



**Mercury IT Governance Center™  
Mercury Change Management Extension  
for PeopleSoft® Enterprise™ Guide**

Version: 6.0.0



**MERCURY™**

This manual, and the accompanying software and other documentation, is protected by U.S. and international copyright laws, and may be used only in accordance with the accompanying license agreement. Features of the software, and of other products and services of Mercury Interactive Corporation, may be covered by one or more of the following patents: United States: 5,511,185; 5,657,438; 5,701,139; 5,870,559; 5,958,008; 5,974,572; 6,137,782; 6,138,157; 6,144,962; 6,205,122; 6,237,006; 6,341,310; 6,360,332, 6,449,739; 6,470,383; 6,477,483; 6,549,944; 6,560,564; 6,564,342; 6,587,969; 6,631,408; 6,631,411; 6,633,912; 6,694,288; 6,738,813; 6,738,933; 6,754,701; 6,792,460 and 6,810,494. Australia: 763468 and 762554. Other patents pending. All rights reserved.

Mercury, Mercury Interactive, the Mercury logo, the Mercury Interactive logo, LoadRunner, WinRunner, SiteScope and TestDirector are trademarks of Mercury Interactive Corporation and may be registered in certain jurisdictions. The absence of a trademark from this list does not constitute a waiver of Mercury's intellectual property rights concerning that trademark.

All other company, brand and product names may be trademarks or registered trademarks of their respective holders. Mercury disclaims any responsibility for specifying which marks are owned by which companies or which organizations.

Mercury  
379 North Whisman Road  
Mountain View, CA 94043  
Tel: (650) 603-5200  
Toll Free: (800) TEST-911  
Customer Support: (877) TEST-HLP  
Fax: (650) 603-5300

© 1997–2005 Mercury Interactive Corporation. All rights reserved.

If you have any comments or suggestions regarding this document, please send email to [documentation@mercury.com](mailto:documentation@mercury.com).

# Table of Contents

---

<b>List of Figures .....</b>	<b>ix</b>
<b>List of Tables .....</b>	<b>xi</b>
<b>Chapter 1: Introduction.....</b>	<b>15</b>
About This Document.....	16
Who Should Read This Document .....	17
Prerequisite Documents .....	17
Related Documents.....	18
Overview of the Extension.....	19
General Overview of Extension for PeopleSoft Enterprise.....	19
What's New in Release 6.0.0.....	20
Support for PeopleSoft Version 8.44.....	20
Desupport of PeopleSoft Version 7.5 and Microsoft SQL Server .....	21
New PeopleSoft Comparison Report .....	21
Enhancements to PeopleSoft Project Rollback Subworkflow.....	21
Enhancements to PeopleSoft Project (ORACLE) Object Type .....	21
Pre-configured Dashboard Pages .....	22
Reports Accessible Only from Standard Interface.....	22
Licensing.....	22
Prerequisite Knowledge and Experience .....	23
<b>Chapter 2: Installing or Upgrading the Extension.....</b>	<b>25</b>
Overview of Installation .....	26
System Requirements .....	26
Installing or Upgrading to Release 6.0.0 .....	29
General Upgrade Impacts and Guidelines.....	29

Impacts of Upgrading from Release 5.0.0.....	30
Impacts of Upgrading from Release 5.5.0.....	31
Dashboard Menus and Tab Templates (Pre-configured Dashboard Pages) .....	31
Desupport of PeopleSoft Version 7.5 and SQL Server .....	32
New PeopleSoft Comparison Report .....	36
Preparing for Installation.....	37
Downloading and Installing the Extension .....	38
Downloading the Installation Files.....	38
Installation Procedure .....	39
Logs Generated During Installation.....	41
Post-Installation Procedures.....	42
<b>Chapter 3: Configuring the Extension.....</b>	<b>43</b>
Configuring Environments.....	44
Configuring PeopleSoft Patch Stage .....	47
Configuring Development/Test/Production Instances of PeopleSoft .....	48
Configuring Development/Test/Production Environment Groups .....	50
Configuring PeopleSoft Users.....	51
Mass Password Change Functionality .....	53
<b>Chapter 4: Extension Entities.....</b>	<b>55</b>
Object Types.....	57
Overview of Object Types .....	57
Reference Object Types .....	58
List of Object Types.....	58
PeopleSoft 8.1.x Patch, 8.4.x Patch, and 8.44 Patch Object Types .....	59
PeopleSoft COBOL Copybook Object Type .....	60
PeopleSoft COBOL DMS Object Type .....	62
PeopleSoft COBOL Program Object Type .....	63
PeopleSoft Data File Object Type.....	65
PeopleSoft Load DMS Object Type .....	66
PeopleSoft Project (ORACLE) Object Type.....	68
PeopleSoft SQC Object Type.....	71
PeopleSoft SQL Script Object Type .....	72
PeopleSoft SQR Object Type.....	74
Request Types .....	75
Overview of Request Types.....	75
Reference Request Types.....	75
List of Request Types .....	76
PeopleSoft Bug Request Type .....	77
PeopleSoft Enhancement Request Type.....	79
PeopleSoft Patch Request Type .....	81
Workflows.....	83

---

Overview of Workflows .....	83
Reference Workflows .....	83
List of Workflows.....	84
PeopleSoft Bug/Enhancement Workflow.....	85
PeopleSoft Development Workflow .....	87
PeopleSoft Development Workflow w/ Project Rollback & Object Warning Workflow .....	89
PeopleSoft Patch Request Workflow .....	96
PeopleSoft Patch Applicator Workflows.....	97
Report Types.....	98
Overview of Report Types.....	98
Reference Report Types .....	98
List of Report Types .....	99
PeopleSoft Compare Custom Database Setup Report.....	100
PeopleSoft Comparison Report .....	102
PeopleSoft Customized Objects Report.....	104
PeopleSoft File Comparison Report .....	106
PeopleSoft Object Customization Detail Report.....	107
PeopleSoft Object Migrations Report .....	109
PeopleSoft Patches Applied to an Environment Report .....	111
PeopleSoft Project Migration Impact Analysis Report.....	114
PeopleSoft Request Detail Report.....	116
PeopleSoft Version Control - Baseline Report.....	119
PeopleSoft Version Control - Break Lock Report.....	121
PeopleSoft Version Control - Check In Report .....	124
PeopleSoft Version Control - Check Out Report.....	127
Special Commands.....	130
Tokens.....	131
<b>Chapter 5: Patch Applicator .....</b>	<b>133</b>
Overview of Patch Applicator .....	134
Customization Considerations for All PeopleSoft Versions .....	135
Patches for PeopleSoft Version 8.1.x.....	136
PeopleSoft 8.1.x Patch Applicator Workflow.....	136
Requirements.....	140
Software Requirements .....	140
File Structure Requirements .....	140
Environment Requirements .....	140
Access Requirements .....	141
Workflow Setup Requirements.....	141
Workflow Notification .....	142
Object Type.....	143
Applying Patches.....	144
Patches for PeopleSoft Version 8.4.x.....	146
PeopleSoft 8.4.x Patch Applicator Workflow.....	146

---

Requirements.....	151
Software Requirements .....	151
File Structure Requirements .....	151
Environment Requirements .....	151
Access Requirements .....	151
Workflow Setup Requirements.....	152
Workflow Notification .....	153
Object Type.....	154
Applying Patches.....	155
Patches for PeopleSoft Version 8.44 .....	157
PeopleSoft 8.44 Patch Applicator Workflow .....	157
Requirements.....	162
Software Requirements .....	162
File Structure Requirements .....	162
Environment Requirements .....	162
Access Requirements .....	162
Workflow Setup Requirements.....	163
Workflow Notification .....	164
Object Type.....	165
Applying Patches.....	167
<b>Chapter 6: Version Control.....</b>	<b>169</b>
Overview of Version Control.....	170
Requirements .....	171
Viewing Version Control Repository Contents .....	171
Version Control Menu Items.....	173
Creating a Baseline for Online Objects .....	174
Checking In Objects.....	176
Checking In Objects on the Project Level.....	176
Checking In Objects on the Project Object Level .....	178
Checking In Flat Files.....	180
Checking Out Objects .....	182
Checking Out Objects on the Project Level.....	182
Checking Out Objects on the Project Object Level.....	185
Checking Out Flat Files.....	187
Breaking Locks on Objects.....	189
Breaking Locks on the Project Level.....	189
Breaking Locks on the Project Object Level.....	191
Breaking Locks on Flat Files .....	193
Configuring Users.....	195
Configuring Environments.....	196
Mercury IT Governance Server Database Environment.....	196

PeopleSoft VC Environment.....	197
PeopleSoft Development Environment.....	199
PeopleSoft Version Control Pre-configured Dashboard Page .....	201
Overview of Pages and Portlets.....	201
Adding Pre-configured Dashboard Pages.....	202
PSVC - Object Status Portlet.....	204
PSVC - Locked Objects Summary Portlet.....	205
PSVC - Object Revision Summary Portlet .....	206
PSVC - Object History Portlet.....	207
<b>Index .....</b>	<b>209</b>



## List of Figures

---

Figure 3-1	Environment window, Extension Data tab, PeopleSoft subtab, sample data .....	44
Figure 3-2	Environment window, Host tab, sample data.....	46
Figure 3-3	User window, PeopleSoft subtab .....	51
Figure 4-1	PeopleSoft COBOL Copybook object type sample data.....	60
Figure 4-2	PeopleSoft COBOL DMS object type sample data.....	62
Figure 4-3	PeopleSoft COBOL Program object type sample data.....	63
Figure 4-4	PeopleSoft Data File object type sample data .....	65
Figure 4-5	PeopleSoft Load DMS object type sample data.....	66
Figure 4-6	PeopleSoft Project (ORACLE) object type sample data .....	69
Figure 4-7	PeopleSoft SQC object type sample data.....	71
Figure 4-8	PeopleSoft SQL Script object type sample data.....	72
Figure 4-9	PeopleSoft SQR object type sample data .....	74
Figure 4-10	PeopleSoft Bug request creation.....	77
Figure 4-11	PeopleSoft Enhancement request creation .....	79
Figure 4-12	PeopleSoft Patch Request creation .....	81
Figure 4-13	PeopleSoft Bug/Enhancement Workflow .....	85
Figure 4-14	PeopleSoft Development workflow.....	87
Figure 4-15	PeopleSoft Development Workflow w/ Project Rollback & Object Warning workflow .....	90
Figure 4-16	PeopleSoft Project Object Warning SubWorkflow.....	90
Figure 4-17	PeopleSoft Project Rollback SubWorkflow .....	91
Figure 4-18	PeopleSoft Patch Request workflow.....	96
Figure 4-19	PeopleSoft Compare Custom Database Setup Report .....	100
Figure 4-20	PeopleSoft Comparison Report.....	102

