path name for script here).

Parameter	Meaning
Common log=	Log file to which Virtual Script Player will write summary statistics for all playback sessions. Default folder is Vap_logs. Used only with Virtual Runner.
Binname=d:\b7\system\bin32\vapplayer.exe	Path by which Virtual Runner finds the Virtual Script Player executable.

Paths

The Path section of the vap.ini file identifies the directories for files that are needed by the Virtual Script Player. The contents of the needed files are:

- Log file, which gives detailed information about each PeopleSoft Virtual AutoPilot script playback session, the script name, and a line-by-line summary of each event in the script. The Virtual Script Player logs each event as it completes. The file also includes the start time and the date of the log.
- PeopleSoft Virtual AutoPilot script file, which stores all scripts that you might use for virtual playback.
- Valid values list file, which stores any valid values lists that the Virtual Script Player draws on for input values to run business functions. The Virtual Script Player uses valid values lists to get a new value each time it runs a business function.

The default file paths are as follows:

File/Contents	Parameter in vap.ini file	Default Path
Log of PeopleSoft Virtual AutoPilot playback events and messages	LogDirectory	c:\autopilot\VAP_LOGS
PeopleSoft Virtual AutoPilot scripts	VirtualScripts	c:\autopilot\VSX
Valid value lists	ValidValueLists	c:\autopilot\ATD

Timing

The Timing initialization parameters of the vap.ini file help you specify the terms under which PeopleSoft Virtual AutoPilot scripts play back:

- Rendezvous of multiple playback sessions, to control the amount of time the Virtual Script player delays a playback session following a rendezvous of multiple scripts running on a single workstation
- Synchronization of playback events, to set limits on the amount of time that threads can be inactive, events can occur behind the start time scheduled by the script, or that a thread has to wait for an API value or a handle parameter
- Playback speed, to adjust the amount of time between events to compensate for a fast or slow client workstation

The following table lists the PeopleSoft Virtual AutoPilot timing initialization parameters, their default values, what they govern, and the kind of timing factor to which they relate:

Parameter Name	Default Value	Meaning	Timing Factor
RandomDelayMax	0 seconds; can be set as high as 3,600	Enables user to set a maximum period that the Virtual Script Player will wait after the LoadRunner OWLogin rendezvous and environment initialization to begin each playback session. The default value means that following rendezvous, each player session proceeds without delay.	Rendezvous of multiple playback sessions
lMaxSleep	10,000 milliseconds	Establishes an upper limit on thread sleep time. Inactive threads must check on system status at least this often. If errors require the Virtual Script Player to shut down all threads, the parameter also determines the maximum amount of time required for the Player to shut down.	Playback synchronization
lTooLate	200 milliseconds; set higher in for debugging	The latest that any event can be run after the script schedules its start without causing virtual script playback to terminate.	Playback synchronization
lTimeout	60 seconds	Maximum number of seconds that an event has to run. If that number exceeds the parameter, Virtual Script Player terminates the playback session.	Playback synchronization
ClientSpeedFactor	100	Controls timing between script events by a constant factor. Decreasing the value of the parameter decreases the time between events.	Playback speed

Log

You use the Log section of the vap.ini file to specify the type of messages that the Virtual Script Player writes to a log file during a PeopleSoft Virtual AutoPilot script playback session. These messages can be important for debugging purposes. The following table summarizes the available log parameters and the debug message level that each one represents:

Log Parameter	Debug Message Level
31	Maximum log output; flush log file after each message (LoadRunner excluded)

Log Parameter	Debug Message Level
15	Parameter values and value substitutions
7	Error, warning, and status messages
3	Error and warning messages
1	Error messages only
0	Minimal messages

Note. You can cause the log file buffer to flush after every message by adding 16 to any parameter less than 31. However, you should not routinely do this, as flushing frequently increases file system overhead. For the same reason, you should not routinely set the log parameter at 31.

See Also

[Chapter 5, “Running Virtual Scripts,” Launching and Managing Multiple Script Playback, page 40](#)

[Chapter 6, “Understanding Special Considerations for Simulated Playback,” Playback Problem Simulation, page 43](#)

Virtual Script Player Command Line

You can launch the Virtual Script Player from LoadRunner, from Virtual Runner, or from the DOS command line. The command line must have entries that specify the user, the user’s password, the environment, and the script name with a default extension of .vsx for any PeopleSoft Virtual AutoPilot script, although this extension is not required.

The following four entries are required on the command line:

Command Line Abbreviation	Meaning	Sample Entry
-u	User	ce5791892
-p	PeopleSoft software user ID	-p pwd
-e	Environment	-e PDEV_VAP
-s	Script Name	-s voucherentry100.vsx

Environment Initialization

The Virtual Script Player does not immediately begin playing a PeopleSoft Virtual AutoPilot script upon launch from the DOS command line, from Virtual Runner, or from LoadRunner. In fact, the Virtual Script Player reads the script and runs events that generate a PeopleSoft software environment structure. The data that drives the generation of the environment comes from entries in the command line. For example, one user might create a PeopleSoft Virtual AutoPilot script, but another user might play the script. During initialization, the Virtual Script Player passes in the user ID of the user playing the script, thereby creating the proper environment. Therefore, you can run the Virtual Script Player in an environment different from the one in which you or someone else created the PeopleSoft Virtual AutoPilot script.

Environment initialization takes about 15 to 30 seconds. LoadRunner regards this passage as initializing time, while the DOS command line reads it as busy activity.

Modes of Operation

The Virtual Script Player automatically detects whether you have launched a PeopleSoft Virtual AutoPilot script from the DOS command line, from Virtual Runner, or from LoadRunner. If LoadRunner launches the script, the Virtual Script Player responds to stop/pause commands and sends transaction times and log output to LoadRunner. In addition, the Virtual Script Player completes a LoadRunner rendezvous just after it has initialized the system environment.

Preprocessing of Valid Values List Data

The preprocessing capability of the Virtual User Tool works with the Virtual Script Editor and Virtual Script Player to use valid values lists during script playback. You must mark valid value lists because virtual playback requires the values contained in these lists as parameters for API database calls.

When a PeopleSoft Virtual AutoPilot script specifies that a particular value originates in an PeopleSoft AutoPilot valid values list, the Virtual Script Player reads the valid values list file. All valid values lists are identified by the extension .atd. Before the Virtual Script Player plays the script, it performs preprocessing that includes looking up the database values in the valid values list and storing them until they are required as parameters for API calls. When the Virtual Script Player runs the PeopleSoft Virtual AutoPilot script, the stored list supplies the parameters needed for JDB or CallObject calls.

Preprocessing plays an important role in the PeopleSoft Virtual AutoPilot scheme because it takes care of the lookup and load of the valid values that the Virtual Script Player needs for PeopleSoft Virtual AutoPilot script execution. This ensures that the required values exist before playback. If the Virtual Script Player had to run database lookups at the time of script playback, the result would be artificial load on the database, which would, in turn, distort the simulation of activity that PeopleSoft Virtual AutoPilot seeks to achieve.

Valid Values List Processing

The Virtual Script Player defines the location of any valid value lists that are part of the PeopleSoft Virtual AutoPilot script in the `evap.ini` file. The Virtual Script Player reads valid value lists that are 64K or smaller into memory. If the file is larger than 64K, the Virtual Script Player must read it from the file. During virtual playback, if the Virtual Script Player reaches the end of a valid values list, it starts back at the beginning of the list, reuses the first value, and continues in sequence until virtual playback is complete.

Date Formatting

The Virtual Script Player expects a certain format for date strings for valid value lists and for literal typed-in values from PeopleSoft AutoPilot. Therefore, the Virtual Script Player supports different date formats that might appear in the PeopleSoft Virtual AutoPilot script, including mm/dd/yyyy and Julian date strings (that is, 102343 or 12/09/2002).

Important! The Virtual Script Editor correctly formats date entries for literal values but not for date entries in valid value lists.

Script Failure

Script failure might occur during the initialization process. For example, a branch event in the script might not refer to a valid event, or the events might not occur in the same thread. In the first example, the script fails before it is launched because the Virtual Script Player cannot validate the events. On initialization, the Virtual Script Player also validates function parameters. For example, a parameter such as Fetch might accept only 0 (zero) or 1 as values. If a different value is used, validation fails and, thus, the script fails before launching.

If the script fails during playback, the failure shuts down script processing. For most API calls, failure to return a success code causes the playback process to halt. The shutdown occurs without user intervention. LoadRunner, for example, returns a failure report, and the Virtual Script Player sends an error message to the log file, for example: `LoadRunner/Test Name/Local1/Subdirectory Name`. One subdirectory exists for every LoadRunner test session, which means that 50 simulated user test sessions produce 50 subdirectories.

If you launch the Virtual Script Player from a command line or from Virtual Runner and script failure occurs, no error message appears on the screen. You must open the log file that stores the test session results and examine the messages, a task you complete by searching on the keyword Error.

Virtual Script Player Limitations

The overriding consideration for PeopleSoft Virtual AutoPilot script playback is that client workstations must not impede the playback process. You must determine how many processes the workstation can realistically support, based on an analysis of workstation memory and CPU capability. Running either Task Manager or Performance Monitor can assess these capabilities.

Other Virtual Script Player limitations are hard-coded. If the Virtual Script Player gets a script that exceeds these limitations, you receive error messages that require a service pack to address. First, the Virtual Script Player supports up to 30 active user handles and 60 active request handles per session. Second, the Virtual Script Player can process only a certain number of status messages per second under LoadRunner. If the playback exceeds that number, some of the messages are lost, but the Virtual Script Player does not shut down.

VSMEditor

This section discusses:

- VSMEditor features.
- All Virtual Scripts list box.
- Master Scripts list box.
- VSM files.

VSM Editor Features

After you create a number of PeopleSoft Virtual AutoPilot scripts, the VSMEditor enables you to concatenate any number of those scripts into a single master script. Concatenating single scripts into a single master script is advantageous because you can run a series of unrelated tasks during testing.

You control the VSMEditor from the VSMEditor form, which you access by clicking the VSMEditor executable.

All Virtual Scripts List Box

The All Virtual Scripts list box contains all PeopleSoft Virtual AutoPilot script files that you have created; these files have a .vsx extension. In addition, any master scripts that you have created appear in this list box; master scripts have a .vsm extension. The location of any PeopleSoft Virtual AutoPilot script files that appear in the All Virtual Scripts list box is determined by the value of the VirtualScripts parameter of the PATHS section in the vap.ini.

You enter the path to the location of your virtual scripts to set the VirtualScripts parameter.

You can use any script in the All Virtual Scripts list box to create a master script. You create the script concatenation by choosing one of the scripts in the box and then holding down the Control key or the Shift key to select additional scripts.

You click the Add button to add the files that you chose to the Master Script list box.

Master Scripts List Box

The Master Scripts list box shows all the scripts that you have currently chosen for addition to a new .vsm (virtual script master) file. You can manipulate the script list in the Master Scripts list box by using the buttons adjacent to the box:

Remove	Deletes the chosen script from the Master Scripts list
Move Up and Move Down	Shift the position of the selected script in the list
Remove All	Deletes all scripts from the list
Save Master Scripts	Saves the list of scripts as a .vsm file

VSM Files

The VSMEditor creates a .vsm text file when you save a master script. You can change these files only through the VSMEditor because the file contains a checksum value that verifies the file's integrity. The PeopleSoft Virtual AutoPilot scripts always run in the sequence listed in the .vsm file. However, the first script to run is chosen randomly when the RandomStart parameter in the text file is set to 1.

The actual PeopleSoft Virtual AutoPilot scripts are not included in the .vsm master file. Therefore, you should not delete scripts from the folder that contains the .vsx files.

Virtual Runner

This section discusses:

- Virtual Runner features.
- Player session columns.
- Actions tools.

Virtual Runner Features

Virtual Runner controls the Virtual Script Player sessions on a single workstation and provides the following command and control functions for Virtual Script Player testing:

- Enables users to start one or more Virtual Script Player sessions on a single workstation
- Enables users to play multiple iterations of a single script
- Reports Player session status (pass/fail) to user
- Summarizes performance statistics over all Virtual Script Player sessions in a test

You use the action tools and the columns in the detail are of the Virtual Runner form to manage your Virtual Runner session.

Player Session Columns

After you finish setting up the parameters for the Virtual Script Player session, Virtual Runner displays the names of the scripts that you want to run. Initially, the status of the script is Down, indicating that you have not yet run it.

After you run a test, Virtual Runner changes the status to indicate success or failure.

Each column displays information about your Virtual Script Player sessions. The following descriptions summarize the purpose of each player session column:

- The State column indicates the current state of the player session. For example, after you successfully execute a player session, this column displays the word Success.
- The Env column indicates the specified environment for the current session. The environment is specified using the Options button or when you use the Virtual Runner Wizard.
- The User column displays the User name that you specified using the Options button or the Virtual Runner Wizard.
- The Repeat column specifies the number of times the script is repeated when you execute the player session. You specify this parameter when you use the Virtual Runner Wizard.
- The Script column specifies the path and file name of the script for the current player session. You specify these parameters when you use the Virtual Runner Wizard.

Actions Tools

You use the Actions tools to set up and launch a Virtual Runner session. You can select the scripts that you want to run as well as the number of script playback iterations. In addition, following playback you can access a log that contains pertinent information about the playback session.

The Virtual Runner toolbar contains the following six buttons:

Option	Enables you to specify the user ID, password, and environment for the virtual playback session.
Wizard	Directs you through the process of specifying all the Virtual Script Player session parameters, including the number of scripts to run and the script playback iterations.
Run	Runs the virtual script playback session.
Log	Displays the Log Viewer screen, which provides information about the last completed Virtual Script Player session.
Report	Prints a copy of the Virtual Script Player session log information.

Close

Closes the Virtual Runner window after you have decided whether to save the results of the Virtual Script Player session.

CHAPTER 4

Creating Virtual Scripts

This chapter provides an overview of script creation and discusses how to:

- Capture and import test results.
- Edit virtual scripts.

Understanding Script Creation

You use PeopleSoft Virtual AutoPilot to create a script that simulates the activities of PeopleSoft software as it handles the workload generated by many users. To achieve this goal, you use two key components of PeopleSoft automated testing tools architecture:

- PeopleSoft AutoPilot
- Virtual Script Editor

Using these two tools, you accomplish the following sequence of tasks to create a virtual script:

1. Create an PeopleSoft AutoPilot script.
2. Run the PeopleSoft AutoPilot script with playback configured so that you can capture system and PeopleSoft AutoPilot data.
3. Import the event stream into the Virtual Script Editor.
4. Create value links between source parameters of API calls and the target parameters of other API calls to ensure that usable data flows between API calls when you run the virtual script.
5. Add loops to the PeopleSoft Virtual AutoPilot script to account for repetitive processing, such as data retrieval.
6. Generate the PeopleSoft Virtual AutoPilot script, which the Virtual Script Player runs.

After you create a virtual script, the Virtual Script Player runs the script. You use Virtual Runner or LoadRunner to manage the number of sessions, either from a single workstation or from multiple workstations.

Capturing and Importing Test Results

This section provides an overview of test results and discusses how to:

- Capture test results.
- Import test results.

- View test results.

Understanding Test Results

PeopleSoft AutoPilot enables you to create scripts that test PeopleSoft software applications. When you create a script, you can configure PeopleSoft AutoPilot's playback function so that it captures and saves the results of your playback session, which it stores in the PeopleSoft AutoPilot Playback Results Detail Table (F97214) as an event stream.

You can view the playback results in a variety of ways. You can view the event stream alone, you can view details of individual events, or you can view timing information about groups of events and thread identifiers, displayed in a horizontal bar graph.

The data that PeopleSoft AutoPilot captures provides the raw material for your PeopleSoft Virtual AutoPilot script. After you capture PeopleSoft AutoPilot script data, you import it to the Virtual Script Editor so that you can prepare a virtual script.

Capturing Test Results

To gather the raw data for a virtual script, you must first write and run an PeopleSoft AutoPilot script and capture the results of the playback as an event stream. You use the Tools option in the menu bar of the PeopleSoft AutoPilot form to set up the capture mechanism.

To capture test results:

1. From your desktop or the appropriate directory, double-click the PeopleSoft AutoPilot executable. Create the PeopleSoft AutoPilot script or open an existing script.
2. From the File menu, select Open to open an PeopleSoft AutoPilot script.

Important! When you run the script, it must sign on to a PeopleSoft software environment. A script that does not include this signon does not function correctly in PeopleSoft Virtual AutoPilot because it does not contain the data required for the Virtual Script Player to initialize the environment.

3. From the Tools menu, select Options.
4. On the Options form, select the Playback tab.
5. Select the following options:
 - Save Results Data after Playback
 - Display Results Data after Playback
6. In the Event Stream Capture Level portion of the Playback tab, select Level 1 API calls.

Note. If you want to capture more script playback events, select the All API call levels option. Remember that you generate a much larger event stream if you select this option.

7. Click OK.
8. Save the PeopleSoft AutoPilot script.
9. In the PeopleSoft AutoPilot menu bar, click Play and select Play From Top.

PeopleSoft AutoPilot runs the script. The Play From Top command generates test results for DENPCX (where DENPCX=the name of the machine on which PeopleSoft AutoPilot resides). The PeopleSoft AutoPilot Results form displays detailed information about the playback session.

10. Click Close to close the Test Results window.
11. Click File/Exit to close PeopleSoft AutoPilot.

Importing Test Results

After you have run an PeopleSoft AutoPilot script and saved the playback results, you can import the event stream into the Virtual Script Editor. Importing the event stream enables you to use the Virtual Script Editor to forge value links between the source and target parameters of API calls; to identify, designate, and edit repetitive processing; and to generate a virtual script.

To import playback results into the Virtual Script Editor:

1. From your desktop or the appropriate directory, double-click the Virtual Script Editor executable.
2. On Virtual Script Editor, select a script to import.
3. Click the Import button.

After the Virtual Script Editor imports the script, an APEdit dialog box appears, confirming that the import was successful.

4. In the Virtual Script Editor dialog box, click OK.

Important! If you attempt to import an PeopleSoft AutoPilot script that you captured without a system signon, PeopleSoft Virtual AutoPilot displays a warning issue.

If a message appears, you should recapture the data, making sure that you sign on to the system through PeopleSoft AutoPilot. To do so, close PeopleSoft software, and then launch the PeopleSoft AutoPilot script. PeopleSoft AutoPilot handles your system signon.

Viewing Test Results

After you successfully import the results of a script playback, the event stream appears in the detail area of the Virtual Script Editor form.

Important! An exclamation point next to a start time (in the Start column) in a line of the event stream indicates that an error occurred during data capture.

If you find exclamation points in the event stream, you should investigate the possible causes for the error, and then edit and rerun the PeopleSoft AutoPilot script.

To view the event stream:

1. From your desktop or the appropriate directory, double-click the Virtual Script Editor executable.
2. In the toolbar of the Virtual Script Editor form, click one of the following options:
 - Details
 - Graph
 - Both

To view the event stream alone, click the Details option in the toolbar. To view categories of playback events, thread activity, or both represented in a horizontal bar graph by duration and time of occurrence, click the Graph option, and then click the scroll bar button in the form to select either View Graph by Message Type or View Graph by Thread ID. Click the Both option to view both the linear event stream and the horizontal bar graph representation.

Editing Virtual Scripts

This section provides an overview of virtual script generation and discusses how to:

- Use the Find feature.
- Add loops.
- Value-link parameters.
- Link values in inquiry scripts.
- Link values in entry scripts.
- Generate PeopleSoft Virtual AutoPilot scripts.
- Create master scripts.

Understanding Virtual Script Generation

After you import an event stream into the Virtual Script Editor, you can create a virtual script by completing the following two primary tasks:

- Adding loops
- Creating value links

After you finish these tasks, you generate the virtual script. The Virtual Script Editor passes the loop and value link information, as well as playback information that it stores automatically, to the Virtual Script Player, which runs the virtual script.

When you add loops, you define the number of data retrievals that PeopleSoft Virtual AutoPilot performs when you run the virtual script. You can limit the number of loops, or you can ensure that the Virtual Script Player loops until no more data is available.

When you create value links, you ensure that data necessary to run the virtual script flows dynamically between parameters in API calls. For example, you must value link APIs that use next numbers so that the Virtual Script Player retrieves the appropriate next number during virtual playback. If you fail to value link the next number parameter in this scenario, the Virtual Script Player passes the same value used in the original script to the API parameter that requires it, which causes a duplicate key error. When you forge a value link, the Virtual Script Editor stores the parameter value in a variable, which ensures that the value changes each time you run the script, preventing duplicate keys and data contention.

The PeopleSoft Virtual AutoPilot set also enables you to concatenate virtual scripts into a master script list using the VSMEditor. Using a master script enables you to test more than one script in a single virtual script playback session.

Using the Find Feature

You use the Find feature in the Virtual Script Editor to search for parameters that you will need to value link to create the PeopleSoft Virtual AutoPilot script.

To use the find feature:

1. Click inside the pane you want to search.
2. Type a value in the Find control.

To search for valid value list values to link, enter a list value to the Find control.

To find loops to process, search for JDB Fetch calls.

To find data dictionary aliases, enter a data dictionary alias, such as AN8.

3. Check the Case Sensitive button, if necessary.
4. Press Enter to run the search.

The Virtual Script Editor finds the first parameter with a data dictionary alias that matches your search criterion and marks it with an arrow.

Note. As you click a button to link or perform another task, you might lose the focus to the pane. Be sure to reset the focus to the pane you are searching, if necessary, by clicking inside the pane.

Adding Loops

Loops in PeopleSoft Virtual AutoPilot scripts simulate how PeopleSoft software functions when it performs inquiries. Without loops, the Virtual Script Player tries to fetch the same number of records that were retrieved during the original playback, regardless of selection criteria or available data. Loops also enable you to identify and reduce the number of events that must be value linked. Because of this advantage, you might want to generate loops before performing the value linking function.

To add a loop:

1. From your desktop or the appropriate directory, double-click the Virtual Script Editor executable.
2. Use the Find feature to search the event stream event pane for AUT: Press Button [Find].
3. From the Find statement in the event stream event pane, move the cursor down the list of events until you find a Fetch statement in the Message column.
4. Right-click and select Add Loop.

The Dynamic Loop Manager form appears.

5. On the Dynamic Loop Manager form, click one of the following options:
 - Loop until No Data
Click if you want PeopleSoft Virtual AutoPilot to exhaust data retrieval.
 - Loop X times
Specifies the number of loops you want the script to perform.
6. Click Apply Loop to add the loop.
7. To undo the loop, find the loop, launch the Dynamic Loop Manager form, and click the Undo Loop button.

Note. PeopleSoft Virtual AutoPilot colors events inside a loop light gray in the event stream, source, and target panes. You do not need to consider these events when you perform value linking.

Value-Linking Parameters

Value linking enables data to flow from function to function within PeopleSoft software. For a PeopleSoft Virtual AutoPilot script to accurately simulate system activities, it must not produce any duplicate key values in the system database. Therefore, for scripts that enter new data to the database or update existing data, at a minimum, you must value link all next number, job number, and batch number parameter values in the Virtual Script Editor. You can run simple inquiry scripts without any value linking, but these scripts might not accurately simulate system operations.

The Virtual Script Editor links some values automatically, but you must link others manually.

To run scripts accurately, you should always value link the parameters that:

- Pass the name of the machine on which you ran the original PeopleSoft AutoPilot script.
- Reference the date on which the original PeopleSoft AutoPilot script ran.
- Pass Next Numbers or serialized values (possibly labels of data items DOC, JOBS, MATH06, PYID, ICU).
- Use valid value list data. Linking these parameters ensures that the Virtual Script Player will use the .atd directory, where you store valid values list data as the source from which to retrieve data during virtual script playback.
- Contain the date that the script ran.

Note. You can use the Find feature to quickly find functions containing data to be linked. Click the column header to reorder the table (usually by label, value, or ID) to group like information.

To value link parameters:

1. From your desktop or the appropriate directory, double-click the Virtual Script Editor executable.
2. Import an PeopleSoft AutoPilot script into the Virtual Script Editor.
3. Click the Link Parameters button in the toolbar.

The Source Parameters pane and Target Parameters pane appear.

4. In the Target Parameters pane, select a target parameter line item.

The source parameters for that target display in the Source Parameters pane.

Note. Do not select the Show All option in the Source Parameters pane because doing so causes the Virtual Script Editor to display all the API calls in the pane.

5. To link a single parameter line item, select it and click the Link button.
6. To link all items in the script that match the source, target, label, and value parameters, select a representative parameter line item in the source pane and click the Link All button.

Note. Some parameters in the Target Parameters pane do not have a value from a source parameter. You can mark these as Literal using the Mark Literal Button. If you do not want to see the parameters that you have marked as Literal, click Link in the menu bar and select Filter Literals.

Linking Values in Inquiry Scripts

Because an inquiry does not change or update any data in the system, you are not required to forge value links between parameters in inquiry scripts. However, you should value link parameters that contain valid values list data to ensure that the data changes during playback of the virtual script.

If your script contains valid values data, you can run the virtual script, change the data, and run it again to extend your stress testing. You can change the data in the list without creating new value links. During virtual script playback, the Virtual Script Editor passes the new valid values list data to the Virtual Script Player for use in the appropriate parameters.

To link values in inquiry scripts:

1. From your desktop or the appropriate directory, double-click the Virtual Script Editor executable.

2. In the PeopleSoft Virtual AutoPilot Script Editor form, add any required loops to the PeopleSoft Virtual AutoPilot Script.
3. Find valid values list data in the event stream.
4. Link all source parameters containing valid values list data to the appropriate target parameters.
5. Document the data dictionary aliases that the Virtual Script Editor links.

Note. You find data dictionary aliases in the Label column of the Source Parameters pane and the Target Parameters pane.

Linking Values in Entry Scripts

Because entry scripts change or update system data, you are required to link values in entry scripts before you generate a virtual script. Value linking ensures that Virtual Script Player can pass values between parameters and that key parameter values change during virtual script playback, preventing record duplication.

To link values in entry scripts:

1. From your desktop or the appropriate directory, double-click the Virtual Script Editor executable.
2. In the Virtual Script Editor form, add any required loops to the PeopleSoft Virtual AutoPilot Script.
3. Find and link any parameters that pass the machine name on which the PeopleSoft AutoPilot script originally ran (these might be marked with CTID or MKEY data dictionary aliases or labels).
4. Find and link parameters that pass the date that the PeopleSoft AutoPilot script originally ran.
5. Find and link parameters that pass Next Numbers or serialized values (possibly data dictionary aliases of DOC, JOBS, MATH06, PYID, and ICU).
6. Find and link parameters that pass valid values list values.
7. Document the data dictionary aliases that the Virtual Script Editor links.

Generating PeopleSoft Virtual AutoPilot Scripts

To generate the PeopleSoft Virtual AutoPilot script:

1. From your desktop or the appropriate directory, double-click the Virtual Script Editor executable.
2. In the event stream event pane, click the Generate button.

When you press the Generate button, the Virtual Script Editor produces a virtual script, which the Virtual Script Player uses to simulate playback. A Script Log form appears following generation, summarizing the number of lines in the script and the number of errors, if any. You must generate an error-free script before you attempt to run it.

3. Assign a file name and location to which you want to save the generated script and click OK to begin script generation.

To generate the PeopleSoft Virtual AutoPilot script for playback using the Virtual Runner program, the script that you want to generate must be open in the script list pane.

After PeopleSoft Virtual AutoPilot generates the script, the Virtual Player Script Log form appears and displays information about the script generation, including the status (complete or incomplete), the number of lines, and the number of errors. If the Script Log form indicates that errors occurred during generation, you must investigate the error summaries that appear in the form and correct them by editing and rerunning the PeopleSoft AutoPilot script, and then repeating the steps for creating a virtual script.

4. Click OK to close the Virtual Player Script Log form.

Note. After you generate a virtual script, it is static. Any script changes that you make in PeopleSoft AutoPilot require re-editing and regeneration in Virtual User Tool. Thus, careful documentation of the editing process is critical to the production of repeatable results.

Creating Master Scripts

Using the VSMEditor tool, you can concatenate PeopleSoft Virtual AutoPilot scripts into a single master script. Concatenation gives you another testing option: you can run test series of unrelated scripts.

To create a master script:

1. From your desktop or the appropriate directory, double-click the VSMEditor executable.
2. In the All Virtual Scripts list box, select the script files that you want to include in the master script.
3. When you have chosen all the virtual script text files that you want, click the Add button.
VSMEditor adds the script files to the Master Script list box.
4. Manipulate the list in the Master Script list box by using the buttons adjacent to the box to remove script files or to change their order.
5. When you have decided on the content and order of the master script, click the Save Master Script button.

The VSMEditor saves the master script as a .vsm file. The file includes:

- Master script version
- Checksum value to verify file integrity
- RandomStart parameter (a value of 1 means that the first script to run is chosen randomly)
- List of PeopleSoft Virtual AutoPilot Script files

CHAPTER 5

Running Virtual Scripts

This chapter provides an overview of virtual script runs and discusses how to:

- Run virtual scripts from a single workstation.
- Launch and manage multiple script playback.

Understanding Virtual Script Runs

After you generate one or more PeopleSoft Virtual AutoPilot scripts, you are ready to execute playback to simulate multiple users running processes. If you want to simulate multiple users on a single workstation, you can launch the script either from a command line or from Virtual Runner.

Using Mercury Interactive's LoadRunner tool, you can also launch one or more PeopleSoft Virtual AutoPilot script playback sessions on more than one workstation. LoadRunner manages the playback sessions. Using LoadRunner as your script playback manager enables you to more accurately simulate the actual stress that users in a business environment might impose on the system.

Running Virtual Scripts from a Single Workstation

This section provides an overview of virtual script runs from a single workstation, lists a prerequisite, and discusses how to:

- Run virtual scripts from a command line.
- Run virtual scripts using Virtual Runner.

Understanding Virtual Script Runs from a Single Workstation

You can launch the Virtual Script Player from a command line on a single machine or from Virtual Runner, which manages virtual script playback, in order to simulate more than one user on a single workstation. The Virtual Script Player accesses the .vsx file that you create when you generate a virtual script on the Virtual Script Editor. After you run the script, you check the log files for errors.

Prerequisite

Before you can use Virtual Runner, you must cut the vap.ini file from \\B9\system\Bin32 and paste it into \\WINNT.