## List of Figures

---

Figure 4-21	PeopleSoft Customized Objects Report.....	104
Figure 4-22	PeopleSoft File Comparison Report.....	106
Figure 4-23	PeopleSoft Object Customization Detail Report.....	108
Figure 4-24	PeopleSoft Object Migrations Report .....	109
Figure 4-25	PeopleSoft Patches Applied to an Environment Report.....	112
Figure 4-26	PeopleSoft Project Migration Impact Analysis Report.....	115
Figure 4-27	PeopleSoft Request Detail Report .....	116
Figure 4-28	PeopleSoft Version Control - Baseline Report .....	119
Figure 4-29	PeopleSoft Version Control - Break Lock Report.....	122
Figure 4-30	PeopleSoft Version Control - Check In Report .....	125
Figure 4-31	PeopleSoft Version Control - Check Out Report .....	128
Figure 5-1	PeopleSoft 8.1.x Patch Applicator workflow .....	137
Figure 5-2	PeopleSoft 8.1.x Patch SubWorkflow .....	137
Figure 5-3	PeopleSoft 8.1.x Patch object type sample data.....	143
Figure 5-4	PeopleSoft 8.4.x Patch Applicator workflow .....	147
Figure 5-5	PeopleSoft 8.4.x Patch SubWorkflow .....	148
Figure 5-6	PeopleSoft 8.4.x Patch object type sample data.....	154
Figure 5-7	PeopleSoft 8.44 Patch Applicator workflow .....	158
Figure 5-8	PeopleSoft 8.44 Patch SubWorkflow.....	159
Figure 5-9	PeopleSoft 8.44 Patch object type sample data .....	165
Figure 6-1	PeopleSoft Version Control pre-configured Dashboard page .....	172
Figure 6-2	Baselining PeopleSoft online objects .....	175
Figure 6-3	Checking in PeopleSoft objects on the project level.....	177
Figure 6-4	Checking in PeopleSoft objects on the project object level.....	179
Figure 6-5	Checking in flat files.....	181
Figure 6-6	Checking out PeopleSoft objects on the project level.....	184
Figure 6-7	Checking out PeopleSoft objects on the project object level.....	186
Figure 6-8	Checking out flat files .....	188
Figure 6-9	Breaking locks on PeopleSoft objects on the project level.....	190
Figure 6-10	Breaking locks on PeopleSoft objects on the project object level.....	192
Figure 6-11	Breaking locks on flat files.....	194
Figure 6-12	PeopleSoft Version Control pre-configured Dashboard page .....	202

## List of Tables

---

Table 2-1	Product name changes in release 5.5.0 .....	30
Table 2-2	Entities that the upgrade revises and renames, or revises, based on usage .....	34
Table 2-3	Entities that the upgrade renames or deletes, based on usage .....	36
Table 2-4	Logs generated during installation.....	42
Table 3-1	Environment window, Extension Data tab, PeopleSoft subtab field definitions....	45
Table 3-2	Environment window required fields for Patch Stage environment.....	47
Table 3-3	Environment window required fields for Development, Test, and Production environments .....	48
Table 3-4	Environment Group window required fields for PeopleSoft Development, Test, and Production environment groups.....	50
Table 3-5	User window, PeopleSoft subtab field definitions.....	51
Table 4-1	Object types included in the Extension.....	58
Table 4-2	PeopleSoft COBOL Copybook object type field definitions .....	61
Table 4-3	PeopleSoft COBOL DMS object type field definitions .....	62
Table 4-4	PeopleSoft COBOL Program object type field definitions.....	64
Table 4-5	PeopleSoft Data File object type field definitions.....	65
Table 4-6	PeopleSoft Load DMS object type field definitions .....	67
Table 4-7	PeopleSoft Project (ORACLE) object type field definitions.....	69
Table 4-8	PeopleSoft SQC object type field definitions.....	71
Table 4-9	PeopleSoft SQL Script object type field definitions .....	72
Table 4-10	PeopleSoft SQR object type field definitions.....	74
Table 4-11	Request types included in the Extension .....	76
Table 4-12	PeopleSoft Bug request type field definitions.....	78

## List of Tables

---

Table 4-13	PeopleSoft Enhancement request type field definitions.....	80
Table 4-14	PeopleSoft Patch Request request type field definitions.....	82
Table 4-15	Workflows included in the Extension.....	84
Table 4-16	PeopleSoft Development Workflow w/ Project Rollback & Object Warning workflow .....	94
Table 4-17	PeopleSoft Project Object Warning SubWorkflow.....	95
Table 4-18	PeopleSoft Project Rollback SubWorkflow .....	95
Table 4-19	Report types included in the Extension .....	99
Table 4-20	PeopleSoft Compare Custom Database Setup Report field definitions.....	101
Table 4-21	PeopleSoft Comparison Report field definitions .....	103
Table 4-22	PeopleSoft Customized Objects Report field definitions .....	105
Table 4-23	PeopleSoft File Comparison Report field definitions .....	106
Table 4-24	PeopleSoft Object Customization Detail Report field definitions.....	108
Table 4-25	PeopleSoft Object Migrations Report field definitions.....	110
Table 4-26	PeopleSoft Patches Applied to an Environment Report field definitions.....	113
Table 4-27	PeopleSoft Project Migration Impact Analysis Report field definitions.....	115
Table 4-28	PeopleSoft Request Detail Report field definitions.....	117
Table 4-29	PeopleSoft Version Control - Baseline Report field definitions.....	120
Table 4-30	PeopleSoft Version Control - Break Lock Report field definitions.....	123
Table 4-31	PeopleSoft Version Control - Check In Report field definitions.....	126
Table 4-32	PeopleSoft Version Control - Check Out Report field definitions.....	129
Table 4-33	Special commands included in the Extension .....	130
Table 4-34	Tokens included in the Extension.....	132
Table 5-1	PeopleSoft 8.1.x Patch Applicator workflow setup.....	141
Table 5-2	PeopleSoft 8.1.x Patch SubWorkflow setup.....	142
Table 5-3	PeopleSoft 8.1.x Patch object type field definitions .....	143
Table 5-4	PeopleSoft 8.4.x Patch Applicator workflow setup.....	152
Table 5-5	PeopleSoft 8.4.x Patch SubWorkflow setup.....	153
Table 5-6	PeopleSoft 8.4.x Patch object type field definitions .....	154
Table 5-7	PeopleSoft 8.44 Patch Applicator workflow setup .....	163
Table 5-8	PeopleSoft 8.44 Patch SubWorkflow setup .....	164
Table 5-9	PeopleSoft 8.44 Patch object type field definitions.....	166
Table 6-1	Menu items in PeopleSoft menu, Version Control submenu .....	173
Table 6-2	User information for Version Control.....	195

Table 6-3	Environment window required fields for ITG_SERVER_DATABASE.....	196
Table 6-4	Environment window required fields for PeopleSoft VC environment.....	197
Table 6-5	Environment window required fields for PeopleSoft Development environment.....	199
Table 6-6	PSVC - Object Status portlet displayed fields.....	204
Table 6-7	PSVC - Object Status portlet filter fields.....	204
Table 6-8	PSVC - Locked Objects Summary portlet displayed fields.....	205
Table 6-9	PSVC - Locked Object Summary portlet filter fields.....	205
Table 6-10	PSVC - Object Revision Summary portlet displayed fields.....	206
Table 6-11	PSVC - Object Revision Summary portlet filter fields.....	206
Table 6-12	PSVC - Object History portlet displayed fields.....	207
Table 6-13	PSVC - Object History portlet filter fields.....	207



# Chapter 1 Introduction

---

## In This Chapter:

- *About This Document*
  - *Who Should Read This Document*
  - *Prerequisite Documents*
  - *Related Documents*
  - *Overview of the Extension*
    - *General Overview of Extension for PeopleSoft Enterprise*
    - *What's New in Release 6.0.0*
    - *Prerequisite Knowledge and Experience*
-

## About This Document

This document provides information about installing or upgrading to release 6.0.0 of Mercury Change Management Extension for PeopleSoft Enterprise™ (often referred to hereafter as “the Extension”), and it provides conceptual, procedural, and reference information about the product.

You can access this document from the Mercury IT Governance Center™ Documentation home page and elsewhere, as described in the *Guide to Documentation*.

This document is organized as follows:

- [Chapter 1, \*Introduction\*, on page 15](#) (this chapter) describes who should read this document, includes information about prerequisite and related documents, and provides an overview of the Extension for this release.
- [Chapter 2, \*Installing or Upgrading the Extension\*, on page 25](#) provides overview and detailed information about installing and upgrading the Extension.
- [Chapter 3, \*Configuring the Extension\*, on page 43](#) provides information related to defining the Extension environments and users, and mass password change functionality.
- [Chapter 4, \*Extension Entities\*, on page 55](#) provides information about the object types, request types, workflows, report types, special commands, and tokens that are unique to the Extension.
- [Chapter 5, \*Patch Applicator\*, on page 133](#) provides guidance, procedural, and reference information about PeopleSoft Patch Applicator, which is part of the Extension.
- [Chapter 6, \*Version Control\*, on page 169](#) provides guidance, procedural, and reference information about PeopleSoft Version Control, which is part of the Extension.

## Who Should Read This Document

This document is intended for the following audiences:

- People responsible for installing, upgrading, configuring, and maintaining the system and database environments where Mercury Change Management Extension for PeopleSoft Enterprise resides, and setting up and maintaining the Mercury IT Governance schema
- People responsible for maintaining access and security for, or supporting use of, Mercury Change Management Extension for PeopleSoft Enterprise

### For More Information

For more information about audience types as defined for Mercury IT Governance Center, see the *Guide to Documentation*.

## Prerequisite Documents

You should be familiar with the information in the following documents:

- *What's New in Release 6.0*
- *Release Notes*
- *Key Concepts*
- *Getting Started*
- *System Administration Guide and Reference*
- *Mercury Change Management: Configuring a Deployment System*
- *Mercury Change Management User's Guide*
- *Security Model Guide and Reference*

Before performing an installation or upgrade, check the latest version of the *Release Notes* for release 6.0 on the Mercury IT Governance Download Center for any last-minute information or issues you need to understand.

## Related Documents

See the *Guide to Documentation* for:

- Descriptions of all of the Mercury documents for Mercury IT Governance Center and Mercury Change Management Extensions, and how to access each document
- Definitions of audience types, reflecting the roles played by various personnel in using Mercury IT Governance Center and Mercury Change Management Extensions
- Descriptions of typographical conventions used in the documentation set
- Additional resources for support in using Mercury IT Governance Center and Mercury Change Management Extensions

## Overview of the Extension

Mercury offers prepackaged change management products to automate migrations, upgrades, and other changes in a variety of leading database, system, and application environments. These products are called Mercury Change Management Extensions.

Extensions are installed on the Mercury IT Governance Server.

### General Overview of Extension for PeopleSoft Enterprise

Mercury Change Management Extension for PeopleSoft Enterprise helps to automate change management in the PeopleSoft environment.

The Extension enhances the functionality of Mercury Change Management™ by providing predefined entities that are unique to PeopleSoft Enterprise environments, such as specialized object types, request types, workflows, report types, commands, and tokens.

This manual documents release 6.0.0 of Mercury Change Management Extension for PeopleSoft Enterprise.

Before using Mercury IT Governance Center release 6.0 or the Extension at release 6.0.0 for productive work, you must do one of the following:

- Install Mercury IT Governance Center release 6.0, then install Extension release 6.0.0
- Upgrade Mercury IT Governance Center to release 6.0, then upgrade the Extension to release 6.0.0

## What's New in Release 6.0.0

Release 6.0.0 of the Extension introduces the Extension functionality described in the following sections.

### For More Information

For more information about Mercury IT Governance Center release 6.0, see the *What's New in Release 6.0* document.