Running Virtual Scripts from the Command Line

The Virtual Script Player accesses the .vsx file generated by the Virtual Script Editor. You can launch the Virtual Script Player from a command line on a single machine or from a LoadRunner controller when you want to run virtual scripts on more than one workstation.

The command line must have entries specifying the user, the environment, and the script name. The following table summarizes the required entries on the command line:

Command Line Abbreviation	Description	Sample Entry
-u	User ID	-u JDE
-p	User password	-p JDE
-e	Environment	-e PRD733
-s	Script Name + number of script iterations to run	-s Script1.vsx

To run virtual scripts from a command line:

1. From the Start menu in Windows, select Command Prompt from the Programs menu.
2. At the C: prompt, type the Virtual Script Player command with appropriate parameters. For example:
Virtual Script Player -u JDE -p JDE -e PRD733 -s5 script1.vsx.
3. Press Enter to run the command.
4. To review the progress of the program, press Ctrl-Alt-Del to access the Windows Task Manager.

Note. The Processes tab displays the executable (Virtual Script Player.exe) and the CPU activity associated with it. Otherwise, there is no indication of activity on the screen.

When playback concludes, the Virtual Script Player.exe task disappears from the Task Manager window and a log in the \\PeopleSoft AutoPilot\VAP_Logs directory displays any errors that were encountered. You can change the directory location in the section of vap.ini file.

5. To search the PeopleSoft Virtual AutoPilot log for errors, click the Search menu, select Find, and search on the keyword *error*.

If errors occur, see the documentation on debugging virtual scripts.

6. If the script contains valid values list data, change the data and play the script again.

See Also

[Chapter 3, “Understanding PeopleSoft Virtual AutoPilot Components,” Virtual Script Player Initialization File Parameters, page 19](#)

Running Virtual Scripts Using Virtual Runner

The Virtual Runner program enables you to manage the playback of virtual scripts. You use it to specify the script, the number of player sessions, and the number of iterations that you want to run in each session. You also specify the system environment against which you want to run the sessions.

Z

You use the Virtual Runner log in conjunction with the system logs and more detailed VAP logs to help debug failed sessions.

To run virtual scripts using Virtual Runner:

1. From your desktop or the appropriate directory, double-click the Virtual Runner executable.
2. Click the Option button located on the Virtual Runner toolbar.
The Option window appears.
3. Complete the following fields and then click OK:
 - User ID
 - Password
 - EnvironmentThe environment against which you want to run the test.
4. Click the Wizard button.
The Virtual Runner Add Wizard - Step 1 of 3 form appears.
5. Click the Browse for script button to select the script that you want to run.
The Choose a Virtual Player Script to run form appears.
6. Select the script that you want and click the Open button.
The name of the script appears in the Select a script to run field.
7. Specify the number of player sessions to run on the workstation.
Example: enter 4 to run four scripts simultaneously.
8. Specify the number of virtual script session iterations to run.
Example: enter 5 to run the script five iterations sequentially.
9. Click the Next button.
The Virtual Runner Add Wizard - Step 2 of 3 form appears. If you entered information into the Option window, then the Wizard pulls that information into this window.
10. If you did not enter information into the Option window, enter your User ID, Password, and Environment.
11. Click Next.
The Virtual Runner Add Wizard - Step 3 of 3 form appears.
12. If you want to add another script, click the button to add more scripts and repeat steps 4 through 10.
13. If you do not want to add additional scripts, click the Finish button to return to the Virtual Runner form.
Virtual Runner displays the script or scripts that you chose and activates the Run button.
14. Click the Run button to begin script processing.
The main Virtual Runner screen displays the message *Starting Up*, indicating that the processing of the scripts has begun. The main Virtual Runner screen displays the message *Running* when Virtual Runner is processing the script or scripts. If the scripts successfully run, the screen displays the message *Success*.
After Virtual Runner finishes running the sessions, it displays the status of each test. You can view log information about a test by clicking the Log button. If processing is not successful, a red failure message appears.
15. Click Log and click Yes to view the current test log.

The Common Log Viewer form appears.

16. Review the log for error and warning messages.

You can expand the nodes in the Common Log Viewer form to see any error or warning messages that might have been issued during the Virtual Runner session. In addition to error and warning messages, the form displays:

- Name of the test
- Number of errors
- Number of warnings
- Status of the test
- Duration of the test
- Time of completion

17. If the script contains valid values list data, change the data and play the script again.

Launching and Managing Multiple Script Playback

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

- Define scripts.
- Define the host machine.
- Define virtual users.
- Gather LoadRunner results.
- Run virtual playback from the LoadRunner controller.

Understanding LoadRunner

LoadRunner enables you to set up multiple workstations, each representing multiple users, from which you can launch playback sessions to simulate actual user load on the system. You provide LoadRunner with selected rendezvous points and transactions, which LoadRunner then reports to its controller. LoadRunner gathers and stores the results of each run. The LoadRunner controller workstation must have network connection to all of the workstations that are involved in the test, and the controller must run Windows NT.

Defining Scripts

You define the script that you want to play back so that LoadRunner can locate the Virtual Script Player and pass the Player the necessary script command line.

Defining the Host Machine

After you have defined the scripts, you define the host machine for the LoadRunner test and the platform on which the test ran.

Defining Virtual Users

After you define the script and the host machine, you define the virtual users who created the scripts that you want to run. You can define users individually or you can define a group as the virtual user.

Setting Rendezvous Points

You set the rendezvous point that defines for LoadRunner the time at which all virtual scripts pause before the tool releases them for virtual playback.

Gathering LoadRunner Results

The LoadRunner results directory typically has the following structure:

- VAPI (the test directory)
- User Name (from those defined in the Users window)
- Session Number Output.txt (the rerouted VAP_log from the client workstation)

Running Virtual Playback from the LoadRunner Controller

After you have prepared virtual script playback, you are ready to run the test from the LoadRunner controller.

CHAPTER 6

Understanding Special Considerations for Simulated Playback

This section discusses:

- Playback problem simulation.
- Playback timing.
- Call level.
- Synchronous and asynchronous calls.
- Think times.

Playback Problem Simulation

The PeopleSoft Virtual AutoPilot solves several simulated playback problems. All of the problems in one way or another revolve around the tool's ability to simulate accurately the workings of the PeopleSoft run-time engine. The Virtual Script Editor and the Virtual Script Player work together; the Virtual Script Editor stores key playback information and passes it to the Virtual Script Player, which in turn uses the information to assume the role of the run-time engine. This section explains important simulated playback problems and the ways that PeopleSoft Virtual AutoPilot resolves them.

Playback Timing

This section discusses:

- Event synchronization.
- API playback timing.
- Interthread timing.

Event Synchronization

To accurately simulate system activities, PeopleSoft Virtual AutoPilot must keep script events synchronized during playback. This presents a challenge because PeopleSoft Virtual AutoPilot attempts to simulate multiple users who are stressing the server and the network, while the data upon which PeopleSoft Virtual AutoPilot scripts are based is captured from a single user's script playback. This means that event start times and duration might change significantly during a virtual script playback session.

To meet this challenge, PeopleSoft Virtual AutoPilot must solve two separate problems:

- Manage changes in the duration of individual API calls and the lengths of time between these calls within a single thread. This involves accurately simulating the process time required as the run-time engine handles user load.
- Handle timing differences that might affect interthread dependencies. These interdependencies occur when, for example, an API call in one thread has a data dependency on an API call in another thread.

API Playback Timing

The goal of PeopleSoft Virtual AutoPilot is to accurately simulate the stress that users place on the server and on the network. However, the PeopleSoft AutoPilot script, which contains the data upon which the PeopleSoft Virtual AutoPilot script is built, is not designed to create this stress. The time intervals between events in the PeopleSoft AutoPilot script reflect the running of a single script against the run-time engine. When you run a PeopleSoft Virtual AutoPilot script, the duration of events, and therefore the time intervals between those events, will likely change due to the server and network stress that the script is trying to simulate. The CPU power and memory capability of individual workstations can also affect the playback timing of PeopleSoft Virtual AutoPilot scripts.

PeopleSoft AutoPilot provides the base for the Virtual Script Player to time the execution of events during virtual script playback. When PeopleSoft AutoPilot processes a script, it captures each kernel function call, and it captures the start time and duration of each API call. Therefore, the script contains the gaps of time between each call, which occur as the system carries out other processes. The API calls within a thread might be represented as blocks of time of various lengths with intervening spatial gaps that symbolize the time duration between each call, as illustrated below:

Preserving these chronological gaps during data capture provides the basis for simulating playback by many users, a situation that is likely to increase the length of time that is required to execute the same API calls.

For example, suppose that virtual playback on a single workstation simulates 10 users using a server that is not as powerful as the server that was in use when the PeopleSoft AutoPilot script playback session originally occurred. In this scenario, the duration of API call is likely to lengthen, which could cause one API call to overlap another, halting playback.

However, since the event stream has preserved the *intervals* between each call, virtual playback can proceed, regardless of the duration of any or all of the calls within a thread.

PeopleSoft AutoPilot's ability to record the duration and length of time between API calls is also important because it can accurately determine the number of virtual users who can be simulated on a single workstation. For example, lengthy API calls might indicate an underpowered server, a workstation lacking the CPU and memory capability to handle the number of virtual user sessions that you desire, or an application bug. In each of these instances, you would likely scale back the number of users you want to simulate in a PeopleSoft Virtual AutoPilot playback session.

Interthread Timing

The Virtual Script Editor also plays a role in handling script playback timing so that PeopleSoft Virtual AutoPilot can simulate a stressed environment. The run-time engine might, for example, create a thread that contains an API call with a data dependency on another call, which might, in turn, exist in a separate, asynchronously running thread.

In a stressed environment, however, the duration of API calls might lengthen unpredictably. This might result in a data-dependent API call in one thread starting before the API upon which it depends has finished.

To deal with this potential problem, the Virtual Script Editor notes the data dependency when you forge value links between two API calls and preserves the timing interval between the calls.

When you run the virtual script, the Virtual Script Player increases the interval between APIs in one thread so that an API in one thread has time to complete before a data-dependent API in a thread running asynchronously to it is called.

In this way, PeopleSoft Virtual AutoPilot preserves the necessary time interval that existed between the data-dependent calls when you originally ran the script.

Clearly, the Virtual Script Player, in this scenario, manipulates the time interval between API calls in the first thread. However, the manipulation represents an attempt to fairly simulate what the system does in reality. The run-time engine manages data-dependent APIs so that they can run without breaking the system. It is, therefore, appropriate that the Virtual Script Player, in assuming the role of the run-time engine, simulate the run-time engine's responsibility—for example, the delay of one API's completion based on its logical relationship to another API.

Call Level

Some API calls invoke other API calls automatically within the same thread. Call level refers to an API call's position in the sequence of calls. For example, an EditLine business function might invoke a JDB Fetch call for a company number. In this example, the call level of the EditLine business function is 1, the call level of the JDB Fetch call is 2, and the PeopleSoft AutoPilot event stream records the two separate API calls.

However, while the run-time engine handles two separate API calls in this example, the processing occurs seamlessly: the second call follows immediately from the first without additional input from the user. For this reason, a PeopleSoft Virtual AutoPilot script contains only those API calls with a depth of 1. The Virtual Script Player automatically handles any API calls invoked by the original call, just as the run-time engine would.

This PeopleSoft Virtual AutoPilot capability is important for playing scripts back in batch mode. If APIs with a call level greater than 1 were treated separately, repetitive processing would occur. Such repetition would not correctly simulate system processing.

The Virtual Script Editor provides a convenient way for you to view call level in the event stream. Each line that displays an initiating API call shows a call level of 1. Any calls that are invoked by the initiating call show a level of 2 or greater.

Note. The Virtual Script Editor displays API calls with a call level greater than 1 only if you select the Capture Performance Statistics option when you configure playback in PeopleSoft AutoPilot. If you do not wish to view call levels greater than 1, select the Capture Virtual Script event stream option.

If you click the detail line of an API call that has a call level of 2 or greater, the event stream detail pane displays no parameters, meaning that you cannot value link any API call with a call level greater than 1. Therefore, no API calls with a call level greater than 1 appear in the Target Parameters pane.

Synchronous and Asynchronous Calls

As part of simulating PeopleSoft software operations, PeopleSoft Virtual AutoPilot must be able to manage synchronous and asynchronous API calls, an important management responsibility of the runtime engine. This ability ties into the Virtual Script Player's management of threads because an asynchronous call generates a separate thread.

A typical example of synchronous and asynchronous API call generation occurs when you enter data in a sales order line. You generate a synchronous call for each line edit; that is, the CallObject API for line 1 in a PeopleSoft grid precedes the CallObject API for line 2, and the CallObject API for line 2 does not occur until you have completed line 1. However, when you reach the end of a line, press the tab button, and proceed to line 2, you also generate an asynchronous API call that includes the data structure for the line that you just completed. The asynchronous CallObject API validates the data that you entered in line 1 through a series of related API calls. Meanwhile, you move ahead and begin entering sales order data in line 2.

The runtime engine manages this situation by generating a new thread for asynchronous calls and sending these calls to a queue to manage on a first-in, first-out (FIFO) basis. For example, you might enter 20 lines to the sales order entry grid. As you reach the end of each line and tab, the system will likely generate a new asynchronous call. Therefore, a number of asynchronous calls might queue for managing. When the runtime engine finishes processing the asynchronous calls, it stops the thread.

PeopleSoft Virtual AutoPilot manages the simulation of asynchronous call management through the operation of each part of its architecture. The PeopleSoft AutoPilot and PeopleSoft software hooks capture the timings of the synchronous and asynchronous calls that script playback generates. The Virtual Script Editor preserves the thread identifiers produced during playback, and the Virtual Script Player generates thread synchronization events in the PeopleSoft Virtual AutoPilot script based on the temporal relationships among events in the captured event stream.

The Virtual Script Player also manages the threads generated during virtual playback. When virtual playback yields an asynchronous call, the Virtual Script Player queues the calls in a new thread and manages them on the same FIFO terms that the runtime engine uses, thereby managing interthread synchronization as well as event timing within threads.

Synchronous and asynchronous call management provides another example of the Virtual User Tool's ability to accurately simulate the system, thereby providing you with a realistic picture of network and server stress.

See Also

[Chapter 6, "Understanding Special Considerations for Simulated Playback," Playback Problem Simulation, page 43](#)

Think Times

You insert wait periods while writing an PeopleSoft AutoPilot script in an attempt to accurately simulate the way people use PeopleSoft software. You click the Wait Before Proceeding button and insert pauses into the playback. These pauses are in millisecond increments.

One possible reason for inserting wait periods in the script is to simulate the pauses that might occur as a user enters vouchers. A user might pause to answer the phone or tend to other tasks, and then return to making the entries.

The event stream generated during PeopleSoft AutoPilot script playback records these wait times. They do not appear, however, with a label in the event stream pane if you import the event stream into the Virtual Script Editor. Rather, you recognize them by noting in the event stream pane the duration of time between the end of one event and the beginning of the next.

The PeopleSoft Virtual AutoPilot script that you create contains the waits inserted in the original script, and the Virtual Script Player manages the delays during script playback. The inclusion of think times provides another element that helps PeopleSoft Virtual AutoPilot simulate the PeopleSoft environment, which includes many users performing different tasks under a variety of circumstances.

You might want to analyze the event stream to see the length of time the events in the PeopleSoft AutoPilot script took to complete. Think times that you insert into the script do not interfere with event duration analysis because the Virtual Script Editor's event graph does not reflect any wait times. If, for example, a five-second wait occurs between a CallObject API and an OpenTable API, the event graph displays only the amount of time required to run the APIs. Thus, you get a true picture of the time that the system required to process the API calls.

CHAPTER 7

Troubleshooting PeopleSoft Virtual AutoPilot

This chapter provides an overview of PeopleSoft Virtual AutoPilot troubleshooting and discusses how to:

- Review the PeopleSoft Virtual AutoPilot log file to locate errors.
- Identify environment problems.
- Debug PeopleSoft Virtual AutoPilot scripts.

Understanding PeopleSoft Virtual AutoPilot Troubleshooting

You might encounter a script failure when you play back your PeopleSoft Virtual AutoPilot script. Troubleshooting PeopleSoft Virtual AutoPilot scripts consists of the following tasks.

1. Locate the source of the script failure, which might be in either PeopleSoft Virtual AutoPilot or the system.
2. Run through a short list of script debugging techniques.

These techniques correct errors in business function and database API calls, transaction timing, and multiple playback sessions. You might also need to debug the system. In some cases, the problem lies in the original PeopleSoft AutoPilot script or in application source code.

3. Review your PeopleSoft AutoPilot script if you created it without first validating it through replay.

You cannot trace all failures of PeopleSoft Virtual AutoPilot scripts to a single source, nor can you debug all scripts using a single method. In learning tips and techniques for troubleshooting PeopleSoft Virtual AutoPilot scripts, you also learn the best solution to apply to a particular problem.

Reviewing the PeopleSoft Virtual AutoPilot Log File to Locate Errors

This section provides an overview of the PeopleSoft Virtual AutoPilot log file and discusses how to:

- Find error entries in the PeopleSoft Virtual AutoPilot log file.
- Locate the log file in the event of early script failure.
- Set the message parameter.

Understanding the PeopleSoft Virtual AutoPilot Log File

The vap.log file contains messages about each PeopleSoft Virtual AutoPilot script that you run. Therefore, it is the primary source of information about errors that might cause your script to fail.

The Virtual Script Player sends an error message to the log file when a PeopleSoft Virtual AutoPilot script fails during processing. If you launched the script from LoadRunner or from Virtual Runner, script failure halts the playback process, sending an instant signal that an error has occurred. However, if you launched the script from the DOS command line, you will not receive an error message.

The message level parameter controls the kind and number of error messages that you receive in the vap.log file when you play back a PeopleSoft Virtual AutoPilot script. You set the message level in your PeopleSoft Virtual AutoPilot initialization file. You should generally set the message parameter at 0, 1, 3, or 7 to minimize the number of messages that you collect. Setting the parameter higher causes slower playback performance and at least potentially skew playback results, thereby making performance analysis difficult. However, when you are attempting to find the source of a script failure, increasing the message level parameter temporarily can help you diagnose the problem. You might find very few messages in the log file as a result of setting the debug parameter too low. For example, if you set the parameter to 0 (zero), you will receive only a minimal number of messages.

If you fail to find the source of the script failure in Virtual User Tool, you can use several procedures to troubleshoot the system.

See Also

Chapter 3, “Understanding PeopleSoft Virtual AutoPilot Components,” Virtual Script Player Initialization File Parameters, page 19

Finding Error Entries in the PeopleSoft Virtual AutoPilot Log File

To isolate an error, go to the log file, which contains the test results, select a test, open the text file, and search on the keyword *error*.

To find entries in the PeopleSoft Virtual AutoPilot log file:

1. Locate and open the vap.ini file.
2. In the vap.ini file, go to the [PATHS] section to determine the location of the LogDirectory.
3. Follow the path to the LogDirectory.
4. Open the text file for the failed script using Notepad.
5. From the Edit menu, select Find.
6. On the Find form, complete the *Find what* field with the word error.
7. Note the line and event in which the error occurred.
8. Click Find Next to go to the next error.

See Also

Chapter 3, “Understanding PeopleSoft Virtual AutoPilot Components,” Virtual Script Player Initialization File Parameters, page 19

Locating the Log File in the Event of Early Script Failure

The Virtual Script Player reads the location of the log directory out of the vap.ini text file. However, the script might fail before the Virtual Script Player has a chance to read the location. Therefore, when you go to the location of the log file that you specified as an initialization parameter, you will not find the test log. Despite the early failure, PeopleSoft Virtual AutoPilot did log the errors.

To locate the PeopleSoft Virtual AutoPilot log file in the event of early script failure:

1. Locate and open the vap.ini file.
2. In the vap.ini file, determine the location of the LogDirectory.
3. Follow the path to the LogDirectory.
4. If the log that you are looking for is not in its usual location, go to the root of the drive and look for the log.

Note. If you do not find a log file in either location, you must examine your PeopleSoft Virtual AutoPilot setup. Make sure that PeopleSoft Virtual AutoPilot is installed completely and correctly.

Setting the MessageLevel Parameter

To set the MessageLevel parameter:

1. Locate and open the vap.ini file.
2. In the vap.ini file, find the [Log] entry.
3. If the MessageLevel parameter is set lower than you want, change the setting.
4. Save your change and close the vap.ini file.

Note. If the Virtual Script Player crashes while you are running a script, you might find very few messages in the log file. This occurs because the Virtual Script Player did not flush the log file buffer, in which messages are stored, before the crash. You can prevent this by setting the message level parameter at 31. This parameter requires that the Virtual Script Player flush the log file buffer after each message. Remember, however, that system performance decreases when you set the message level at 31, so you should not leave it at that level permanently.

Identifying Environment Problems

This section provides an overview of environment problems and discusses how to:

- Diagnose environment problems.
- Investigate system errors.

Understanding Environment Problems

If your PeopleSoft Virtual AutoPilot script fails very early, even before the system completes its initial system logon, you might not be initializing the environment. In this case, you can troubleshoot system operations rather than PeopleSoft Virtual AutoPilot operations. For example, you can try to log on to Explorer and run it through several sample tasks, such as opening an application. Use the same user ID, password, and environment name when you log onto the system that you used when logging on to the Virtual Script Player. You also can troubleshoot system errors, as these might also prevent you from replaying your PeopleSoft Virtual AutoPilot script. If you have cleared any problems that might exist in running Explorer, try running your PeopleSoft Virtual AutoPilot script again.

Diagnosing Environment Problems

Because Virtual User Tool's primary task is to simulate system operations, it must be able to initialize an environment at script playback time. For this to happen, the system itself must be initializing correctly. To exclude the Virtual Script Player as the source of script failure, you might attempt to sign on to the system to make sure that it is opening and running correctly.

To diagnose an environment problem:

1. Close the Virtual Script Player.
2. Sign on to Explorer.
3. Perform several operations, such as accessing an application, changing forms, adding data, and so on.
4. If you are certain that the system is running correctly, rerun the script.

Note. Be sure to use the same user ID, password, and environment that you use when you log on to the Virtual Script Player.

Investigating System Errors

Even if the system is initializing correctly, you might find errors that occur when you attempt to enter or edit data in an application. To isolate errors that occur in the system, you can select debugging and attempt to correct the errors.

To investigate system errors:

1. Click the Windows Start menu and select Run.
2. In the Open control of the Run form, type JDE.INI.
The JDE.INI file appears.
3. In the JDE.INI file, go to the [DEBUG] section.
4. Enter the Output parameter as File.
5. Run an PeopleSoft AutoPilot script or access the system and run applications that are failing in the PeopleSoft Virtual AutoPilot script.

Note. If error messages display in the status bar, click the stop sign. Read the error messages that appear. You can right-click error messages to display more troubleshooting information about each one.

6. Open the jdedebug.log file to evaluate any errors that occur.

Important! Change the Output parameter in the [DEBUG] section of the JDE.INI file to NONE after you have corrected errors that prevent the PeopleSoft Virtual AutoPilot script from functioning correctly.

Debugging PeopleSoft Virtual AutoPilot Scripts

This section provides an overview of PeopleSoft Virtual AutoPilot debugging and discusses how to:

- Display business function parameters.
- Diagnose business function failures in PeopleSoft Explorer.
- Research value-linking errors in Virtual Script Editor.
- Verify that value linking is functioning.
- Identify and correct variable value-linking errors.
- Verify the validity of PeopleSoft Virtual AutoPilot script data.
- Identify and correct duplicate key errors.

- Rectify irregular transaction times.
- Prevent multiple playback errors.
- Correct uninitialized user handle errors.

Understanding PeopleSoft Virtual AutoPilot Debugging

If you have been troubleshooting problems with PeopleSoft Virtual AutoPilot script playback but are still having trouble running the scripts, a business function call is likely causing the failure. You can review the log file to locate the source of the error, and you can identify the particular business function call that failed. You should have the message level in the vap.ini file set at 15 so that the log file displays parameter values.

You might also encounter problems that complicate your performance characterization efforts. For example, transaction information that you incorrectly or incompletely enter in an PeopleSoft AutoPilot script might cause irregular transaction times in the PeopleSoft Virtual AutoPilot script, thus making it difficult to draw accurate conclusions about system performance. In this case, you should troubleshoot the PeopleSoft AutoPilot script, making sure that you have completely and accurately scripted input commands. If you modify the PeopleSoft AutoPilot script, remember to run it again, capture the playback results, and reimport the event stream into the Virtual Script Editor.

PeopleSoft Virtual AutoPilot also enables you to play back a script multiple times in succession, another important feature for performance characterization. However, doing so might cause playback to lock, again defeating your efforts to draw clearly and confidently characterize system performance. In this event, check your disk space to make sure you have enough to handle the testing.

If you have exhausted all of the debugging possibilities discussed here, you must turn your attention to debugging Explorer. Remember that if the same errors that appear in your PeopleSoft Virtual AutoPilot script appear when you run the application in PeopleSoft, you likely have a system problem that you must debug. Your debugging efforts might include a call to PeopleSoft System Support.

You can gain additional insight into potential system problems by double-clicking the stop sign that appears in the status bar of a PeopleSoft form when an error occurs. When you perform this action, the system displays explanatory text, including possible causes and solutions, that helps you diagnose the source of the error. You can get additional troubleshooting information by setting the Output parameter in the JDE.INI file to FILE. Remember that doing so will degrade system performance, so you should return the Output parameter in the JDE.INI file to NONE after you have diagnosed and corrected any problems with the script.

For a PeopleSoft Virtual AutoPilot script to run correctly, you must value link all required target parameters to the appropriate source parameters using the Virtual Script Editor. Failing to do so, or forging value links improperly, could cause your script to fail.

Displaying Business Function Parameters

Displaying the business function call parameters helps you to debug your PeopleSoft Virtual AutoPilot scripts. To do so, you set the MessageLevel parameter in the vap.ini file at 15 or at 31. At this level, the log file displays all the input and output parameter values of the following:

- Business function API calls in the script
- Text of any error messages
- File name of the business function
- Line number in the source code that contains the error

To display business function parameters:

1. Locate and open the vap.ini file.
2. In the vap.ini file, find the [Log] entry.
3. Set the Message Level parameter at 15 or 31.
4. Save your change and close the vap.ini file.

Important! Remember that you should not set the MessageLevel parameter permanently at 31 as this will cause performance to degrade. Leaving the MessageLevel parameter at 15 does not significantly degrade performance, but it can cause many messages and a great deal of text to accumulate in the log file. You should not leave the message level permanently set at 15, as doing so could consume a significant amount of disk space.

Diagnosing Business Function Failures in PeopleSoft Explorer

Your scripts must run properly in Explorer before PeopleSoft Virtual AutoPilot can run them properly. Therefore, you should determine early whether business function API calls are failing in the system when you run an PeopleSoft AutoPilot script. To do so, you can select PeopleSoft debugging in the JDE.INI file.

Note. When you run an application, right-click and select View System Log to view the jdedebug.log.

You might set breakpoints in the PeopleSoft AutoPilot script after commands that initialize a business function API call, which will enable you to check the jdedebug.log at these key points.

By verifying the system's ability to process the commands in the PeopleSoft AutoPilot script, you either pinpoint or exclude the system as a source of script failure. If it is causing the script failure, you work on debugging the system; conversely, if the business functions process properly when you run the script in the system, concentrate on finding the source of the script failure in the Virtual User Tool.

To diagnose business function failures in PeopleSoft:

1. Locate and open the JDE.INI file.
2. In the JDE.INI file, go to the [DEBUG] section and set the output parameter to FILE.
3. In the PeopleSoft AutoPilot script pane, right-click a command line that follows a command that runs a business function (optional).
4. Click Toggle Breakpoint (optional).
5. Play back the PeopleSoft AutoPilot script, either to the end or to a designated breakpoint.
6. Right-click inside a PeopleSoft form.
7. Select View System Log.
8. Click File.
9. In the drop-down menu, select c:\jdedebug.log.
10. Troubleshoot the jdedebug.log file, searching for business functions.

Researching Value-Linking Errors in the Virtual Script Editor

A business function API call might fail when you run your PeopleSoft Virtual AutoPilot script because you incompletely value linked the business function parameters in the event stream to the parameters in the PeopleSoft Virtual AutoPilot script while you were working in the Virtual Script Editor.

To research value linking errors in the Virtual Script Editor:

1. Locate and open the `vap.ini` file.
2. Set the `MessageLevel` parameter at 15.
3. Run the PeopleSoft Virtual AutoPilot script.
4. In the `vap.log` file, search for business function errors.
5. In the Virtual Script Editor, verify your value linking.

Remember that you must value link any parameters that do not use constant values during script playback. If you do not value link these parameters, the script fails because, typically, the script playback creates duplicate keys.

You are required to provide value links for the following parameters:

- Job number
- Document number
- Batch number
- Any parameter that uses a value from a valid values list

The following parameters might frequently require value linking:

- Computer identification
- Those that require dates

6. Perform any necessary value linking in the Virtual Script Editor.
7. Rerun the script with the `MessageLevel` parameter in the [LOG] section of your `vap.ini` file set at 15.
This setting enables you to capture parameter values and value substitutions in the log file.
8. Recheck the `vap.log` file and look for business function API errors.

Verifying That Value Linking Is Functioning

You can verify that PeopleSoft Virtual AutoPilot is linking parameter values by creating valid values lists in your PeopleSoft AutoPilot script. In the Virtual Script Editor, you value link any parameters that use values from the valid values list. The Virtual Script Player should link the values in the valid values lists to the appropriate parameters in the PeopleSoft Virtual AutoPilot script during PeopleSoft Virtual AutoPilot script playback.

To verify that PeopleSoft Virtual AutoPilot performs the value linking, you can set your `MessageLevel` parameter at 15 and run the PeopleSoft Virtual AutoPilot script. After you run the script, you search the log file for valid values list data, identify that data, and change the data in the `.atd` file, which stores your valid values list data.

To verify that value linking is functioning:

1. Locate and open the `vap.ini` file.
2. In the `vap.ini` file, set the `MessageLevel` parameter at 15.
3. Run the PeopleSoft Virtual AutoPilot script.
4. Review the log file for valid values list data.

Note. You can search for valid values list data using the `.atd` extension. Verify that the values you expect are present and look for any error messages associated with the data.