For descriptions of supported upgrade paths, see [Installing or Upgrading to Release 6.0.0 on page 29](#). For general information about the impacts of upgrading from any release to release 6.0.0, see [General Upgrade Impacts and Guidelines on page 29](#). For specific information about the impacts of upgrading from release 5.0.0, see [Impacts of Upgrading from Release 5.0.0 on page 30](#). For specific information about the impacts of upgrading from release 5.5.0, see [Impacts of Upgrading from Release 5.5.0 on page 31](#).

## Support for PeopleSoft Version 8.44

Release 6.0.0 of the Extension adds support for PeopleSoft Version 8.44, including the following new entities for the Version 8.44's new patch application process:

- PeopleSoft 8.44 Patch object type
- PeopleSoft 8.44 Patch Applicator workflow
- PeopleSoft 8.44 Patch SubWorkflow

All Extension functionality that has supported PeopleSoft Versions 8.1.x and 8.4.x now supports Version 8.44 as well.

### For More Information

For more information, see [Patches for PeopleSoft Version 8.44 on page 157](#).

## ***Desupport of PeopleSoft Version 7.5 and Microsoft SQL Server***

All functionality and content related to PeopleSoft Version 7.5 and SQL Server are removed in release 6.0.0 of the Extension.

### **For More Information**

For more information, see [Desupport of PeopleSoft Version 7.5 and SQL Server](#) on page 32 in the discussion of upgrade impacts.

## ***New PeopleSoft Comparison Report***

Release 6.0.0 of the Extension introduces the PeopleSoft Comparison Report, which is used to compare PeopleSoft instances.

### **For More Information**

For more information about this report, see [PeopleSoft Comparison Report](#) on page 102.

## ***Enhancements to PeopleSoft Project Rollback Subworkflow***

The PeopleSoft Project Rollback subworkflow has been enhanced to support PeopleSoft Version 8.44, to provide rollback at the object level (not only at the project level), and to automatically delete new projects during rollback.

### **For More Information**

For more information, see [PeopleSoft Development Workflow w/ Project Rollback & Object Warning Workflow](#) on page 89.

## ***Enhancements to PeopleSoft Project (ORACLE) Object Type***

The PeopleSoft Project (ORACLE) object type has been enhanced with new fields to support the enhancement to the PeopleSoft Project Rollback subworkflow and to manage multiple project build scripts as well as a single script.

### **For More Information**

For more information, see [PeopleSoft Project \(ORACLE\) Object Type](#) on page 68.

## ***Pre-configured Dashboard Pages***

As with Mercury IT Governance Center release 6.0, Extension-related pre-configured Dashboard pages in the Mercury IT Governance Dashboard™ replace the tab templates used in earlier releases and provide the same functionality.

## ***Reports Accessible Only from Standard Interface***

As with Mercury IT Governance Center release 6.0, Extension-related reports are accessible only from the standard interface. As a result, the option to create a baseline of PeopleSoft online objects by running a report has been added to the **PeopleSoft > Version Control** menu in the standard interface. Each of the Version Control menu items invokes a different report.

### **For More Information**

For more information about the Version Control menu items, see [Version Control Menu Items on page 173](#).

## ***Licensing***

Extension licensing has been updated, consistent with the new Mercury IT Governance Center release 6.0 licensing strategy. For more information, see the *Security Model Guide and Reference*.

## Prerequisite Knowledge and Experience

To install, upgrade, configure, maintain, or use release 6.0.0 of the Extension, you need to understand the following:

- Particular key concepts described in the *Key Concepts* document:
  - Change management
  - Environments
  - Mercury IT Governance Workbench
  - Object types
  - Packages
  - Portlets
  - Software deployment
  - Tokens
  - Workflows and workflow steps
  - Request types
  - Report types
- The Mercury IT Governance Center entities installed by the Extension

In addition, you must have practical experience installing, upgrading, configuring, and using Mercury IT Governance Center, if you are responsible for configuring the Extension.

You should have experience using PeopleSoft Enterprise applications.

### For More Information

For more information about documents and resources that help you gain the required knowledge and experience, see the *Guide to Documentation*.



## **Installing or Upgrading the Extension**

---

### **In This Chapter:**

- *Overview of Installation*
    - *System Requirements*
    - *Installing or Upgrading to Release 6.0.0*
    - *General Upgrade Impacts and Guidelines*
    - *Impacts of Upgrading from Release 5.0.0*
    - *Impacts of Upgrading from Release 5.5.0*
  - *Preparing for Installation*
  - *Downloading and Installing the Extension*
    - *Downloading the Installation Files*
    - *Installation Procedure*
    - *Logs Generated During Installation*
  - *Post-Installation Procedures*
-

## Overview of Installation

This section discusses the following overview subjects:

- System requirements
- Known issues
- Installing or upgrading to release 6.0.0
- General upgrade impacts and guidelines
- Impacts of upgrading from release 5.0.0
- Impacts of upgrading from release 5.5.0

In addition, this chapter discusses the following:

- Preparing for installation
- Downloading and installing the Extension
- Post-installation procedures

## System Requirements

Mercury IT Governance Center release 6.0 and Mercury Change Management Extension for PeopleSoft Enterprise release 6.0.0 are installed on the same system and have the same system requirements. For more information about version level and other general system requirements, see the *System Requirements and Compatibility Matrix* document, which is available from the Mercury IT Governance Download Center:

<http://itg.merc-int.com/support/download/login.jsp>

The following specific system requirements must also be met:

- PeopleSoft endorsed COBOL Compiler (recommended to be installed on the same server as PeopleSoft):
  - Net Express for NT
  - Server Express for UNIX
- The file server on which PeopleSoft is installed should have PeopleSoft clients installed on it for each database instance that will be used in the migrations.

- Identical file storage directory structure (i.e. cblbin, scripts, sqr) between migration environments.
  - `src/cbl/base` for COBOL source files on Windows
  - `src/cbl/base/compile` for COBOL compiler on Windows
  - `src/cbl/base` for COBOL source files on UNIX
  - PeopleSoft Version 8.1.x: `install` for COBOL compiler on UNIX
  - PeopleSoft Version 8.4.x or 8.44: `setup` for COBOL compiler on UNIX
  - `cblbin` for COBOL executables on Windows and UNIX
  - `sqr` for SQR and SQC report files on Windows and UNIX
  - `crw/eng` for Crystal Reports on Windows
  - `nvision/layout` for nVision Layouts on Windows
  - `scripts` for Data Mover Script (DMS) and SQL files on Windows
  - `data` for Data Mover Script (DMS) files on Windows
- For Oracle databases, database links from the IT Governance schema to the PeopleSoft databases should be created.
- BRIO SQR: Copy `pssqr.ini` into the `drive:\winnt` directory (for example, `C:\winnt`) and rename that copy `sqr.ini`.

- For each instance of PeopleSoft:

- The following directory must exist for NT servers:

```
\\src\cbl\base\compile\
```

- The file CBLMAKE.BAT must reside in each instance of the `\compile\` directory. This file should be a copy of the batch file located in the `PSOFT_HOME\src\cbl\win32` directory.

- The following lines should be added to the CBLMAKE.BAT file as the first command after the `rem` comments:

```
set PS_HOME= base directory for PeopleSoft instance
set COBROOT= base directory of COBOL compiler
```

The COBROOT base directory is probably `drive:\netexpress\base` (for example, `C:\netexpress\base`).

- Create a backup directory under `$PS_Home`.

Make sure that you complete all the activities in this bullet for each instance of PeopleSoft.

- To run the PeopleSoft Comparison Report introduced in release 6.0.0, you must change the Windows registry of the file server where the report will be generated. Create a DWORD value called `UpgHtmlOutput` and set the value to 1 under the registry key:

```
HKEY_CURRENT_USER\Software\PeopleSoft\PeopleTools\
Release8.40\PSIDE
```

This allows generation of an HTML version of the report, with a naming convention of `Upgxxobjecttype.htm` (for example, `Upg02Fields.htm`).

## Installing or Upgrading to Release 6.0.0

Before installing or upgrading the Extension, you must install (or upgrade to) and configure Mercury IT Governance Center release 6.0. For information about those procedures, see the *System Administration Guide and Reference*.

After you have installed Mercury IT Governance Center release 6.0, you can install Mercury Change Management Extension for PeopleSoft Enterprise release 6.0.0 or upgrade to release 6.0.0 from either of the following releases:

- Mercury Change Management Extension for PeopleSoft Enterprise release 5.5.0
- Kintana™ Accelerator for PeopleSoft Enterprise release 5.0.0

The Mercury IT Governance Center release 6.0 upgrade process detects the existence of either release of the Extension in your environment and advises you that you will need to upgrade the Extension. This chapter provides information about upgrading the Extension from either release 5.5.0 or release 5.0.0.

If you are upgrading the Extension from any release earlier than Kintana Accelerator for PeopleSoft Enterprise release 5.0.0, you must first upgrade it to release 5.0.0, then upgrade it to release 6.0.0. The Mercury IT Governance Center upgrade process enforces this rule. For information about upgrades to (and features of) releases earlier than release 5.0.0, see the documentation for all of the releases from your current product release through release 5.0.0 as necessary.

Upgrade the Extension before using Mercury IT Governance Center release 6.0 for productive work.

## General Upgrade Impacts and Guidelines

The Extension upgrade process replaces code entities installed in earlier releases, including all reference (REFERENCE) versions of the Extension entities described in [Chapter 4, \*Extension Entities\*, on page 55](#).

During the upgrade process, existing non-reference entities, including those that you have copied, renamed, or customized for your specific environment, are not modified in any way. In other words, these non-reference entities retain all your customizations and they do not reflect any changes made in release 6.0.0, so they might not be compatible with Mercury IT Governance Center release 6.0.

Therefore, after upgrading to release 6.0.0, you should review the new reference entities. If necessary, copy and rename them, then apply the same customizations that were previously made to the corresponding non-reference entities for the earlier release.

Dashboard tab template files are replaced by corresponding pre-configured Dashboard pages during upgrade. Menu XML files are backed up during upgrade.

## Impacts of Upgrading from Release 5.0.0



Note

If you are upgrading to release 6.0.0 from release 5.5.0, go to the section [Impacts of Upgrading from Release 5.5.0 on page 31](#).

If you are upgrading from release 5.0.0 directly to release 6.0.0, without ever installing release 5.5.0, you need to review both of the following sections:

- This section, which discusses the impacts of an upgrade from release 5.0.0 to release 5.5.0.
- The section [Impacts of Upgrading from Release 5.5.0 on page 31](#), which discusses the impacts of an upgrade from release 5.5.0 to release 6.0.0.

The only significant impact seen in the Extension for an upgrade from release 5.0.0 to release 5.5.0 is that some product names changed as shown in [Table 2-1](#).

*Table 2-1. Product name changes in release 5.5.0*

Product Name in Release 5.0.0	Product Name in Release 5.5.0
Kintana Accelerator	Mercury Change Management Extension
Kintana Deliver	Mercury Change Management
Kintana Drive	Mercury Project Management™
Kintana Create	Mercury Demand Management™

## Impacts of Upgrading from Release 5.5.0

The following sections discuss the following impacts of the upgrade from release 5.5.0 to release 6.0.0:

- Dashboard menus and tab templates (pre-configured Dashboard pages)
- Desupport of PeopleSoft Version 7.5 and SQL Server
- New PeopleSoft Comparison Report

### *Dashboard Menus and Tab Templates (Pre-configured Dashboard Pages)*



Note

Tab templates, as they were called prior to release 6.0.0, are called pre-configured Dashboard pages or modules in release 6.0.0. See the *Configuring the Standard Interface* document for information about how to configure them.

The Extension installation procedure makes backup copies of your existing menu and tab XML files in the format `file_name.PeopleSoft`, for example `PeopleSoft_menu.xml.PeopleSoft`.

The provided file is `ITG_Home/conf/menus/PeopleSoft_menu.xml`.

If you do not currently use the Extension menus and tab templates in a release earlier than release 6.0.0 and you do not plan to use them immediately, move the files to `ITG_Home/deploy/600/PeopleSoft` for storage until you need them.

If you have been using the Extension menus in release 5.0.0 or release 5.5.0, you need to make some changes for your existing menus to function correctly with release 6.0.0. The Access Type and License Type security entries in the menus should be modified to use the release 6.0.0 values. The menus delivered with the 6.0.0 Extension can be used as a model.

For additional information on customizing menus and pre-configured Dashboard pages, see the *Customizing the Standard Interface* document.

## ***Desupport of PeopleSoft Version 7.5 and SQL Server***

In general, an upgrade process reinstalls the latest version of all reference (REFERENCE) entities. However, because release 6.0.0 of the Extension no longer supports PeopleSoft Version 7.5 functionality (including AUDB, AU DMS, and Application Upgrade) or SQL Server functionality, this Extension's upgrade process has different consequences for certain reference entities, as described in the following sections:

- Reference entities that the upgrade deletes
- Reference entities that the upgrade revises
- Reference entities that the upgrade revises and renames, or revises, based on usage
- Reference entities that the upgrade renames or deletes, based on usage



Note

Remember that the upgrade process always leaves all previously established non-reference entities unchanged, so your use of them should not be affected by the upgrade.

### ***Reference Entities That the Upgrade Deletes***

The upgrade to release 6.0.0 deletes and does not replace the following reference entities that are associated with PeopleSoft Version 7.5 and SQL Server, therefore they become unavailable:

- Object types
  - (REFERENCE) PeopleSoft Project (MSSQL)
  - (REFERENCE) PeopleSoft AU DMS
  - (REFERENCE) PeopleSoft 7.5.x Patch Object Type
- Workflow
  - (REFERENCE) PeopleSoft 7.5.x Patch Applicator

### ***Reference Entities That the Upgrade Revises***

In release 6.0.0, the following reference entities have been revised to exclude functionality related to PeopleSoft Version 7.5.x and SQL Server:

- Object types
  - (REFERENCE) PeopleSoft 8.1.x Patch Object Type
  - (REFERENCE) PeopleSoft 8.4.x Patch Object Type
  - (REFERENCE) PeopleSoft COBOL DMS
  - (REFERENCE) PeopleSoft Load DMS
  - (REFERENCE) PeopleSoft Project (ORACLE)
  - (REFERENCE) PeopleSoft SQL Script
  - (REFERENCE) PeopleSoft SQR
- Workflows
  - (REFERENCE) PeopleSoft 8.1.x Patch Applicator
  - (REFERENCE) PeopleSoft 8.1.x Patch SubWorkflow
  - (REFERENCE) PeopleSoft 8.4.x Patch Applicator
  - (REFERENCE) PeopleSoft 8.4.x Patch SubWorkflow
  - (REFERENCE) PeopleSoft Project Rollback SubWorkflow
- Report types
  - (REFERENCE) PeopleSoft Version Control - Baseline
  - (REFERENCE) PeopleSoft Version Control - Break Lock
  - (REFERENCE) PeopleSoft Version Control - Check In
  - (REFERENCE) PeopleSoft Version Control - Check Out

**Reference Entities That the Upgrade Revises and Renames, or Revises, Based on Usage**

During the upgrade to release 6.0.0, the particular previously installed reference entities in the left column of *Table 2-2* that are being used by existing non-reference entities in your current installation are:

- Revised to remove PeopleSoft Version 7.5 and SQL Server functionality.
- Renamed as shown in the right column of *Table 2-2*, using the prefix DEP (or, in the case of special commands, the first characters of the command names are changed from `ksc` to `sc_dep`) to indicate that these entities are deprecated. The operation of existing non-reference entities that use the renamed entities should not be affected.
- Replaced with new reference versions of the same name (in the left column of *Table 2-2*), having the same revisions.

Reference entities in the left column of *Table 2-2* that are not being used by existing non-reference entities in your current installation prior to the upgrade are simply replaced with revised versions of the same names.

In the unlikely event that you already have entities with any of the new names in the right column of *Table 2-2*, rename those entities before you perform the upgrade.

*Table 2-2. Entities that the upgrade revises and renames, or revises, based on usage*

Reference Entity Name Prior to Upgrade	Reference Entity Name After Upgrade, If Entity Was Used Before Upgrade
<b>Workflow Step Sources</b>	
(R) PS - Check New Project	DEP PS - Check New Project
(R) PS - Copy Project	DEP PS - Copy Project
(R) PS - Distribute Files	DEP PS - Distribute Files
(R) PS - Import Project to Demo	DEP PS - Import Project to Demo
(R) PS - List Patch Files	DEP PS - List Patch Files
(R) PS - Unlock Demo	DEP PS - Unlock Demo
<b>Validations</b>	
PS - Projects in Development Environment	DEP PS - Projects in Development Environment
PS - Project Objects in Development Environment	DEP PS - Project Objects in Development Environment

Table 2-2. Entities that the upgrade revises and renames, or revises, based on usage

Reference Entity Name Prior to Upgrade	Reference Entity Name After Upgrade, If Entity Was Used Before Upgrade
<b>Special Commands</b>	
ksc_psft_check_for_locked_objects	sc_dep_psft_check_for_locked_objects
ksc_psft_list_objects	sc_dep_psft_list_objects
ksc_psft_lock_objects	sc_dep_psft_lock_objects
ksc_psft_unlock_objects	sc_dep_psft_unlock_objects
<b>Shell Scripts</b>	
kacc_psft84_project_export.sh	DEP_kacc_psft84_project_export.sh
kacc_psft84_project_import.sh	DEP_kacc_psft84_project_import.sh
kacc_psft_check_lock.sh	DEP_kacc_psft_check_lock.sh
kacc_psft_lock.sh	DEP_kacc_psft_lock.sh
kacc_psft_unlock.sh	DEP_kacc_psft_unlock.sh

### *Reference Entities That the Upgrade Renames or Deletes, Based on Usage*

The upgrade process renames (but does not otherwise change) the particular previously installed reference entities in the left column of [Table 2-3](#) that are being used by existing non-reference entities in your current installation.

As shown in the right column of [Table 2-3](#), the renaming adds the prefix DEP (or, in the case of the special command, it changes the first characters of the command name from `ksc` to `sc_dep`) to indicate that these entities are deprecated. The operation of existing non-reference entities that use the renamed entities should not be affected.

Reference entities in the left column of [Table 2-3](#) that are not being used by existing non-reference entities in your current installation prior to the upgrade are deleted (not replaced) in release 6.0.0.

In the unlikely event that you already have entities with any of the new names in the right column of [Table 2-3](#), rename those entities before you perform the upgrade.

Table 2-3. Entities that the upgrade renames or deletes, based on usage

Reference Entity Name Prior to Upgrade	Reference Entity Name After Upgrade, If Entity Was Used Before Upgrade
<b>Subworkflow</b>	
(REFERENCE) PeopleSoft 7.5.x Patch SubWorkflow	DEP PeopleSoft 7.5.x Patch SubWorkflow
<b>Workflow Step Sources</b>	
(R) PS - AUDMS Exist	DEP PS - AUDMS Exist
(R) PS - Import Project to AADB	DEP PS - Import Project to AADB
(R) PS - Unlock AADB	DEP PS - Unlock AADB
<b>Validations</b>	
PS - PeopleSoft AU Environments	DEP PS - PeopleSoft AU Environments
PS - Projects (MSSQL)	DEP PS - Projects (MSSQL)
<b>Special Command</b>	
ksc_psft_run_sql_mssql	sc_dep_psft_run_sql_mssql
<b>Shell Scripts</b>	
kacc_psft75_project_export.sh	DEP_kacc_psft75_project_export.sh
kacc_psft75_project_import.sh	DEP_kacc_psft75_project_import.sh

### ***New PeopleSoft Comparison Report***

For an upgrade to release 6.0.0 (or for a new installation), using the PeopleSoft Comparison Report introduced in release 6.0.0 requires changing the Windows registry of the file server where the report will be generated. See [System Requirements on page 26](#).

#### **For More Information**

For more information about this report, see [PeopleSoft Comparison Report on page 102](#).

## Preparing for Installation

To prepare for installation of the Extension:

1. Purchase the Extension, if you have not already done so.

You will not be allowed to download the Extension installation files from the Mercury IT Governance Download Center unless you have the username and password given to you at the time of purchase.

2. Decide where to put the installation file (`mitg-600-PeopleSoft.jar`).

The installation script looks in one of these locations for this file:

- `ITG_Home` directory (this is the recommended location)
  - If the `ITG_DEPLOYMENT_HOME` environment variable is set: `ITG_Home/Deployment_Home` directory
  - If you use the `-D` option on the `sh` command to start the installation procedure: `Deployment_Home` directory
3. Collect the following information, which you will need to supply during the installation procedure:
    - The username and password you were given at the time you purchased the Extension.
    - The database password for the Mercury IT Governance Center schema
    - A Mercury IT Governance Center logon username

The username (typically “admin”) must belong to a security group that has the following access grants:

- SysAdmin: Migrate ITG Objects
  - SysAdmin: Server Administrator
- The password for the Mercury IT Governance Center logon username
4. Stop the Mercury IT Governance Server.

5. Set the Mercury IT Governance Server to restricted mode by running the following script:

```
setServerMode.sh Restricted
```

For more information about this script, see the *System Administration Guide and Reference*.

6. Start the Mercury IT Governance Server.

## Downloading and Installing the Extension

This section describes the download and installation procedures to install or upgrade the Extension.

### Downloading the Installation Files

To download the installation files for the Extension:

1. Go to the Mercury IT Governance Download Center:

<http://itg.merc-int.com/support/download/login.jsp>

2. Log on using the username and password your company was given when it purchased the Extension.
3. In the ITG Download > License window:
  - a. Read the Software End User License Agreement.
  - b. Select the **I accept these terms** radio button.
  - c. Click **Submit**.
4. In the ITG Download > Main window, click the **Installs** link under the name.
5. To download the Extension, click **mitg-600-PeopleSoft.jar**.

## Installation Procedure

To install the Extension:

1. Be sure you have completed all the steps in [Preparing for Installation on page 37](#).

In particular, be sure Mercury IT Governance Center is running in restricted mode. This procedure is documented in [step 5 on page 38](#).

2. Be sure you have completed all the steps in [Downloading the Installation Files on page 38](#).

3. Copy the `mitg-600-PeopleSoft.jar` file to the `ITG_Home` directory.

For information about using another location for this file, see [step 2 on page 37](#).



### Note

You do not need to unpack the installation file. The installation process automatically unpacks it.

4. Navigate to the `ITG_Home/bin` directory.

5. Run the script:

```
sh kDeploy.sh -i PeopleSoft
```



### Note

The `kDeploy` script can upgrade one Extension. For more information, see the *System Administration Guide and Reference*.