5. In the c:\atd file, change the valid values list data.
6. In the Virtual Script Editor, make sure that you have value-linked all of the new data in the valid values list to the correct parameters in the PeopleSoft Virtual AutoPilot script.
7. Rerun the PeopleSoft Virtual AutoPilot script.
When you replay the script, PeopleSoft Virtual AutoPilot should use the new data from the valid values list.
8. Review the log file for old valid values list data, to make sure that the Virtual Script Player used the new data rather than any of the old values.
9. If you find any of the old valid values list data, review the value linking in the Virtual Script Editor.

Identifying and Correcting Variable Value-Linking Errors

Another type of value linking related error occurs if you declare a value in a PeopleSoft AutoPilot script but do not set its value. In this case, if you value link the variable, the Virtual Script Editor registers errors in the script log during the virtual script generation process. To correct the errors, you must modify the PeopleSoft AutoPilot script by setting the value of the variable.

To identify and correct variable value-linking errors:

1. From your desktop or the appropriate directory, double-click the Virtual Script Editor executable.
2. Select a test and click the Generate button.
3. Review the Virtual Player Script Log form for validation error messages.
4. If validation error messages appear in the Virtual Script Log form, reopen the PeopleSoft AutoPilot script.
5. Set a value for any declared variables that do not have a value in the Virtual Script Editor.
6. Save and rerun the PeopleSoft AutoPilot script.

Important! Be sure that playback remains configured to capture the virtual script event stream.

7. Re-import the event stream into the Virtual Script Editor and regenerate the PeopleSoft Virtual AutoPilot script.

Verifying the Validity of PeopleSoft Virtual AutoPilot Script Data

Business function errors that occur in the PeopleSoft Virtual AutoPilot script might be caused by data errors. Data errors occur because the environment against which you wrote the PeopleSoft AutoPilot script differs from the environment against which you attempt to play back the PeopleSoft Virtual AutoPilot script.

To verify the validity of data in the PeopleSoft Virtual AutoPilot script:

1. Locate and open the vap.ini file.
2. In the vap.ini file, set the MessageLevel parameter to 15.
3. Run the PeopleSoft Virtual AutoPilot script.
4. Search the log file for business function errors.
5. Verify that the environment against which you wrote the PeopleSoft AutoPilot script and against which you ran the PeopleSoft Virtual AutoPilot script is the same.
6. If the two environments are different, recreate your valid values lists so that they contain values that are valid for the environment against which you are running the PeopleSoft Virtual AutoPilot script.

Note. You can also replay the PeopleSoft AutoPilot script in the same environment against which you are running the PeopleSoft Virtual AutoPilot script. In that case, follow the next two steps.

7. Re-import the event stream into the Virtual Script Editor.
8. Regenerate a PeopleSoft Virtual AutoPilot script by forging value links between the source and target parameters.

Identifying and Correcting Duplicate Key Errors

JDB Insert and Update API calls might fail in the PeopleSoft Virtual AutoPilot script because of duplicate key errors. These errors occur when you attempt to enter two records with the same value into a key column.

The duplicate key error prevents you from doing this. Failure to value link all the necessary parameters in the PeopleSoft Virtual AutoPilot script could cause duplicate key errors. You can view updated and inserted JDB API parameter values in the Virtual Script Editor.

Note. Duplicate keys could also result from an application error.

To identify and correct duplicate key errors:

1. Locate and open the JDE.INI file.
2. In the JDE.INI file, go to the [DEBUG] section.
3. Change the Output parameter to FILE.
4. Play the PeopleSoft AutoPilot script.
5. Locate and open the jdedebug.log file.

If you have duplicate key errors, you will find them in the jdedebug.log file.

6. Open the script in the Virtual Script Editor.
7. Check value linking for all JDB Insert and Update API calls.
8. When you are sure that you have value linked all JDB Insert and Update API calls, rerun the script.
9. If you continue to get duplicate key errors, review the application for errors that might be causing the problem.

Rectifying Irregular Transaction Times

You measure transaction times by choosing events as start and endpoints in your PeopleSoft AutoPilot script. For example, you might launch an application, move from one form to another by clicking the Add button, and then make entries to several header controls and grid columns in an active form before closing that form.

You might label that entire sequence of commands, from launching the application to closing the form, as a transaction. To see how efficiently the system manages this transaction, you label launching the application as the start of the transaction and closing the form as the end of the transaction. You also apply a name to the transaction and attach that name to the start and to the end. You use the Wait/Comment command in PeopleSoft AutoPilot to insert the start and end of the transaction into the script and to apply a name to the transaction.

If you do not include both a start and an end time for the transaction, you might find irregular or inexplicable transaction times in the log, or you might find that the transaction fails. Failing to ensure that the name that you applied to the start of the transaction matches precisely the name that you applied to the end of the transaction, including capital letters and any special characters, might also cause irregular transaction times or transaction failures.

Important! PeopleSoft Virtual AutoPilot transaction timing accuracy has several limitations that make broad-based performance characterization assertions impossible. Accurate timings can be achieved only on a discrete workstation, while PeopleSoft Virtual AutoPilot simulates server load.

To rectify irregular transaction times:

1. From your desktop or the appropriate directory, double-click the PeopleSoft AutoPilot executable.
2. In the script pane of the PeopleSoft AutoPilot form, place the insertion cursor directly above the command line that represents the start of the transaction.
3. In the menu bar, click Command and then select Wait/Comment.
4. In the uncompleted Comment list of the PeopleSoft AutoPilot command pane, enter Start, a space, and a name for the transaction.

5. Click the Insert button.

PeopleSoft AutoPilot inserts a command line marking the start of the transaction.

6. Place the insertion cursor after the command line that represents the end of the transaction.
7. In the PeopleSoft AutoPilot menu bar, click Command and then select Wait/Comment.
8. In the Comment list of the PeopleSoft AutoPilot command pane, enter End, a space, and a name for the transaction.

PeopleSoft AutoPilot inserts a command line marking the end of the transaction.

Important! 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.

9. Click the Insert button.

Preventing Multiple Script Playback Problems

PeopleSoft Virtual AutoPilot enables you to play back the same script in consecutive sessions or to simulate multiple users playing back scripts simultaneously. In either case, you must make sure that you have sufficient disk space to handle the load created by PeopleSoft Virtual AutoPilot script playback, particularly if you plan to run a long test involving many playback iterations or simulation of a large number of users. If you do not have sufficient disk space, you might find that PeopleSoft Virtual AutoPilot script playback locks up after only a few playbacks.

Debugging Virtual Runner

If Virtual Runner fails immediately after you click the Run button, first check the vapplayer.exe path specified in the vap.ini file. The vap.ini [COMMAND] section binname parameter specifies the full path of the VAPPlayer.exe file.

Virtual Script Player should operate the same whether you run under Virtual Runner control or from a command line. If it does not, you might be running two different copies of the vapplayer.exe. This might occur if the vap.ini [COMMAND] binname parameter is pointing to an old version of the Virtual Script Player. Make sure that binname parameter points to the correct drive and directory, and that you discard any old versions of the Virtual Script Player that you might have on your workstation.

If you set Virtual Script Player to run a virtual script multiple times in succession, and the script only runs a few times before locking up, you should review the available disk space. If you have set the JDE.INI error logging settings at a high level, the jde.log and jddebug.log can fill a disk very quickly. Make sure that enough free space is available on all relevant disk drives before you start a long test.

Debugging LoadRunner

If you have set the JDE.INI or vap.ini error logging settings at a high level, and you run many virtual user sessions, the network might become saturated, communications between LoadRunner controller and the host machines might become scrambled, or both. You can address this problem by setting the MessageLevel parameter in the vap.ini files on all machines lower. This will decrease the volume of log file traffic.

The following table summarizes steps that you can take to minimize PeopleSoft Virtual AutoPilot script playback problems:

Situation Affecting Playback	Possible Solution
jde.log and jddeb.log messages fill up disk quickly during PeopleSoft Virtual AutoPilot script playback	In [DEBUG] section of the JDE.INI file, set Output parameter to NONE
PeopleSoft Virtual AutoPilot log file fills with messages, consuming disk space	In [LOG] section of the vap.ini file, set MessageLevel parameter to 0, 1, 3, or 7

Correcting Uninitialized User Handle Errors

An error labeled *Uninitialized User Handle* might cause your PeopleSoft Virtual AutoPilot script to fail. This error occurs when you attempt to create a PeopleSoft Virtual AutoPilot script using playback results that you obtained from the first run of a PeopleSoft application when just-in-time installation occurs, or when you have system debugging turned on when you capture the results of PeopleSoft AutoPilot script playback.

To correct uninitialized user handle errors:

1. From your desktop or the appropriate directory, double-click the Virtual User Tool executable.
2. In Virtual User Tool, discard the results of the script generation attempt that failed.
3. In PeopleSoft AutoPilot, rerun the script in the same environment that you created it.
4. Use the new results data to generate a new PeopleSoft Virtual AutoPilot script.

Glossary of PeopleSoft Terms

absence entitlement	This element defines rules for granting paid time off for valid absences, such as sick time, vacation, and maternity leave. An absence entitlement element defines the entitlement amount, frequency, and entitlement period.
absence take	This element defines the conditions that must be met before a payee is entitled to take paid time off.
academic career	In PeopleSoft Enterprise Campus Solutions, all course work that a student undertakes at an academic institution and that is grouped in a single student record. For example, a university that has an undergraduate school, a graduate school, and various professional schools might define several academic careers—an undergraduate career, a graduate career, and separate careers for each professional school (law school, medical school, dental school, and so on).
academic institution	In PeopleSoft Enterprise Campus Solutions, an entity (such as a university or college) that is independent of other similar entities and that has its own set of rules and business processes.
academic organization	In PeopleSoft Enterprise Campus Solutions, an entity that is part of the administrative structure within an academic institution. At the lowest level, an academic organization might be an academic department. At the highest level, an academic organization can represent a division.
academic plan	In PeopleSoft Enterprise Campus Solutions, an area of study—such as a major, minor, or specialization—that exists within an academic program or academic career.
academic program	In PeopleSoft Enterprise Campus Solutions, the entity to which a student applies and is admitted and from which the student graduates.
accounting class	In PeopleSoft Enterprise Performance Management, the accounting class defines how a resource is treated for generally accepted accounting practices. The Inventory class indicates whether a resource becomes part of a balance sheet account, such as inventory or fixed assets, while the Non-inventory class indicates that the resource is treated as an expense of the period during which it occurs.
accounting date	The accounting date indicates when a transaction is recognized, as opposed to the date the transaction actually occurred. The accounting date and transaction date can be the same. The accounting date determines the period in the general ledger to which the transaction is to be posted. You can only select an accounting date that falls within an open period in the ledger to which you are posting. The accounting date for an item is normally the invoice date.
accounting split	The accounting split method indicates how expenses are allocated or divided among one or more sets of accounting ChartFields.
accumulator	You use an accumulator to store cumulative values of defined items as they are processed. You can accumulate a single value over time or multiple values over time. For example, an accumulator could consist of all voluntary deductions, or all company deductions, enabling you to accumulate amounts. It allows total flexibility for time periods and values accumulated.
action reason	The reason an employee's job or employment information is updated. The action reason is entered in two parts: a personnel action, such as a promotion, termination, or change from one pay group to another—and a reason for that action. Action reasons are used by PeopleSoft Human Resources, PeopleSoft Benefits Administration,

	PeopleSoft Stock Administration, and the COBRA Administration feature of the Base Benefits business process.
action template	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.
activity	<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>
address usage	In PeopleSoft Enterprise Campus Solutions, a grouping of address types defining the order in which the address types are used. For example, you might define an address usage code to process addresses in the following order: billing address, dormitory address, home address, and then work address.
adjustment calendar	In PeopleSoft Enterprise Campus Solutions, the adjustment calendar controls how a particular charge is adjusted on a student's account when the student drops classes or withdraws from a term. The charge adjustment is based on how much time has elapsed from a predetermined date, and it is determined as a percentage of the original charge amount.
administrative function	In PeopleSoft Enterprise Campus Solutions, a particular functional area that processes checklists, communication, and comments. The administrative function identifies which variable data is added to a person's checklist or communication record when a specific checklist code, communication category, or comment is assigned to the student. This key data enables you to trace that checklist, communication, or comment back to a specific processing event in a functional area.
admit type	In PeopleSoft Enterprise Campus Solutions, a designation used to distinguish first-year applications from transfer applications.
agreement	In PeopleSoft eSettlements, provides a way to group and specify processing options, such as payment terms, pay from a bank, and notifications by a buyer and supplier location combination.
allocation rule	In PeopleSoft Enterprise Incentive Management, an expression within compensation plans that enables the system to assign transactions to nodes and participants. During transaction allocation, the allocation engine traverses the compensation structure from the current node to the root node, checking each node for plans that contain allocation rules.
alternate account	A feature in PeopleSoft General Ledger that enables you to create a statutory chart of accounts and enter statutory account transactions at the detail transaction level, as required for recording and reporting by some national governments.
analysis database	In PeopleSoft Enterprise Campus Solutions, database tables that store large amounts of student information that may not appear in standard report formats. The analysis database tables contain keys for all objects in a report that an application program can use to reference other student-record objects that are not contained in the printed report. For instance, the analysis database contains data on courses that are considered for satisfying a requirement but that are rejected. It also contains information on

	courses captured by global limits. An analysis database is used in PeopleSoft Enterprise Academic Advisement.
AR specialist	Abbreviation for <i>receivables specialist</i> . In PeopleSoft Receivables, an individual in who tracks and resolves deductions and disputed items.
arbitration plan	In PeopleSoft Enterprise Pricer, defines how price rules are to be applied to the base price when the transaction is priced.
assessment rule	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.
asset class	An asset group used for reporting purposes. It can be used in conjunction with the asset category to refine asset classification.
attribute/value pair	In PeopleSoft Directory Interface, relates the data that makes up an entry in the directory information tree.
audience	In PeopleSoft Enterprise Campus Solutions, a segment of the database that relates to an initiative, or a membership organization that is based on constituent attributes rather than a dues-paying structure. Examples of audiences include the Class of '65 and Undergraduate Arts & Sciences.
authentication server	A server that is set up to verify users of the system.
base time period	In PeopleSoft Business Planning, the lowest level time period in a calendar.
benchmark job	In PeopleSoft Workforce Analytics, a benchmark job is a job code for which there is corresponding salary survey data from published, third-party sources.
billing career	In PeopleSoft Enterprise Campus Solutions, the one career under which other careers are grouped for billing purposes if a student is active simultaneously in multiple careers.
bio bit or bio brief	In PeopleSoft Enterprise Campus Solutions, a report that summarizes information stored in the system about a particular constituent. You can generate standard or specialized reports.
book	In PeopleSoft Asset Management, used for storing financial and tax information, such as costs, depreciation attributes, and retirement information on assets.
branch	A tree node that rolls up to nodes above it in the hierarchy, as defined in PeopleSoft Tree Manager.
budgetary account only	An account used by the system only and not by users; this type of account does not accept transactions. You can only budget with this account. Formerly called "system-maintained account."
budget check	In commitment control, the processing of source transactions against control budget ledgers, to see if they pass, fail, or pass with a warning.
budget control	In commitment control, budget control ensures that commitments and expenditures don't exceed budgets. It enables you to track transactions against corresponding budgets and terminate a document's cycle if the defined budget conditions are not met. For example, you can prevent a purchase order from being dispatched to a vendor if there are insufficient funds in the related budget to support it.
budget period	The interval of time (such as 12 months or 4 quarters) into which a period is divided for budgetary and reporting purposes. The ChartField allows maximum flexibility to define operational accounting time periods without restriction to only one calendar.