If you have upgraded Mercury IT Governance Center to release 6.0 and if one or more Mercury Change Management Extensions were previously installed on your server at the release 5.0.0 or 5.5.0 level, you can upgrade multiple Extensions by running one script. To do so, download one `.jar` file for each Extension to the `ITG_Home` directory, and run the `mitg_extensions.sh` upgrade script from the `upgrade_600` directory.

6. As `kDeploy.sh` runs, respond to prompts for the information you collected in [step 3 on page 37](#) (for example, the database password for the Mercury IT Governance Center schema, and the Mercury IT Governance Center logon name and password).

Files are installed in various subdirectories under `ITG_Home`. Data is also placed in the Mercury IT Governance database.

When the installation procedure is complete, the following message appears:

```
Deployment PeopleSoft has been successfully installed.
```

7. Using a Web browser, view and check the installation summary report, which is located in:

```
ITG_Home/logs/deploy/600/PeopleSoft/log_x/installLog.html
```

where `log_x` is a random number generated by `kDeploy.sh` to make each log file name unique. The number increments by one each time the installation script is run, so the log file for the most recent run is the one with the highest log number.

The summary report lists all Mercury entities installed as part of the Extension installation process.

Each entity that was installed correctly is marked as “Completed.” If there is an error for a particular entity, the report contains a direct link to another log file (HTML page) with additional information.

Installation of the Extension generates the logs that are listed and described in [Table 2-4](#).

Correct the errors, if necessary, and repeat the installation procedure. Otherwise, proceed to [step 8](#).

8. (Optional) To verify that Mercury Change Management Extension for PeopleSoft Enterprise is listed among the installed release 6.0.0 Mercury Extensions, issue the following command:

```
sh kDeploy.sh -l
```

The name `PeopleSoft` should appear in the list of installed Mercury Extensions.

For example, if you have both Mercury Change Management Extension for PeopleSoft Enterprise and Mercury Change Management Extension for E-Commerce Technologies™ installed, you could see a table similar to the following:

Deployment	Version	Deployed	Description
PeopleSoft	600	2005-02-15	PeopleSoft Extension
eCommerce	600	2005-03-22	eCommerce Extension

9. (Optional) Install other Mercury Extensions you have purchased, if you have not already done so.

For specific installation procedures for the other Mercury Extensions, see the appropriate Extension document described in the *Guide to Documentation*.

10. Stop and restart the Mercury IT Governance Center in normal mode, as follows.

- a. Stop the Mercury IT Governance Server.

- b. Run the script:

```
setServerMode.sh NORMAL
```

For more information about this script, see the *System Administration Guide and Reference*.

- c. Start the Mercury IT Governance Server.

## Logs Generated During Installation

The logs listed and described in [Table 2-4](#) are generated during installation and saved in the `ITG_Home/logs/deploy/600/PeopleSoft` directory. The log number (#####) is a random number generated by `kDeploy.sh` to make each log file name unique.

Table 2-4. Logs generated during installation

Directory or File	Description
<code>ddlDriver.#####.log</code>	Contains information about data model changes made during installation.
<code>jarxvf.#####.log</code>	Contains information from the procedure that unpacks the <code>jar</code> file.
<code>packageDriver.#####.log</code>	Contains information about the installation of database code, for example, reports.
<code>postXMLDriver.#####.log</code>	Contains information about the application of SQL scripts required after the installation of Mercury-supplied data.
<code>preXMLDriver.#####.log</code>	Contains information about the application of SQL scripts required prior to the installation of Mercury-supplied data, such as the definition for Mercury Change Management Extension for PeopleSoft Enterprise.

## Post-Installation Procedures

After you have finished installing Mercury Change Management Extension for PeopleSoft Enterprise, you need to:

- Review the object types
- Define environments for each application environment that will be a source or destination of migration, for example:
  - Host connection information
  - Database connection information
- If required, customize the logic in the various installed entities.

After the Extension has been running successfully for a substantial period of time, you can optionally delete all of the installation or upgrade files. However, Mercury recommends that you retain (or copy) the log files.

## Chapter

# 3

## Configuring the Extension

---

### In This Chapter:

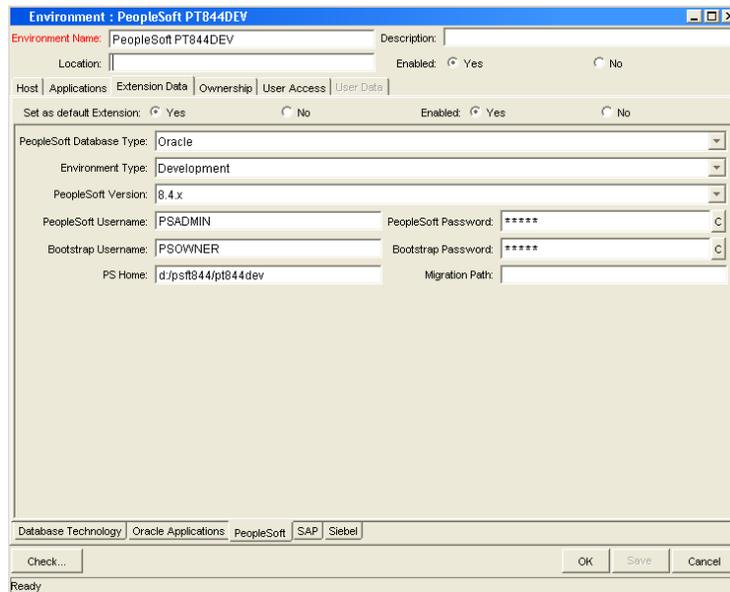
- *Configuring Environments*
    - *Configuring PeopleSoft Patch Stage*
    - *Configuring Development/Test/Production Instances of PeopleSoft*
    - *Configuring Development/Test/Production Environment Groups*
  - *Configuring PeopleSoft Users*
  - *Mass Password Change Functionality*
-



Because the menu items and the pre-configured Dashboard page that are unique to this Extension are related to version control functionality, they are documented in [Chapter 6, \*Version Control\*, on page 169](#).

## Configuring Environments

The environment for Mercury Change Management Extension for PeopleSoft Enterprise is specified in the Mercury IT Governance Workbench Environment window, **Extension Data** tab, **PeopleSoft** subtab and in the Environment window **Host** tab. [Figure 3-1](#) shows a sample **Extension Data** tab, **PeopleSoft** subtab. [Table 3-1](#) lists the field names and their definitions. [Figure 3-2 on page 46](#) shows a sample **Host** tab.



*Figure 3-1. Environment window, Extension Data tab, PeopleSoft subtab, sample data*

Table 3-1. Environment window, Extension Data tab, PeopleSoft subtab field definitions

Field Name	Definition
Set as default Extension	Whether or not <b>PeopleSoft</b> is the default subtab to show for this window's <b>Extension Data</b> tab.
Enabled	Whether or not the fields on this tab are available to be configured.
PeopleSoft Database Type	Database type used in the PeopleSoft implementation.
Environment Type	Environment type: <b>Demo, Development, Patch Stage, Production, Test,</b> or <b>Version Control.</b>
PeopleSoft Version	PeopleSoft version: <b>8.1.x</b> or <b>8.4.x.</b>
PeopleSoft Username	PeopleSoft operator ID. For more information, see the second Note below.
PeopleSoft Password	PeopleSoft operator ID password.
Bootstrap Username	System administrator operator ID. For more information, see the second Note below.
Bootstrap Password	System administrator password.
PS Home	PS Home directory for this environment.
Migration Path	Alternate path to the base path for Connect and Copy commands in cases where the path for the files is accessed through a mapped drive instead of through TELNET or SSH.



Note

Server logins to be used on the Extension for PeopleSoft should have read, write, and execute privileges to all PeopleSoft drives.



Note

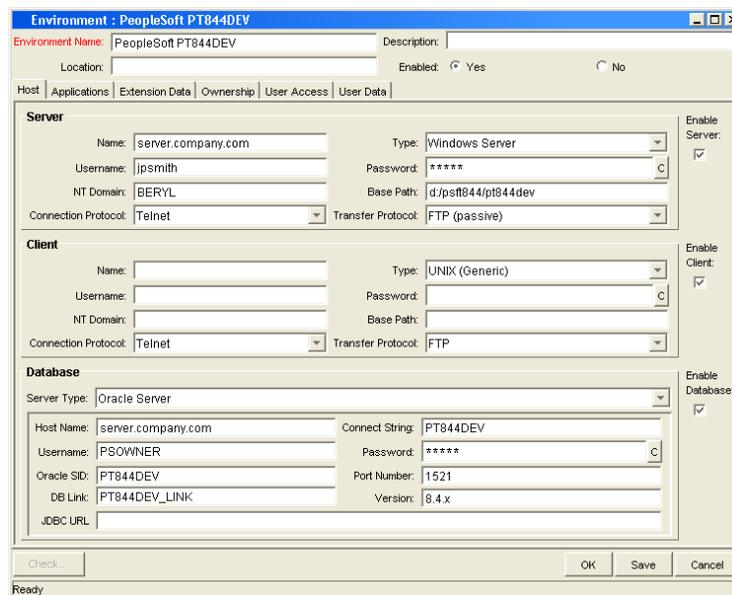
The PeopleSoft Username and Bootstrap Username must have access to the following files:

- sqlplus.exe
- pstools.exe
- psdmt.exe

To access the **PeopleSoft** subtab on the **Extension Data** tab in the Workbench Environments window:

1. Open the Workbench.
2. From the shortcut bar, click the **Environments** button and click the **Environments** icon.
3. Open a new environment or, if the environment of interest already exists, open it. The Environment window opens.
4. Click the **Extension Data** tab.
5. At the bottom of the window, click the **PeopleSoft** subtab.

*Figure 3-2* shows the Environment window **Host** tab with sample data.



*Figure 3-2. Environment window, Host tab, sample data*

The following sections discuss the following items that must be configured for the Extension to function correctly:

- PeopleSoft Patch Stage
- Development/Test/Production instances of PeopleSoft
- Development/Test/Production environment groups

## Configuring PeopleSoft Patch Stage

PeopleSoft Patch Stage is the environment where patch files are stored. Define this environment as shown in [Table 3-2](#).

*Table 3-2. Environment window required fields for Patch Stage environment*

Field Name	Value
<b>Host tab, Server section</b>	
Enable Server	Enabled (select this checkbox).
Name	Name of server where the patch stage resides.
Type	<b>Windows Server</b> or <b>UNIX (Generic)</b> .
User Name	User name.
Password	User password.
NT Domain	Domain name if server type is <b>Windows Server</b> .
Base Path	Base path for patch stage.
Connection Protocol	Connection protocol.
Transfer Protocol	Transfer protocol.
<b>Host tab, Client section</b>	
Enable Client	Disabled (deselect this checkbox).
<b>Host tab, Database section</b>	
Enable Database	Disabled (deselect this checkbox).
<b>Extension Data tab, PeopleSoft subtab</b>	
Set as default Extension	<b>Yes.</b>
Enabled	<b>Yes.</b>
Environment Type	<b>Patch Stage.</b>
PeopleSoft Version	<b>8.1.x</b> or <b>8.4.x.</b>

## Configuring Development/Test/Production Instances of PeopleSoft

If the PeopleSoft setup has the file server on a different machine than the batch server, then environments for the file server and batch server need to be defined separately. An environment group must be defined to group those servers and identify them as one PeopleSoft instance. In the workflow, the environment group needs to be defined as the destination environment in the migration steps.

Define the Development, Test, and Production environments as shown in [Table 3-3](#).

*Table 3-3. Environment window required fields for Development, Test, and Production environments*

Field Name	Value
<b>Host tab, Server section</b>	
Enable Server	Enabled (select this checkbox).
Name	Name of server where the Development, Test, or Production instance files reside.
Type	<b>Windows Server</b> or <b>UNIX (Generic)</b> .
Username	Username.
Password	Password for Username.
NT Domain	Domain name if server type is <b>Windows Server</b> .
Base Path	Base path for development, test, or production files.
Connection Protocol	Connection protocol.
Transfer Protocol	Transfer protocol.
<b>Host tab, Client section</b>	
Enable Client	Disabled (deselect this checkbox).
<b>Host tab, Database section</b>	
Enable Database	Enabled (select this checkbox).
Type	<b>Oracle Server</b> .
Host Name	Name of the server where the instance files reside.
Connect String	Connect string for database.
Username	Logon name for database.

*Table 3-3. Environment window required fields for Development, Test, and Production environments [continued]*

Field Name	Value
Password	Password for user name.
Oracle SID	(Oracle database only) SID for the database.
DB Link	(Oracle database only) Link from the Mercury IT Governance Center instance to the PeopleSoft database
Port Number	Port number.
<b>Extension Data</b> tab, <b>PeopleSoft</b> subtab	
Enabled	<b>Yes.</b>
Set as default Extension	<b>Yes.</b>
PeopleSoft Database Type	<b>Oracle.</b>
Environment Type	<b>Development, Test, or Production.</b>
PeopleSoft Version	<b>8.1.x or 8.4.x.</b>
PeopleSoft Username	Userid for PeopleSoft operator. For more information, see the second note on page 45.
PeopleSoft Password	Password for PeopleSoft operator.
Bootstrap Username	Userid for system administrator. For more information, see the second note on page 45.
Bootstrap Password	Password for system administrator.
PS Home	PS_HOME for the instance.
Migration Path	Alternate to using the base path for Connect and Copy commands in cases where the path for the files is accessed through a mapped drive instead of through TELNET or SSH.

## Configuring Development/Test/Production Environment Groups

PeopleSoft implementations that have their file servers and batch servers on different machines use environment groups to identify them to a workflow as a PeopleSoft instance. In configuring an environment group:

- Use serial execution.
- The source environment can be either the file server or the batch server.
- To allow commands to be run against multiple servers in one execution, define the destination environment as the environment group.
- Use a custom field in the object type to allow the user to choose the source environment during creation of the package line.

Define the Development, Test, and Production environment groups as shown in [Table 3-4](#).

*Table 3-4. Environment Group window required fields for PeopleSoft Development, Test, and Production environment groups*

Field Name	Value
Environment Group Name	Name of the environment group
Execution Order	<b>Serial.</b>
Source Environment (No App Code)	PeopleSoft environment defining the file server.
<b>Environments tab</b>	
Associated Environments	PeopleSoft environments defining the batch and file server of the instance.

## Configuring PeopleSoft Users

The Extension allows an organization to include additional information specific to how a user or administrator accesses the PeopleSoft Application and Version Control servers.

*Figure 3-3* shows the User window, **Extension Data** tab, **PeopleSoft** subtab. *Table 3-5* contains field definitions.

*Figure 3-3.* User window, PeopleSoft subtab

*Table 3-5.* User window, PeopleSoft subtab field definitions

Field Name	Definition
Set as default Extension	Whether or not <b>PeopleSoft</b> is the default subtab to show for this window's <b>Extension Data</b> tab.
Enabled	Whether or not the fields on this tab are available to be configured.
PeopleSoft Operator ID	Operator ID for developer or administrator to interact with the PeopleSoft Application server.
Password	Password for PeopleSoft Operator ID.
Version Control Server Username	Version Control server username for developer or administrator to connect to the Version Control server.
Password	Password for Version Control server username.

To access the PeopleSoft subtab:

1. Open the Workbench.
2. In the shortcut bar, click **Sys Admin** and click the **Users** icon.
3. Open a new user or, if the user of interest already exists, open it. The User window opens.
4. Click the tabs as required to configure a user. For this Extension in particular, click the **Extension Data** tab and then, at the bottom of the window, click the **PeopleSoft** subtab.

### **For More Information**

For more information about defining Mercury IT Governance Center users and privileges, see the *System Administration Guide and Reference*.

## Mass Password Change Functionality

Mercury Change Management Extension for PeopleSoft Enterprise includes a script that enables the user to do a mass password change for fields in the Extension **Environment** tab. When the Extension is installed, the script `kPSFTEnvUpdatePassword.sh` is created.

You can run a script from the `ITG_Home/bin` directory to execute a mass password change for the PeopleSoft user password or the Bootstrap user password.

For the PS user password, run:

```
kPSFTEnvUpdatePassword.sh  
1 ITG_Username ITG_PW PeopleSoft_Username Old_PW New_PW
```

For the Bootstrap user password, run:

```
kPSFTEnvUpdatePassword.sh  
2 ITG_Username ITG_PW Bootstrap_Username Old_Bootstrap_PW New_Bootstrap_PW
```

The value of the first argument (**1** or **2**) determines whether the PeopleSoft user password or the Bootstrap password is changed.

For example, to change the password for the PeopleSoft user PSUSER1 from PS1 to PS2, run the script:

```
$ ../sh kPSFTEnvUpdatePassword.sh 1 admin admin PSUSER1 PS1 PS2
```



# Chapter 4 Extension Entities

---

## In This Chapter:

- *Object Types*
  - *Overview of Object Types*
  - *List of Object Types*
  - *PeopleSoft 8.1.x Patch, 8.4.x Patch, and 8.44 Patch Object Types*
  - *PeopleSoft COBOL Copybook Object Type*
  - *PeopleSoft COBOL DMS Object Type*
  - *PeopleSoft COBOL Program Object Type*
  - *PeopleSoft Data File Object Type*
  - *PeopleSoft Load DMS Object Type*
  - *PeopleSoft Project (ORACLE) Object Type*
  - *PeopleSoft SQC Object Type*
  - *PeopleSoft SQL Script Object Type*
  - *PeopleSoft SQR Object Type*
- *Request Types*
  - *Overview of Request Types*
  - *Reference Request Types*
  - *List of Request Types*
  - *PeopleSoft Bug Request Type*
  - *PeopleSoft Enhancement Request Type*
  - *PeopleSoft Patch Request Type*
- *Workflows*
  - *Overview of Workflows*
  - *Reference Workflows*
  - *List of Workflows*
  - *PeopleSoft Bug/Enhancement Workflow*

- *PeopleSoft Development Workflow*
  - *PeopleSoft Development Workflow w/ Project Rollback & Object Warning Workflow*
  - *PeopleSoft Patch Request Workflow*
  - *PeopleSoft Patch Applicator Workflows*
  - *Report Types*
    - *Overview of Report Types*
    - *Reference Report Types*
    - *List of Report Types*
    - *PeopleSoft Compare Custom Database Setup Report*
    - *PeopleSoft Comparison Report*
    - *PeopleSoft Customized Objects Report*
    - *PeopleSoft File Comparison Report*
    - *PeopleSoft Object Customization Detail Report*
    - *PeopleSoft Object Migrations Report*
    - *PeopleSoft Patches Applied to an Environment Report*
    - *PeopleSoft Project Migration Impact Analysis Report*
    - *PeopleSoft Request Detail Report*
    - *PeopleSoft Version Control - Baseline Report*
    - *PeopleSoft Version Control - Break Lock Report*
    - *PeopleSoft Version Control - Check In Report*
    - *PeopleSoft Version Control - Check Out Report*
  - *Special Commands*
  - *Tokens*
-

## Object Types

This section provides reference information about the PeopleSoft-specific object types provided in the Extension. These object types are listed and defined in [Table 4-1](#).

### Overview of Object Types

Migration and compilation of object types are driven by commands included within the object types. For more information about commands in the Mercury IT Governance Center environment, see the *Commands, Tokens, and Validations Guide and Reference*. For more information about using object types in packages, see the *Mercury Change Management User's Guide*.

You can view or modify an object type as follows:

1. Open the Workbench.
2. From the shortcut bar, click **Change Mgmt** and click the **Object Types** icon.
3. If you want, select a particular Extension from the Extension drop-down list.
4. In the Object Type Workbench window, click **List**.
5. Select the object type of interest and click **Open**.
6. Edit the object type. (On the **Fields** tab, the list of fields in the Prompts column is alphabetized.)

Subsequent figures in this section show screens you can use to revise object types in conjunction with adding package lines. You can access these screens as follows:

1. Open the Workbench.
2. From the shortcut bar, click **Change Mgmt** and click the **Packages** icon.
3. Add a new or open an existing package, as necessary.
4. Select a workflow.
5. Add a line.
6. Select the object type of interest.

## Reference Object Types

Reference (REFERENCE) object types cannot be edited, but you can copy and rename them and edit the copies to meet your needs. You can also use existing non-reference object types as is or configure them further to meet your needs.

## List of Object Types

*Table 4-1* lists and defines the object types included in the Extension. Each is described in subsequent sections.

*Table 4-1. Object types included in the Extension*

Object Type Name	Definition
PeopleSoft 8.1.x Patch Object Type	Applies PeopleSoft Version 8.1.x patches.
PeopleSoft 8.4.x Patch Object Type	Applies PeopleSoft Version 8.4.x patches.
PeopleSoft 8.44 Patch Object Type	Applies PeopleSoft Version 8.44 patches.
PeopleSoft COBOL Copybook	Copies a copybook from one environment to another, and then recompiles the entire <code>cblbin</code> directory.
PeopleSoft COBOL DMS	Copies a data mover script from one environment to another, and then runs the data mover tool, loading appropriate SQL statements into a table.
PeopleSoft COBOL Program	Copies a program from one environment to another and then compiles it in the new location.
PeopleSoft Data File	Data file that accompanies a data mover script.
PeopleSoft Load DMS	Loads online object or functional data into a database instance.
PeopleSoft Project (ORACLE)	Copies a project and associated online objects from one environment to another.
PeopleSoft SQC	Copies the SQC from one environment to another.
PeopleSoft SQL Script	Runs SQL script generated by project rebuild.
PeopleSoft SQR	Copies the SQR from one environment to another.

## **PeopleSoft 8.1.x Patch, 8.4.x Patch, and 8.44 Patch Object Types**

The Extension includes the following object types that apply PeopleSoft patches:

- PeopleSoft 8.1.x Patch Object Type
- PeopleSoft 8.4.x Patch Object Type
- PeopleSoft 8.44 Patch Object Type

For information about these object types, see [Chapter 5, \*Patch Applicator\*, on page 133](#).