business event	<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>
business unit	A corporation or a subset of a corporation that is independent with regard to one or more operational or accounting functions.
buyer	In PeopleSoft 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.
campus	In PeopleSoft Enterprise Campus Solutions, an entity that is usually associated with a distinct physical administrative unit, that belongs to a single academic institution, that uses a unique course catalog, and that produces a common transcript for students within the same academic career.
catalog item	In PeopleSoft Enterprise Learning Management, a specific topic that a learner can study and have tracked. For example, "Introduction to Microsoft Word." A catalog item contains general information about the topic and includes a course code, description, categorization, keywords, and delivery methods. A catalog item can have one or more learning activities.
catalog map	In PeopleSoft Catalog Management, translates values from the catalog source data to the format of the company's catalog.
catalog partner	In PeopleSoft Catalog Management, shares responsibility with the enterprise catalog manager for maintaining catalog content.
categorization	Associates partner offerings with catalog offerings and groups them into enterprise catalog categories.
category	In PeopleSoft Enterprise Campus Solutions, a broad grouping to which specific comments or communications (contexts) are assigned. Category codes are also linked to 3C access groups so that you can assign data-entry or view-only privileges across functions.
channel	In PeopleSoft MultiChannel Framework, email, chat, voice (computer telephone integration [CTI]), or a generic event.
ChartField	A field that stores a chart of accounts, resources, and so on, depending on the PeopleSoft application. ChartField values represent individual account numbers, department codes, and so forth.
ChartField balancing	You can require specific ChartFields to match up (balance) on the debit and the credit side of a transaction.
ChartField combination edit	The process of editing journal lines for valid ChartField combinations based on user-defined rules.
ChartKey	One or more fields that uniquely identify each row in a table. Some tables contain only one field as the key, while others require a combination.
checkbook	In PeopleSoft Promotions Management, enables you to view financial data (such as planned, incurred, and actual amounts) that is related to funds and trade promotions.
checklist code	In PeopleSoft Enterprise Campus Solutions, a code that represents a list of planned or completed action items that can be assigned to a staff member, volunteer, or unit. Checklists enable you to view all action assignments on one page.

class	In PeopleSoft Enterprise Campus Solutions, a specific offering of a course component within an academic term. See also <i>course</i> .
Class ChartField	A ChartField value that identifies a unique appropriation budget key when you combine it with a fund, department ID, and program code, as well as a budget period. Formerly called <i>sub-classification</i> .
clearance	In PeopleSoft Enterprise Campus Solutions, the period of time during which a constituent in PeopleSoft Contributor Relations is approved for involvement in an initiative or an action. Clearances are used to prevent development officers from making multiple requests to a constituent during the same time period.
clone	In PeopleCode, to make a unique copy. In contrast, to <i>copy</i> may mean making a new reference to an object, so if the underlying object is changed, both the copy and the original change.
cohort	In PeopleSoft Enterprise Campus Solutions, the highest level of the three-level classification structure that you define for enrollment management. You can define a cohort level, link it to other levels, and set enrollment target numbers for it. See also <i>population</i> and <i>division</i> .
collection	To make a set of documents available for searching in Verity, you must first create at least one collection. A collection is set of directories and files that allow search application users to use the Verity search engine to quickly find and display source documents that match search criteria. A collection is a set of statistics and pointers to the source documents, stored in a proprietary format on a file server. Because a collection can only store information for a single location, PeopleSoft maintains a set of collections (one per language code) for each search index object.
collection rule	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.
comm key	See <i>communication key</i> .
communication key	In PeopleSoft Enterprise Campus Solutions, a single code for entering a combination of communication category, communication context, communication method, communication direction, and standard letter code. Communication keys (also called <i>comm keys</i> or <i>speed keys</i>) can be created for background processes as well as for specific users.
compensation object	In PeopleSoft Enterprise Incentive Management, a node within a compensation structure. Compensation objects are the building blocks that make up a compensation structure's hierarchical representation.
compensation structure	In PeopleSoft Enterprise Incentive Management, a hierarchical relationship of compensation objects that represents the compensation-related relationship between the objects.
condition	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.
configuration parameter catalog	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.
configuration plan	In PeopleSoft Enterprise Incentive Management, configuration plans hold allocation information for common variables (not incentive rules) and are attached to a node without a participant. Configuration plans are not processed by transactions.

constituents	In PeopleSoft Enterprise Campus Solutions, friends, alumni, organizations, foundations, or other entities affiliated with the institution, and about which the institution maintains information. The constituent types delivered with PeopleSoft Enterprise Contributor Relations Solutions are based on those defined by the Council for the Advancement and Support of Education (CASE).
content reference	Content references are pointers to content registered in the portal registry. These are typically either URLs or iScripts. Content references fall into three categories: target content, templates, and template pagelets.
context	<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>
control table	Stores information that controls the processing of an application. This type of processing might be consistent throughout an organization, or it might be used only by portions of the organization for more limited sharing of data.
cost profile	A combination of a receipt cost method, a cost flow, and a deplete cost method. A profile is associated with a cost book and determines how items in that book are valued, as well as how the material movement of the item is valued for the book.
cost row	A cost transaction and amount for a set of ChartFields.
course	<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>
course share set	In PeopleSoft Enterprise Campus Solutions, a tag that defines a set of requirement groups that can share courses. Course share sets are used in PeopleSoft Enterprise Academic Advisement.
current learning	In PeopleSoft Enterprise Learning Management, a self-service repository for all of a learner's in-progress learning activities and programs.
data acquisition	In PeopleSoft Enterprise Incentive Management, the process during which raw business transactions are acquired from external source systems and fed into the operational data store (ODS).
data elements	<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>
dataset	A data grouping that enables role-based filtering and distribution of data. You can limit the range and quantity of data that is displayed for a user by associating dataset rules with user roles. The result of dataset rules is a set of data that is appropriate for the user's roles.
delivery method	In PeopleSoft Enterprise Learning Management, identifies the primary type of delivery method in which a particular learning activity is offered. Also provides

	<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>
delivery method type	In PeopleSoft Enterprise Learning Management, identifies how learning activities can be delivered—for example, through online learning, classroom instruction, seminars, books, and so forth—in an organization. The type determines whether the delivery method includes scheduled components.
directory information tree	In PeopleSoft Directory Interface, the representation of a directory's hierarchical structure.
division	<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>
document sequencing	A flexible method that sequentially numbers the financial transactions (for example, bills, purchase orders, invoices, and payments) in the system for statutory reporting and for tracking commercial transaction activity.
dynamic detail tree	A tree that takes its detail values—dynamic details—directly from a table in the database, rather than from a range of values that are entered by the user.
edit table	A table in the database that has its own record definition, such as the Department table. As fields are entered into a PeopleSoft application, they can be validated against an edit table to ensure data integrity throughout the system.
effective date	A method of dating information in PeopleSoft applications. You can predate information to add historical data to your system, or postdate information in order to enter it before it actually goes into effect. By using effective dates, you don't delete values; you enter a new value with a current effective date.
EIM ledger	Abbreviation for <i>Enterprise Incentive Management ledger</i> . In PeopleSoft Enterprise Incentive Management, an object to handle incremental result gathering within the scope of a participant. The ledger captures a result set with all of the appropriate traces to the data origin and to the processing steps of which it is a result.
elimination set	In PeopleSoft General Ledger, a related group of intercompany accounts that is processed during consolidations.
entry event	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.
equitization	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.
equity item limit	In PeopleSoft Enterprise Campus Solutions, the amounts of funds set by the institution to be awarded with discretionary or gift funds. The limit could be reduced by amounts equal to such things as expected family contribution (EFC) or parent contribution. Students are packaged by Equity Item Type Groups and Related Equity Item Types. This limit can be used to assure that similar student populations are packaged equally.

event	<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>
event propagation process	<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>
exception	<p>In PeopleSoft Receivables, an item that either is a deduction or is in dispute.</p>
exclusive pricing	<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>
fact	<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>
financial aid term	<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>
forecast item	<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>
fund	<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>
gap	<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>
generic process type	<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>
gift table	<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>
GL business unit	<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>
GL entry template	<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.

GL Interface process	Abbreviation for <i>General Ledger Interface process</i> . In PeopleSoft Enterprise Campus Solutions, a process that is used to send transactions from PeopleSoft Enterprise Student Financials to the general ledger. Item types are mapped to specific general ledger accounts, enabling transactions to move to the general ledger when the GL Interface process is run.
group	<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>
incentive object	In PeopleSoft Enterprise Incentive Management, the incentive-related objects that define and support the PeopleSoft Enterprise Incentive Management calculation process and results, such as plan templates, plans, results data, user interaction objects, and so on.
incentive rule	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.
incur	In PeopleSoft Promotions Management, to become liable for a promotional payment. In other words, you owe that amount to a customer for promotional activities.
initiative	In PeopleSoft Enterprise Campus Solutions, the basis from which all advancement plans are executed. It is an organized effort targeting a specific constituency, and it can occur over a specified period of time with specific purposes and goals. An initiative can be a campaign, an event, an organized volunteer effort, a membership drive, or any other type of effort defined by the institution. Initiatives can be multipart, and they can be related to other initiatives. This enables you to track individual parts of an initiative, as well as entire initiatives.
inquiry access	<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>
institution	In PeopleSoft Enterprise Campus Solutions, an entity (such as a university or college) that is independent of other similar entities and that has its own set of rules and business processes.
item	<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>
item shuffle	In PeopleSoft Enterprise Campus Solutions, a process that enables you to change a payment allocation without having to reverse the payment.

joint communication	In PeopleSoft Enterprise Campus Solutions, one letter that is addressed jointly to two people. For example, a letter might be addressed to both Mr. Sudhir Awat and Ms. Samantha Mortelli. A relationship must be established between the two individuals in the database, and at least one of the individuals must have an ID in the database.
keyword	In PeopleSoft Enterprise Campus Solutions, a term that you link to particular elements within PeopleSoft Student Financials, Financial Aid, and Contributor Relations. You can use keywords as search criteria that enable you to locate specific records in a search dialog box.
KPI	An abbreviation for <i>key performance indicator</i> . A high-level measurement of how well an organization is doing in achieving critical success factors. This defines the data value or calculation upon which an assessment is determined.
LDIF file	Abbreviation for <i>Lightweight Directory Access Protocol (LDAP) Data Interchange Format file</i> . Contains discrepancies between PeopleSoft data and directory data.
learner group	In PeopleSoft Enterprise Learning Management, a group of learners who are linked to the same learning environment. Members of the learner group can share the same attributes, such as the same department or job code. Learner groups are used to control access to and enrollment in learning activities and programs. They are also used to perform group enrollments and mass enrollments in the back office.
learning components	In PeopleSoft Enterprise Learning Management, the foundational building blocks of learning activities. PeopleSoft Enterprise Learning Management supports six basic types of learning components: web-based, session, webcast, test, survey, and assignment. One or more of these learning component types compose a single learning activity.
learning environment	In PeopleSoft Enterprise Learning Management, identifies a set of categories and catalog items that can be made available to learner groups. Also defines the default values that are assigned to the learning activities and programs that are created within a particular learning environment. Learning environments provide a way to partition the catalog so that learners see only those items that are relevant to them.
learning history	In PeopleSoft Enterprise Learning Management, a self-service repository for all of a learner's completed learning activities and programs.
ledger mapping	You use ledger mapping to relate expense data from general ledger accounts to resource objects. Multiple ledger line items can be mapped to one or more resource IDs. You can also use ledger mapping to map dollar amounts (referred to as <i>rates</i>) to business units. You can map the amounts in two different ways: an actual amount that represents actual costs of the accounting period, or a budgeted amount that can be used to calculate the capacity rates as well as budgeted model results. In PeopleSoft Enterprise Warehouse, you can map general ledger accounts to the EW Ledger table.
library section	In PeopleSoft Enterprise Incentive Management, a section that is defined in a plan (or template) and that is available for other plans to share. Changes to a library section are reflected in all plans that use it.
linked section	In PeopleSoft Enterprise Incentive Management, a section that is defined in a plan template but appears in a plan. Changes to linked sections propagate to plans using that section.
linked variable	In PeopleSoft Enterprise Incentive Management, a variable that is defined and maintained in a plan template and that also appears in a plan. Changes to linked variables propagate to plans using that variable.
LMS	Abbreviation for <i>learning management system</i> . In PeopleSoft Enterprise Campus Solutions, LMS is a PeopleSoft Student Records feature that provides a common set of interoperability standards that enable the sharing of instructional content and data between learning and administrative environments.

load	In PeopleSoft 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.
local functionality	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.
location	Locations enable you to indicate the different types of addresses—for a company, for example, one address to receive bills, another for shipping, a third for postal deliveries, and a separate street address. Each address has a different location number. The primary location—indicated by a <i>1</i> —is the address you use most often and may be different from the main address.
logistical task	In PeopleSoft Services Procurement, an administrative task that is related to hiring a service provider. Logistical tasks are linked to the service type on the work order so that different types of services can have different logistical tasks. Logistical tasks include both preapproval tasks (such as assigning a new badge or ordering a new laptop) and postapproval tasks (such as scheduling orientation or setting up the service provider email). The logistical tasks can be mandatory or optional. Mandatory preapproval tasks must be completed before the work order is approved. Mandatory postapproval tasks, on the other hand, must be completed before a work order is released to a service provider.
market template	In PeopleSoft Enterprise Incentive Management, additional functionality that is specific to a given market or industry and is built on top of a product category.
mass change	In PeopleSoft Enterprise Campus Solutions, mass change is a SQL generator that can be used to create specialized functionality. Using mass change, you can set up a series of Insert, Update, or Delete SQL statements to perform business functions that are specific to the institution. See also <i>3C engine</i> .
match group	In PeopleSoft Receivables, a group of receivables items and matching offset items. The system creates match groups by using user-defined matching criteria for selected field values.
MCF server	Abbreviation for <i>PeopleSoft MultiChannel Framework server</i> . Comprises the universal queue server and the MCF log server. Both processes are started when <i>MCF Servers</i> is selected in an application server domain configuration.
merchandising activity	In PeopleSoft Promotions Management, a specific discount type that is associated with a trade promotion (such as off-invoice, billback or rebate, or lump-sum payment) that defines the performance that is required to receive the discount. In the industry, you may know this as an offer, a discount, a merchandising event, an event, or a tactic.
meta-SQL	Meta-SQL constructs expand into platform-specific Structured Query Language (SQL) substrings. They are used in functions that pass SQL strings, such as in SQL objects, the SQLExec function, and PeopleSoft Application Engine programs.
metastring	Metastrings are special expressions included in SQL string literals. The metastrings, prefixed with a percent (%) symbol, are included directly in the string literals. They expand at run time into an appropriate substring for the current database platform.
multibook	In PeopleSoft General Ledger, multiple ledgers having multiple-base currencies that are defined for a business unit, with the option to post a single transaction to all base currencies (all ledgers) or to only one of those base currencies (ledgers).
multicurrency	The ability to process transactions in a currency other than the business unit's base currency.