## PeopleSoft COBOL Copybook Object Type

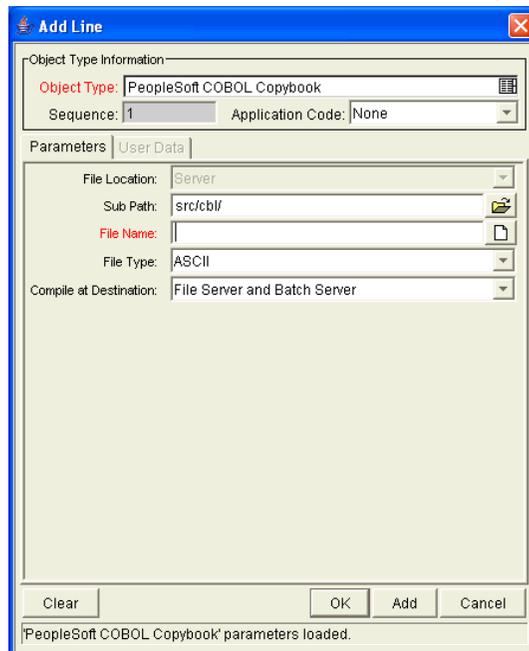
The PeopleSoft COBOL Copybook object type copies COBOL Copybook programs from one instance to another, recompiles the programs at the destination specified in the fields, and moves the executables generated during the compile to the `cblbin` directory.

In UNIX batch server environments, be sure the environment variables are set when users log on to the system. If you are using only one logon name for all instances, add a line in the commands to set the environment variables before running any compilations. For example:

```
. DEST_ENV.DB_CONNECT_STRING
cd DEST_ENV.SERVER_BASE_PATH/install
PS_HOME="DEST_ENV.AC.PS_HOME"
./pscbl.mak
```

For this example, set the path for `DEST_ENV.DB_CONNECT_STRING` in the **Host** tab of the Environment window.

*Figure 4-1* shows sample data for the PeopleSoft COBOL Copybook object type when adding a package line. *Table 4-2* contains field definitions for the object type.



The screenshot shows a dialog box titled "Add Line" with a blue header and a close button. The main area is divided into two sections: "Object Type Information" and "Parameters".

- Object Type Information:**
  - Object Type: PeopleSoft COBOL Copybook
  - Sequence: 1
  - Application Code: None
- Parameters:**
  - File Location: Server
  - Sub Path: src/cbl/
  - File Name: (empty)
  - File Type: ASCII
  - Compile at Destination: File Server and Batch Server

At the bottom of the dialog are buttons for "Clear", "OK", "Add", and "Cancel". A status bar at the very bottom of the window reads "PeopleSoft COBOL Copybook' parameters loaded."

*Figure 4-1. PeopleSoft COBOL Copybook object type sample data*

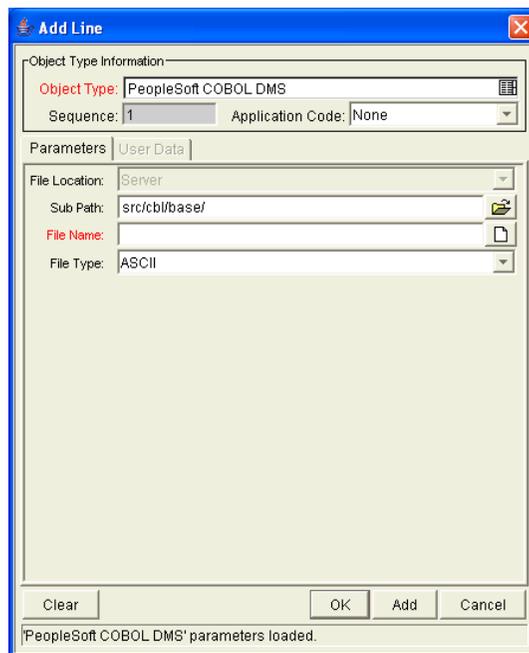
Table 4-2. PeopleSoft COBOL Copybook object type field definitions

Field Name	Definition
File Location	File location: <b>Client</b> or <b>Server</b> .
Sub Path	Sub path (relative to base path) to the COBOL source files.
File Name	Name of the file to be migrated.
File Type	File type: <b>ASCII</b> or <b>Binary</b> .
Compile at Destination	<b>File Server and Batch Server, File Server, Batch Server, or None.</b>

## PeopleSoft COBOL DMS Object Type

The PeopleSoft COBOL DMS object type copies PeopleSoft Data Mover script files from one instance to another and executes them by invoking PeopleSoft Data Mover. It logs onto the Data Mover using the specified operator ID and password (PeopleSoft or Bootstrap) of the destination instance, runs the script, and generates a log file containing the results of the execution.

*Figure 4-2* shows sample data for the PeopleSoft COBOL DMS object type when adding a package line. *Table 4-3* contains field definitions for the object type.



*Figure 4-2. PeopleSoft COBOL DMS object type sample data*

*Table 4-3. PeopleSoft COBOL DMS object type field definitions*

Field Name	Definition
File Location	File location: <b>Client</b> or <b>Server</b> .
Sub Path	Sub path (relative to base path) to the DMS files.
File Name	COBOL DMS file to be migrated.
File Type	File type: <b>ASCII</b> or <b>Binary</b> .

## PeopleSoft COBOL Program Object Type

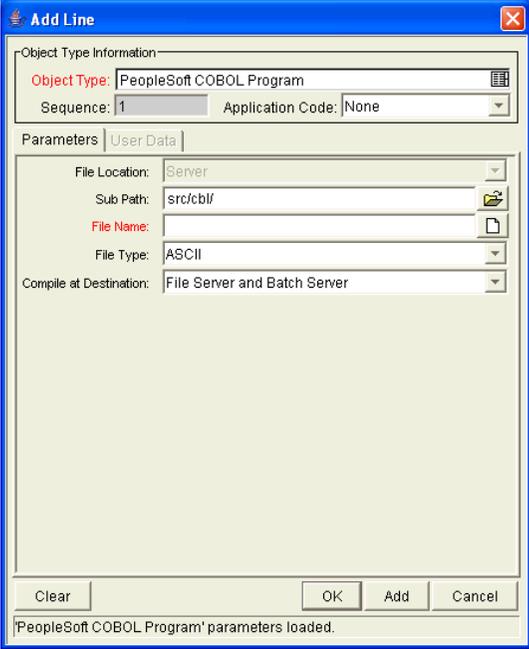
The PeopleSoft COBOL Program object type copies COBOL programs from one instance to another, compiles them at the destination specified in the fields, and moves all executables generated during the compile to the `cblbin` directory.

In UNIX batch server environments, be sure the environment variables are set when users log on to the system. If you are using only one logon name for all instances, add a line in the commands to set the environment variables before running any compilations. For example:

```
. DEST_ENV.DB_CONNECT_STRING
cd DEST_ENV.SERVER_BASE_PATH/install
PS_HOME="DEST_ENV.AC.PS_HOME"
./pscbl.mak
```

For this example, set the path for `DEST_ENV.DB_CONNECT_STRING` in the **Host** tab of the Environment window.

[Figure 4-3](#) shows sample data for the PeopleSoft COBOL Program object type when adding a package line. [Table 4-4](#) contains field definitions for the object type.



The screenshot shows a dialog box titled "Add Line" with a close button in the top right corner. The dialog is divided into two main sections: "Object Type Information" and "Parameters".

**Object Type Information:**

- Object Type:** PeopleSoft COBOL Program
- Sequence:** 1
- Application Code:** None

**Parameters:** (User Data)

- File Location:** Server
- Sub Path:** src/cbl/
- File Name:** (empty field)
- File Type:** ASCII
- Compile at Destination:** File Server and Batch Server

At the bottom of the dialog, there are four buttons: "Clear", "OK", "Add", and "Cancel". Below the buttons, a status bar reads: "PeopleSoft COBOL Program' parameters loaded."

*Figure 4-3. PeopleSoft COBOL Program object type sample data*

*Table 4-4. PeopleSoft COBOL Program object type field definitions*

Field Name	Definition
File Location	File location: <b>Client</b> or <b>Server</b> .
Sub Path	Sub path (relative to base path) to the COBOL source files.
File Name	COBOL program file to be migrated.
File Type	File type: <b>ASCII</b> or <b>Binary</b> .
Compile at Destination	Destination compilation location: <b>File Server and Batch Server, File Server, Batch Server, or None</b> .

## PeopleSoft Data File Object Type

The PeopleSoft Data File object type copies Data Mover Script (DMS) data files from one instance to another.

*Figure 4-4* shows sample data for the PeopleSoft Data File object type when adding a package line. *Table 4-5* contains field definitions for the object type.

*Figure 4-4. PeopleSoft Data File object type sample data*

*Table 4-5. PeopleSoft Data File object type field definitions*

Field Name	Definition
File Location	File location: <b>Client</b> or <b>Server</b> .
Sub Path	Sub path (relative to base path) to the data files.
File Name	Name of the data file to be migrated.
File Type	File type: <b>ASCII</b> or <b>Binary</b> .

## PeopleSoft Load DMS Object Type

The PeopleSoft Load DMS object type copies Load DMS files from one instance to another and executes them by invoking PeopleSoft Data Mover. It logs onto the Data Mover using the specified operator id and password (PeopleSoft or Bootstrap) of the destination instance, runs the script, and generates a log file containing the results of the execution. If a data file is associated with the DMS file, the data file needs to be moved to the destination environment first, using the PeopleSoft Data File object type.

*Figure 4-5* shows sample data for PeopleSoft Load DMS object type when adding a package line. *Table 4-6* contains field definitions for the object type.

The screenshot shows a dialog box titled "Add Line" with a close button in the top right corner. The dialog is divided into two main sections: "Object Type Information" and "Parameters".

**Object Type Information:**

- Object Type: PeopleSoft Load DMS
- Sequence: 1
- Application Code: None

**Parameters:** (User Data)

- File Location: Server
- Sub Path: scripts/
- File Name: (empty)
- File Type: ASCII
- Execution Type: NORMAL
- Sub Path for Data File: data/

At the bottom of the dialog, there are four buttons: "Clear", "OK", "Add", and "Cancel". Below the buttons, a status bar reads: "PeopleSoft Load DMS' parameters loaded."

*Figure 4-5. PeopleSoft Load DMS object type sample data*

Table 4-6. PeopleSoft Load DMS object type field definitions

Field Name	Definition
File Location	File location: <b>Client</b> or <b>Server</b> .
Sub Path	Sub path (relative to base path) to the Load DMS file.
File Name	Load DMS file to be migrated.
File Type	File type: <b>ASCII</b> or <b>Binary</b> .
Execution Type	Execution type: <b>NORMAL</b> or <b>BOOTSTRAP</b> .
Sub Path for Data File	Sub path where corresponding data files for Load DMS are located.

## PeopleSoft Project (ORACLE) Object Type

The PeopleSoft Project (ORACLE) object type invokes Pstools to copy a project and all its entities from one instance to another. It logs onto Pstools using the PeopleSoft operator id and password of the source instance, copies the project to the destination, and generates a log file containing the results of the copy.

To build the project in the destination, set the Build Project field to **Yes**. The settings used to create the build script(s) are read from the Windows registry. At least one project build must have been created on that computer using values in the registry collected by the Application Designer. To execute the build script(s), set the Execute Build Script field to **Yes**.

The PeopleSoft Project (ORACLE) object type can also be used after copying a project from a source to a destination environment to roll back the project to its previous state. In this case, the project being copied must exist in the destination database. If the project copied is a new project, then for rollback purposes it must be manually deleted from the destination environment. You should also create a backup directory in the `scripts` directory of the destination instance where the backup copy of the project to be modified will be created.

The PeopleSoft Development Workflow w/ Project Rollback & Object Warning workflow can be used with the PeopleSoft Project (ORACLE) object type in both rollback and locking scenarios. For more information, see [PeopleSoft Development Workflow w/ Project Rollback & Object Warning Workflow on page 89](#).