national allowance	In PeopleSoft 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.
need	In PeopleSoft Enterprise Campus Solutions, the difference between the cost of attendance (COA) and the expected family contribution (EFC). It is the gap between the cost of attending the school and the student's resources. The financial aid package is based on the amount of financial need. The process of determining a student's need is called <i>need analysis</i> .
node-oriented tree	A tree that is based on a detail structure, but the detail values are not used.
pagelet	Each block of content on the home page is called a pagelet. These pagelets display summary information within a small rectangular area on the page. The pagelet provide users with a snapshot of their most relevant PeopleSoft and non-PeopleSoft content.
participant	In PeopleSoft Enterprise Incentive Management, participants are recipients of the incentive compensation calculation process.
participant object	Each participant object may be related to one or more compensation objects. See also <i>compensation object</i> .
partner	A company that supplies products or services that are resold or purchased by the enterprise.
pay cycle	In PeopleSoft Payables, a set of rules that define the criteria by which it should select scheduled payments for payment creation.
payment shuffle	In PeopleSoft Enterprise Campus Solutions, a process allowing payments that have been previously posted to a student's account to be automatically reapplied when a higher priority payment is posted or the payment allocation definition is changed.
pending item	In PeopleSoft Receivables, an individual receivable (such as an invoice, a credit memo, or a write-off) that has been entered in or created by the system, but hasn't been posted.
PeopleCode	PeopleCode is a proprietary language, executed by the PeopleSoft 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.
PeopleCode event	An action that a user takes upon an object, usually a record field, that is referenced within a PeopleSoft page.
PeopleSoft Internet Architecture	The fundamental architecture on which PeopleSoft 8 applications are constructed, consisting of a relational database management system (RDBMS), an application server, a web server, and a browser.
performance measurement	In PeopleSoft Enterprise Incentive Management, a variable used to store data (similar to an aggregator, but without a predefined formula) within the scope of an incentive plan. Performance measures are associated with a plan calendar, territory, and participant. Performance measurements are used for quota calculation and reporting.
period context	In PeopleSoft Enterprise Incentive Management, because a participant typically uses the same compensation plan for multiple periods, the period context associates a plan context with a specific calendar period and fiscal year. The period context references the associated plan context, thus forming a chain. Each plan context has a corresponding set of period contexts.
person of interest	A person about whom the organization maintains information but who is not part of the workforce.

personal portfolio	In PeopleSoft Enterprise Campus Solutions, the user-accessible menu item that contains an individual's name, address, telephone number, and other personal information.
plan	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.
plan context	In PeopleSoft Enterprise Incentive Management, correlates a participant with the compensation plan and node to which the participant is assigned, enabling the PeopleSoft Enterprise Incentive Management system to find anything that is associated with the node and that is required to perform compensation processing. Each participant, node, and plan combination represents a unique plan context—if three participants are on a compensation structure, each has a different plan context. Configuration plans are identified by plan contexts and are associated with the participants that refer to them.
plan template	In PeopleSoft Enterprise Incentive Management, the base from which a plan is created. A plan template contains common sections and variables that are inherited by all plans that are created from the template. A template may contain steps and sections that are not visible in the plan definition.
planned learning	In PeopleSoft Enterprise Learning Management, a self-service repository for all of a learner's planned learning activities and programs.
planning instance	In PeopleSoft Supply Planning, a set of data (business units, items, supplies, and demands) constituting the inputs and outputs of a supply plan.
population	In PeopleSoft Enterprise Campus Solutions, the middle level of the three-level classification structure that you define in PeopleSoft Enterprise Recruiting and Admissions for enrollment management. You can define a population level, link it to other levels, and set enrollment target numbers for it. See also <i>division</i> and <i>cohort</i> .
portal registry	In PeopleSoft 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.
price list	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.
price rule	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.
price rule condition	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.
price rule key	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.
primacy number	In PeopleSoft Enterprise Campus Solutions, a number that the system uses to prioritize financial aid applications when students are enrolled in multiple academic careers and academic programs at the same time. The Consolidate Academic Statistics process uses the primacy number indicated for both the career and program at the institutional level to determine a student's primary career and program. The system also uses the

	number to determine the primary student attribute value that is used when you extract data to report on cohorts. The lowest number takes precedence.
primary name type	In PeopleSoft Enterprise Campus Solutions, the name type that is used to link the name stored at the highest level within the system to the lower-level set of names that an individual provides.
process category	In PeopleSoft Process Scheduler, processes that are grouped for server load balancing and prioritization.
process group	In PeopleSoft Financials, a group of application processes (performed in a defined order) that users can initiate in real time, directly from a transaction entry page.
process definition	Process definitions define each run request.
process instance	A unique number that identifies each process request. This value is automatically incremented and assigned to each requested process when the process is submitted to run.
process job	You can link process definitions into a job request and process each request serially or in parallel. You can also initiate subsequent processes based on the return code from each prior request.
process request	A single run request, such as a Structured Query Report (SQR), a COBOL or Application Engine program, or a Crystal report that you run through PeopleSoft Process Scheduler.
process run control	A PeopleTools variable used to retain PeopleSoft Process Scheduler values needed at runtime for all requests that reference a run control ID. Do not confuse these with application run controls, which may be defined with the same run control ID, but only contain information specific to a given application process request.
product category	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.
programs	In PeopleSoft Enterprise Learning Management, a high-level grouping that guides the learner along a specific learning path through sections of catalog items. PeopleSoft Enterprise Learning Systems provides two types of programs—curricula and certifications.
progress log	In PeopleSoft Services Procurement, tracks deliverable-based projects. This is similar to the time sheet in function and process. The service provider contact uses the progress log to record and submit progress on deliverables. The progress can be logged by the activity that is performed, by the percentage of work that is completed, or by the completion of milestone activities that are defined for the project.
project transaction	In PeopleSoft Project Costing, an individual transaction line that represents a cost, time, budget, or other transaction row.
promotion	In PeopleSoft Promotions Management, a trade promotion, which is typically funded from trade dollars and used by consumer products manufacturers to increase sales volume.
prospects	In PeopleSoft Enterprise Campus Solutions, students who are interested in applying to the institution. In PeopleSoft Enterprise Contributor Relations, individuals and organizations that are most likely to make substantial financial commitments or other types of commitments to the institution.
publishing	In PeopleSoft Enterprise Incentive Management, a stage in processing that makes incentive-related results available to participants.

rating components	In PeopleSoft Enterprise Campus Solutions, variables used with the Equation Editor to retrieve specified populations.
record group	A set of logically and functionally related control tables and views. Record groups help enable TableSet sharing, which eliminates redundant data entry. Record groups ensure that TableSet sharing is applied consistently across all related tables and views.
record input VAT flag	Abbreviation for <i>record input value-added tax flag</i> . Within PeopleSoft 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.
record output VAT flag	Abbreviation for <i>record output value-added tax flag</i> . See <i>record input VAT flag</i> .
recname	The name of a record that is used to determine the associated field to match a value or set of values.
recognition	In PeopleSoft Enterprise Campus Solutions, the recognition type indicates whether the PeopleSoft Enterprise Contributor Relations donor is the primary donor of a commitment or shares the credit for a donation. Primary donors receive hard credit that must total 100 percent. Donors that share the credit are given soft credit. Institutions can also define other share recognition-type values such as memo credit or vehicle credit.
reference data	In PeopleSoft Sales Incentive Management, system objects that represent the sales organization, such as territories, participants, products, customers, channels, and so on.
reference object	In PeopleSoft Enterprise Incentive Management, this dimension-type object further defines the business. Reference objects can have their own hierarchy (for example, product tree, customer tree, industry tree, and geography tree).
reference transaction	In commitment control, a reference transaction is a source transaction that is referenced by a higher-level (and usually later) source transaction, in order to automatically reverse all or part of the referenced transaction's budget-checked amount. This avoids duplicate postings during the sequential entry of the transaction at different commitment levels. For example, the amount of an encumbrance transaction (such as a purchase order) will, when checked and recorded against a budget, cause the system to concurrently reference and relieve all or part of the amount of a corresponding pre-encumbrance transaction, such as a purchase requisition.
regional sourcing	In PeopleSoft Purchasing, provides the infrastructure to maintain, display, and select an appropriate vendor and vendor pricing structure that is based on a regional sourcing model where the multiple ship to locations are grouped. Sourcing may occur at a level higher than the ship to location.
relationship object	In PeopleSoft Enterprise Incentive Management, these objects further define a compensation structure to resolve transactions by establishing associations between compensation objects and business objects.
remote data source data	Data that is extracted from a separate database and migrated into the local database.
REN server	Abbreviation for <i>real-time event notification server</i> in PeopleSoft MultiChannel Framework.
requester	In PeopleSoft eSettlements, an individual who requests goods or services and whose ID appears on the various procurement pages that reference purchase orders.

reversal indicator	In PeopleSoft Enterprise Campus Solutions, an indicator that denotes when a particular payment has been reversed, usually because of insufficient funds.
role	Describes how people fit into PeopleSoft Workflow. A role is a class of users who perform the same type of work, such as clerks or managers. Your business rules typically specify what user role needs to do an activity.
role user	A PeopleSoft Workflow user. A person's role user ID serves much the same purpose as a user ID does in other parts of the system. PeopleSoft Workflow uses role user IDs to determine how to route worklist items to users (through an email address, for example) and to track the roles that users play in the workflow. Role users do not need PeopleSoft user IDs.
roll up	In a tree, to roll up is to total sums based on the information hierarchy.
run control	A run control is a type of online page that is used to begin a process, such as the batch processing of a payroll run. Run control pages generally start a program that manipulates data.
run control ID	A unique ID to associate each user with his or her own run control table entries.
run-level context	In PeopleSoft Enterprise Incentive Management, associates a particular run (and batch ID) with a period context and plan context. Every plan context that participates in a run has a separate run-level context. Because a run cannot span periods, only one run-level context is associated with each plan context.
search query	You use this set of objects to pass a query string and operators to the search engine. The search index returns a set of matching results with keys to the source documents.
search/match	In PeopleSoft Enterprise Campus Solutions and PeopleSoft Enterprise Human Resources Management Solutions, a feature that enables you to search for and identify duplicate records in the database.
seasonal address	In PeopleSoft Enterprise Campus Solutions, an address that recurs for the same length of time at the same time of year each year until adjusted or deleted.
section	In PeopleSoft Enterprise Incentive Management, a collection of incentive rules that operate on transactions of a specific type. Sections enable plans to be segmented to process logical events in different sections.
security event	In commitment control, security events trigger security authorization checking, such as budget entries, transfers, and adjustments; exception overrides and notifications; and inquiries.
serial genealogy	In PeopleSoft Manufacturing, the ability to track the composition of a specific, serial-controlled item.
serial in production	In PeopleSoft Manufacturing, enables the tracing of serial information for manufactured items. This is maintained in the Item Master record.
service impact	In PeopleSoft Enterprise Campus Solutions, the resulting action triggered by a service indicator. For example, a service indicator that reflects nonpayment of account balances by a student might result in a service impact that prohibits registration for classes.
service indicator	In PeopleSoft Enterprise Campus Solutions, indicates services that may be either withheld or provided to an individual. Negative service indicators indicate holds that prevent the individual from receiving specified services, such as check-cashing privileges or registration for classes. Positive service indicators designate special services that are provided to the individual, such as front-of-line service or special services for disabled students.

session	<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>
session template	In PeopleSoft Enterprise Learning Management, enables you to set up common activity characteristics that may be reused while scheduling a PeopleSoft Enterprise Learning Management activity—characteristics such as days of the week, start and end times, facility and room assignments, instructors, and equipment. A session pattern template can be attached to an activity that is being scheduled. Attaching a template to an activity causes all of the default template information to populate the activity session pattern.
setup relationship	In PeopleSoft Enterprise Incentive Management, a relationship object type that associates a configuration plan with any structure node.
share driver expression	In PeopleSoft 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.
single signon	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.
source key process	In PeopleSoft Enterprise Campus Solutions, a process that relates a particular transaction to the source of the charge or financial aid. On selected pages, you can drill down into particular charges.
source transaction	In commitment control, any transaction generated in a PeopleSoft 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.
speed key	See <i>communication key</i> .
SpeedChart	A user-defined shorthand key that designates several ChartKeys to be used for voucher entry. Percentages can optionally be related to each ChartKey in a SpeedChart definition.
SpeedType	A code representing a combination of ChartField values. SpeedTypes simplify the entry of ChartFields commonly used together.
staging	A method of consolidating selected partner offerings with the offerings from the enterprise's other partners.
standard letter code	In PeopleSoft Enterprise Campus Solutions, a standard letter code used to identify each letter template available for use in mail merge functions. Every letter generated in the system must have a standard letter code identification.
statutory account	Account required by a regulatory authority for recording and reporting financial results. In PeopleSoft, this is equivalent to the Alternate Account (ALTACCT) ChartField.

step	In PeopleSoft Sales Incentive Management, a collection of sections in a plan. Each step corresponds to a step in the job run.
storage level	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.
subcustomer qualifier	A value that groups customers into a division for which you can generate detailed history, aging, events, and profiles.
Summary ChartField	You use summary ChartFields to create summary ledgers that roll up detail amounts based on specific detail values or on selected tree nodes. When detail values are summarized using tree nodes, summary ChartFields must be used in the summary ledger data record to accommodate the maximum length of a node name (20 characters).
summary ledger	An accounting feature used primarily in allocations, inquiries, and PS/nVision reporting to store combined account balances from detail ledgers. Summary ledgers increase speed and efficiency of reporting by eliminating the need to summarize detail ledger balances each time a report is requested. Instead, detail balances are summarized in a background process according to user-specified criteria and stored on summary ledgers. The summary ledgers are then accessed directly for reporting.
summary time period	In PeopleSoft Business Planning, any time period (other than a base time period) that is an aggregate of other time periods, including other summary time periods and base time periods, such as quarter and year total.
summary tree	A tree used to roll up accounts for each type of report in summary ledgers. Summary trees enable you to define trees on trees. In a summary tree, the detail values are really nodes on a detail tree or another summary tree (known as the <i>basis</i> tree). A summary tree structure specifies the details on which the summary trees are to be built.
syndicate	To distribute a production version of the enterprise catalog to partners.
system function	In PeopleSoft Receivables, an activity that defines how the system generates accounting entries for the general ledger.
TableSet	A means of sharing similar sets of values in control tables, where the actual data values are different but the structure of the tables is the same.
TableSet sharing	Shared data that is stored in many tables that are based on the same TableSets. Tables that use TableSet sharing contain the SETID field as an additional key or unique identifier.
target currency	The value of the entry currency or currencies converted to a single currency for budget viewing and inquiry purposes.
tax authority	In PeopleSoft Enterprise Campus Solutions, a user-defined element that combines a description and percentage of a tax with an account type, an item type, and a service impact.
template	A template is HTML code associated with a web page. It defines the layout of the page and also where to get HTML for each part of the page. In PeopleSoft, 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.
territory	In PeopleSoft Sales Incentive Management, hierarchical relationships of business objects, including regions, products, customers, industries, and participants.
3C engine	Abbreviation for <i>Communications, Checklists, and Comments engine</i> . In PeopleSoft Enterprise Campus Solutions, the 3C engine enables you to automate business processes that involve additions, deletions, and updates to communications, checklists,

and comments. You define events and triggers to engage the engine, which runs the mass change and processes the 3C records (for individuals or organizations) immediately and automatically from within business processes.

3C group	Abbreviation for <i>Communications, Checklists, and Comments group</i> . In PeopleSoft Enterprise Campus Solutions, a method of assigning or restricting access privileges. A 3C group enables you to group specific communication categories, checklist codes, and comment categories. You can then assign the group inquiry-only access or update access, as appropriate.
TimeSpan	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.
trace usage	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.
transaction allocation	In PeopleSoft Enterprise Incentive Management, the process of identifying the owner of a transaction. When a raw transaction from a batch is allocated to a plan context, the transaction is duplicated in the PeopleSoft Enterprise Incentive Management transaction tables.
transaction state	In PeopleSoft Enterprise Incentive Management, a value assigned by an incentive rule to a transaction. Transaction states enable sections to process only transactions that are at a specific stage in system processing. After being successfully processed, transactions may be promoted to the next transaction state and “picked up” by a different section for further processing.
Translate table	A system edit table that stores codes and translate values for the miscellaneous fields in the database that do not warrant individual edit tables of their own.
tree	The graphical hierarchy in PeopleSoft systems that displays the relationship between all accounting units (for example, corporate divisions, projects, reporting groups, account numbers) and determines roll-up hierarchies.
tuition lock	In PeopleSoft Enterprise Campus Solutions, a feature in the Tuition Calculation process that enables you to specify a point in a term after which students are charged a minimum (or <i>locked</i>) fee amount. Students are charged the locked fee amount even if they later drop classes and take less than the normal load level for that tuition charge.
unclaimed transaction	In PeopleSoft Enterprise Incentive Management, a transaction that is not claimed by a node or participant after the allocation process has completed, usually due to missing or incomplete data. Unclaimed transactions may be manually assigned to the appropriate node or participant by a compensation administrator.
universal navigation header	Every PeopleSoft portal includes the universal navigation header, intended to appear at the top of every page as long as the user is signed on to the portal. In addition to providing access to the standard navigation buttons (like Home, Favorites, and signoff) the universal navigation header can also display a welcome message for each user.
update access	In PeopleSoft Enterprise Campus Solutions, a type of security access that permits the user to edit and update data. See also <i>inquiry access</i> .
user interaction object	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).

variable	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.
VAT exception	Abbreviation for <i>value-added tax exception</i> . A temporary or permanent exemption from paying VAT that is granted to an organization. This terms refers to both VAT exoneration and VAT suspension.
VAT exempt	Abbreviation for <i>value-added tax exempt</i> . Describes goods and services that are not subject to VAT. Organizations that supply exempt goods or services are unable to recover the related input VAT. This is also referred to as exempt without recovery.
VAT exoneration	Abbreviation for <i>value-added tax exoneration</i> . An organization that has been granted a permanent exemption from paying VAT due to the nature of that organization.
VAT suspension	Abbreviation for <i>value-added tax suspension</i> . An organization that has been granted a temporary exemption from paying VAT.
warehouse	A PeopleSoft data warehouse that consists of predefined ETL maps, data warehouse tools, and DataMart definitions.
work order	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.
worker	A person who is part of the workforce; an employee or a contingent worker.
workset	A group of people and organizations that are linked together as a set. You can use worksets to simultaneously retrieve the data for a group of people and organizations and work with the information on a single page.
worksheet	A way of presenting data through a PeopleSoft Business Analysis Modeler interface that enables users to do in-depth analysis using pivoting tables, charts, notes, and history information.
worklist	The automated to-do list that PeopleSoft Workflow creates. From the worklist, you can directly access the pages you need to perform the next action, and then return to the worklist for another item.
XML schema	An XML definition that standardizes the representation of application messages, component interfaces, or business interlinks.
yield by operation	In PeopleSoft Manufacturing, the ability to plan the loss of a manufactured item on an operation-by-operation basis.
zero-rated VAT	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

- abbreviations in message column of event stream event pane 10
- action tools in Virtual Runner 26
- adding loops 33
- additional documentation viii
- All Virtual Scripts list box 25
- API playback timing 44
- application fundamentals vii
- automatic parameter value linking 14

B

- buttons in the event stream event pane 11
- buttons in the script list pane 12

C

- capturing and importing test results 30
 - capturing results 30
 - importing test results 31
 - viewing test results 31
- capturing test results 30
- code for data capture 3
- columns in the script list pane 12
- command line for Virtual Script Player launch 22
- Command section of initialization file parameters 19
- comments, submitting xii
- common elements xii
- contact information xii
- correcting uninitialized user handle errors 59
- creating master scripts 36
- creating virtual scripts 29
 - capturing and importing test results 30, 31
 - See Also* capturing test results; importing test results; viewing test results
 - editing the virtual script 32, 33, 34, 35, 36
 - See Also* adding loops; creating master scripts; linking values in entry scripts; linking values in

- inquiry scripts; script generation; value-linking parameters
- cross-references xi
- Customer Connection website viii

D

- data capture components 3
 - event stream 5
 - functions of EnterpriseOne software hooks 5
 - functions of PeopleSoft AutoPilot hooks 5
 - PeopleSoft AutoPilot 4, 5
 - See Also* script creation; script playback; script playback configuration
 - PeopleSoft AutoPilot and EnterpriseOne software code 5
 - PeopleSoft AutoPilot Playback Results Detail Table (F97214) 6
- data capture components, PeopleSoft AutoPilot 4
- data capture for virtual scripts 3
 - code for data capture 3
 - data capture components 4, 5, 6
 - See Also* PeopleSoft AutoPilot Playback Results Detail Table (F97214); event stream; PeopleSoft AutoPilot; PeopleSoft AutoPilot and EnterpriseOne software code
 - event stream 3
 - PeopleSoft AutoPilot 3
 - PeopleSoft AutoPilot Playback Results Detail Table (F97214) 3
- data capture for virtual scripts, data capture components 3
- date formatting 23
- debugging LoadRunner 59
- debugging Virtual Runner 58
- debugging Virtual User Tool scripts 52
 - correcting uninitialized user handle errors 59
 - diagnosing business function failures in EnterpriseOne 54

- displaying business function
 - parameters 53
- identifying and correcting duplicate key errors 57
- preventing multiple script playback problems 58, 59
 - See Also* debugging LoadRunner; debugging Virtual Runner
- rectifying irregular transaction times 57
- troubleshooting value-linking errors 53, 54, 55, 56
 - See Also* identifying and correcting variable value-linking errors; researching value-linking errors in the Virtual Script Editor; verifying that value linking is functioning
- verifying the validity of Virtual User Tool script data 56
- defining a script using LoadRunner 40
- defining the host machine using LoadRunner 40
- defining virtual users using LoadRunner 41
- diagnosing a EnterpriseOne environment problem 51
- diagnosing business function failures in EnterpriseOne 54
- displaying business function parameters 53
- documentation
 - printed viii
 - related viii
 - updates viii
- dynamic loop creation 16
- dynamic loop designation 16
- dynamic loop editing 16

E

- editing the virtual script 32
 - adding loops 33
 - creating master scripts 36
 - linking values in entry scripts 35
 - linking values in inquiry scripts 34
 - script generation 35
 - value-linking parameters 33
- EnterpriseOne Virtual User Tool
 - components, Virtual Script Editor 15
 - See Also* looping identification
- EnterpriseOne environment initialization by the Virtual Script Player 22

- EnterpriseOne Virtual User Tool
 - components 7, 8, 19, 24, 25
 - See Also* Virtual Runner; Virtual Script Editor; Virtual Script Player; VSMEditor
 - creating virtual scripts 29, 30, 32
 - See Also* capturing and importing test results; editing the virtual script
 - data capture for virtual scripts 3
 - See Also* code for data capture; data capture components; event stream; PeopleSoft AutoPilot; PeopleSoft AutoPilot Playback Results Detail Table (F97214)
 - running virtual scripts 37, 38, 40
 - See Also* from a command line; from a single workstation; launching and managing multiple script playback; using Virtual Runner
 - special considerations for simulated playback 43, 45, 46
 - See Also* call level; playback timing; synchronous and asynchronous calls; think times
 - Virtual Script Editor 13
 - See Also* parameter value linking
- EnterpriseOne Virtual user tool components
 - Virtual Script Player 24
 - See Also* limitations
- EnterpriseOne Virtual User Tool
 - components 7
 - Virtual Runner 25
 - Virtual Script Editor 8, 11, 12, 16, 17, 18
 - See Also* event graph; hRequest handle value linking; parameter detail pane; script list pane; thread identification; timing interval maintenance; virtual script generation
 - Virtual Script Player 19, 22, 23, 24
 - See Also* command line launch; date formatting; initialization file parameters; modes of operation; preprocessing of valid values list data; script failure; valid values list processing
 - VSMEditor 24, 25
 - See Also* All Virtual Scripts list box; Master Scripts list box; VSM files

- event graph 11
- event pane 10
- event stream 3, 5
- event stream components
 - event pane 10
 - See Also* abbreviations in message column
- event stream event pane
 - abbreviations in message column 10
 - buttons 11

F

- finding error entries in the Virtual User Tool log file 50
- functions of EnterpriseOne software hooks 5
- functions of PeopleSoft AutoPilot hooks 5

G

- gathering LoadRunner results 41
- generating the virtual script 35
- glossary 61

H

- hRequest handle value linking 16

I

- identifying a EnterpriseOne environment problem 51, 52
- identifying and correcting duplicate key errors 57
- identifying and correcting variable value-linking errors 56
- importing test results 31
- initialization file parameters for Virtual Script Player 19
 - Command section 19
 - Log section 21
 - Paths section 20
 - Timing section 20
- interthread timing 44
- investigating EnterpriseOne errors 52

L

- launching and managing multiple script playback 40
 - defining a script 40
 - defining the host machine 40
 - defining virtual users 41

- gathering LoadRunner results 41
 - running virtual playback from the LoadRunner controller 41
 - setting rendezvous points 41
- limitations of Virtual Script Player 24
- linking parameters in entry scripts 35
- linking parameters in inquiry scripts 34
- locating the causes of Virtual User Tool script failure
 - finding error entries in the log file 50
 - identifying a EnterpriseOne environment problem 51
 - locating the log file in the event of early script failure 50
 - setting the MessageLevel parameter 51
- locating the causes of Virtual User Tool script failures 49
 - identifying a EnterpriseOne environment problem 51
 - See Also* diagnosing a EnterpriseOne environment problem
 - identifying an EnterpriseOne environment problem 52
 - See Also* investigating EnterpriseOne errors
- locating the log file in the event of early script failure 50
- Log section of initialization file parameters 21
- looping identification 15
 - dynamic loop creation 16
 - dynamic loop designation 16
 - dynamic loop editing 16

M

- manual parameter value linking 15
- Master Scripts list box 25
- MMA Partners viii
- modes of operation for Virtual Script Player 23

N

- notes xi

P

- parameter detail pane 11
- parameter identification
 - source parameters 13
 - target parameters 13

- parameter value linking 13
 - automatic 14
 - manual 15
- paths section of initialization file
- parameters 20
- PeopleBooks
 - ordering viii
- PeopleCode, typographical conventions x
- PeopleSoft application fundamentals vii
- PeopleSoft AutoPilot 3, 4
 - script creation 4
 - script playback 5
 - script playback configuration 4
- PeopleSoft AutoPilot and EnterpriseOne
 - software code 5
- PeopleSoft AutoPilot playback
 - configuration 4
- PeopleSoft AutoPilot Playback Results
 - Detail Table (F97214) 3
- PeopleSoft AutoPilot script creation 4
- PeopleSoft AutoPilot script playback 5
- PeopleSoft Virtual User Tool components
 - Virtual Script Editor 10
 - See Also* event pane
- placement of hooks for data capture 5
- playback timing 43
 - API playback timing 44
 - interthread timing 44
- player session columns in Virtual
 - Runner 26
- preprocessing of valid values list data 23
- prerequisites vii
- preventing multiple script playback
 - problems 58
 - debugging LoadRunner 59
 - debugging Virtual Runner 58
- printed documentation viii

R

- rectifying irregular transaction times 57
- related documentation viii
- researching value-linking errors in the
 - Virtual Script Editor 54
- running virtual playback from the
 - LoadRunner controller 41
- running virtual scripts 37
 - from a command line 38
 - from a single workstation 37
 - launching and managing multiple script
 - playback 40, 41

- See Also* defining a script; defining the host machine; defining virtual users; gathering LoadRunner results; running virtual playback from the LoadRunner controller; setting rendezvous points
- using Virtual Runner 38
- running virtual scripts from a command line 38
- running virtual scripts from a single workstation 37
- running virtual scripts using Virtual Runner 38

S

- script failure 24
- script list pane 12
 - buttons 12
 - columns 12
- Setting rendezvous points using
 - LoadRunner 41
- setting the MessageLevel parameter 51
- source parameters 13
- special considerations for simulated
 - playback 43
 - call level 45
 - playback timing 43, 44
 - See Also* API playback timing; interthread timing
 - synchronous and asynchronous calls 45
 - Think times 46
- suggestions, submitting xii
- synchronous and asynchronous calls 45

T

- target parameters 13
- terms 61
- Think Times 46
- Thread identification 17
- timing interval maintenance 17
- timing section of initialization file
 - parameters 20
- troubleshooting tips and techniques for
 - Virtual User Tool scripts 49
 - debugging Virtual User Tool scripts 52, 53, 54, 56, 57, 58, 59
 - See Also* correcting uninitialized user handle errors; diagnosing business function failures in EnterpriseOne;

- displaying business function parameters; identifying and correcting duplicate key errors; preventing multiple script playback problems; rectifying irregular transaction times; troubleshooting value-linking errors; verifying the validity of Virtual User Tool script data
- locating the causes of Virtual User Tool script failures 49, 50, 51
 - See Also* finding error entries in the log file; identifying a OneWorld environment problem; locating the log file in the event of early script failure; setting the MessageLevel parameter
- troubleshooting value-linking errors 53
 - identifying and correcting variable value-linking errors 56
 - researching value-linking errors in the Virtual Script Editor 54
 - verifying that value linking is functioning 55
- typographical conventions x

V

- valid values list processing 23
- value-linking parameters 33
- verifying that value linking is functioning 55
- verifying the validity of Virtual User Tool script data 56
- viewing test results 31
- Virtual Runner 25
 - components 26
 - See Also* Actions tools; Player session columns
- Virtual script editor
 - parameter identification 13
 - See Also* target parameters
- Virtual Script Editor
 - event graph 11
 - event pane 10
 - hRequest handle value linking 16
 - looping identification 15, 16
 - See Also* dynamic loop creation; dynamic loop designation; dynamic loop editing
 - parameter detail pane 11

- parameter identification 13
 - See Also* source parameters
- parameter value linking 13, 14, 15
 - See Also* automatic; manual
- script list pane 12
- thread identification 17
- timing interval maintenance 17
- virtual script generation 18
- virtual script generation 18
- Virtual Script Player 19
 - command line launch 22
 - date formatting 23
 - EnterpriseOne environment initialization 22
 - initialization file parameters 19, 20, 21
 - See Also* Command section; Log section; Paths section; Timing section
 - limitations 24
 - modes of operation 23
 - preprocessing of valid values list data 23
 - script failure 24
 - valid values list processing 23
- visual cues xi
- VSM files 25
- VSMEEditor 24
 - All Virtual Scripts list box 25
 - Master Scripts list box 25
 - VSM files 25

W

- warnings xi