[Figure 4-6](#) shows sample data for the PeopleSoft Project (ORACLE) object type when adding a package line. [Table 4-7](#) contains field definitions for the object type.

The screenshot shows the 'Add Line' dialog box for the 'PeopleSoft Project (ORACLE)' object type. The 'Object Type Information' section shows 'Object Type: PeopleSoft Project (ORACLE)', 'Sequence: 1', and 'Application Code: None'. The 'Parameters' section is active, showing 'User Data' with the following fields: 'Project Backup Directory' (scripts/backup/), 'Project Name' (empty), 'Build Project?' (radio buttons for Yes and No, with No selected), 'Execute Build Script?' (radio buttons for Yes and No, with No selected), 'Whenever sqlerror' (dropdown menu showing 'exit failure'), 'Spool Filename' (spool\_[PKG.ID]\_[PKGL.SEQ].lst), and 'Check Objects Being Migrated?' (radio buttons for Yes and No, with No selected). At the bottom, there are 'Clear', 'OK', 'Add', and 'Cancel' buttons. A status bar at the bottom indicates 'PeopleSoft Project (ORACLE) parameters loaded.'

Figure 4-6. PeopleSoft Project (ORACLE) object type sample data

Table 4-7. PeopleSoft Project (ORACLE) object type field definitions

Field Name	Definition
File Location (Hidden by default*)	File location: <b>Client</b> or <b>Server</b> .
Project Backup Directory	Directory that will hold the project backup file (that is, an export of the project).
Project Name	Name of the project be copied.
Build Project	Whether or not to build the project in the destination database.
Psbuilt.sql filename with path (Hidden by default*)	Full path and filename of Psbuilt.sql for single build scripts.
Psbuilt.log filename with path (Hidden by default*)	Full path and filename of Psbuilt.log. The value of this field is set internally.
Execute Build Script	Whether or not to execute the build script(s).
Whenever sqlerror	Action to take when an SQL error is encountered during the build process: <b>exit failure</b> or <b>continue</b> .
Spool Filename	Spool file to capture the log from executing the build script(s).

Table 4-7. PeopleSoft Project (ORACLE) object type field definitions [continued]

Field Name	Definition
New Project (Hidden by default*)	Whether the project is new. The value of this field is set internally.
Check Objects Being Migrated	Whether or not to check for other packages that are also migrating the objects contained in the specified project.
Objects Exist (Hidden by default*)	Whether or not package lines were found containing any of the entities in the project being migrated by the current package line. If so, a warning message is presented.
Multiple Build Script (Hidden by default* and new in release 6.0.0)	Whether or not to create multiple build scripts. If set to <b>No</b> , creates a single build script.
Pstable.sql filename with path (Hidden by default* and new in release 6.0.0)	Full path and filename of the Project Build create table SQL script.
Psindex.sql filename with path (Hidden by default* and new in release 6.0.0)	Full path and filename of the Project Build create index SQL script.
Pstrigger.sql filename with path (Hidden by default* and new in release 6.0.0)	Full path and filename of the Project Build create trigger SQL script.
Psview.sql filename with path (Hidden by default* and new in release 6.0.0)	Full path and filename of the Project Build create view SQL script.
Psalter.sql filename with path (Hidden by default* and new in release 6.0.0)	Full path and filename of the Project Build create alter table SQL script.
Temp Project Exist (Hidden by default* and new in release 6.0.0)	Whether or not a temp project was created for backing up any existing object definitions found in the destination. This field is provided for status only and cannot be changed.

\* Hidden fields perform their functions in the background when this object type is used.

## PeopleSoft SQC Object Type

The PeopleSoft SQC object type copies SQC program files from one instance to another.

*Figure 4-7* shows sample data for the PeopleSoft SQC object type when adding a package line. *Table 4-8* contains field definitions for the object type.

*Figure 4-7. PeopleSoft SQC object type sample data*

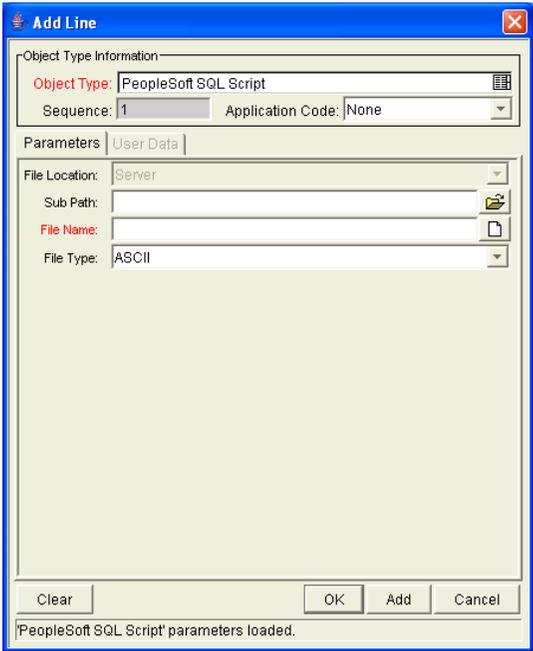
*Table 4-8. PeopleSoft SQC object type field definitions*

Field Name	Definition
File Location	File location: <b>Client</b> or <b>Server</b> .
Sub Path	Sub path (relative to base path) to the SQC files.
File Name	SQC file to be migrated.
File Type	File type: <b>ASCII</b> or <b>Binary</b> .

## PeopleSoft SQL Script Object Type

The PeopleSoft SQL Script object type executes the SQL script in the destination instance. The SQL script file must exist in the destination directory prior to execution.

*Figure 4-8* shows sample data for the PeopleSoft SQL Script object type when adding a package line. *Table 4-9* contains field definitions for the object type.



*Figure 4-8. PeopleSoft SQL Script object type sample data*

*Table 4-9. PeopleSoft SQL Script object type field definitions*

Field Name	Definition
File Location	File location: <b>Client</b> or <b>Server</b> .
Sub Path	Sub path (relative to base path) to the SQC files.
File Name	SQC file to be migrated.
File Type	File type: <b>ASCII</b> or <b>Binary</b> .

Table 4-9. PeopleSoft SQL Script object type field definitions [continued]

Field Name	Definition
Spool Filename (Hidden by default*)	Name of the spool file. The default name is <code>spool_package_package_line.lst</code> . Specify your own spool file name.
Whenever sqlerror (Hidden by default*)	Action to take when an SQL error is encountered during the build process: <b>exit failure</b> or <b>continue</b> .

\* Hidden fields perform their functions in the background when this object type is used.

## PeopleSoft SQR Object Type

The PeopleSoft SQR object type copies SQR program files from one instance to another and executes the SQR file at the destination for Oracle instances on Windows.

Figure 4-9 shows sample data for the PeopleSoft SQR object type when adding a package line. Table 4-10 contains field definitions for the object type.

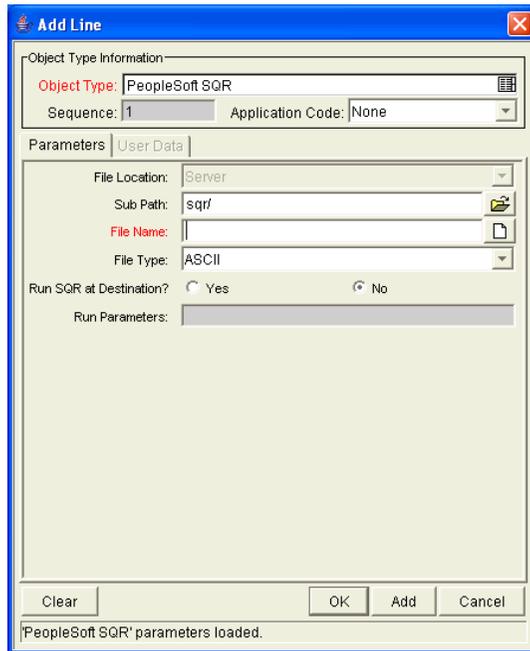


Figure 4-9. PeopleSoft SQR object type sample data

Table 4-10. PeopleSoft SQR object type field definitions

Field Name	Definition
File Location	File location: <b>Client</b> or <b>Server</b> .
Sub Path	Sub path (relative to base path) to the SQR files.
File Name	SQR file to be migrated.
File Type	File type: <b>ASCII</b> or <b>Binary</b> .
Run SQR at Destination	Whether or not to execute the SQR in the destination environment.
Run Parameters	Parameters to be passed to execute the SQR. Parameters passed usually include the <code>-ZIF sqr.ini</code> file and <code>-i instance sqr</code> directory flags. For example: <code>-ZIFc:/winnt40/sqr.ini -i e:/psft8/sqr/</code>

## Request Types

This section provides reference information about the PeopleSoft-specific request types provided in the Extension. These request types are listed and defined in [Table 4-11](#).

### Overview of Request Types

Migration and compilation of request types can be driven by commands included within the request types. For more information about commands in the Mercury IT Governance Center environment, see the *Commands, Tokens, and Validations Guide and Reference*. For more information about using request types in packages, see the *Mercury Change Management User's Guide*.

Subsequent figures of request type sample data show screens you can use to create requests from existing request types. You can access this screen as follows:

1. Make sure that the desired request type is enabled. In the Workbench shortcut bar, click **Demand Mgmt** and click the **Request Types** icon. If the desired request type needs to be enabled, open and enable it.
2. In the standard interface, select **Create > Request**. The Create New Request page appears.
3. Select the Request Type of interest from the drop-down list and click **Create**.

### Reference Request Types

Reference (REFERENCE) request types cannot be edited, but you can copy and rename them and edit the copies to meet your needs. You can also use existing non-reference request types as is or configure them further to meet your needs.

## List of Request Types

*Table 4-11* lists and defines the request types included in the Extension. Each is described in subsequent sections.

*Table 4-11. Request types included in the Extension*

Request Type Name	Definition
PeopleSoft Bug	Logs and tracks requests for bug fixes.
PeopleSoft Enhancement	Logs and tracks requests for enhancements.
PeopleSoft Patch Request Type	Logs and tracks requests for patches that must be applied to the PeopleSoft instance.

## PeopleSoft Bug Request Type

The PeopleSoft Bug request type logs and tracks requests for bug fixes. Once the appropriate resolution has been identified, a Mercury Change Management package can be generated and the PeopleSoft Patch Applicator, Customization Applicator, or individual object migration process can be used to apply the required changes.

You should use the PeopleSoft Bug/Enhancement workflow (see [PeopleSoft Bug/Enhancement Workflow on page 85](#)) as the default workflow for this request type.

You can modify the Product Category and Product Line fields in this request type to include a list of the PeopleSoft applications and modules you are using. To do this, generate a new validation with a drop-down list that includes the application and module names. Reference the validation in the appropriate request type fields.

[Figure 4-10](#) shows the screen for creating a PeopleSoft Bug request based on the provided request type. [Table 4-12](#) contains field definitions for the request type.

Figure 4-10. PeopleSoft Bug request creation

Table 4-12. PeopleSoft Bug request type field definitions

Field Name	Definition
<b>Summary</b> tab (not expanded in <a href="#">Figure 4-10</a> )	
Workflow	PeopleSoft Bug/Enhancement Workflow
<b>Bug</b> tab	
PeopleSoft Product	PeopleSoft product impacted by the bug.
Product Category	Product category impacted by the bug.
Product Line	Product line impacted by the bug.
Product Release	Product release impacted by the bug.
Problem	Problem that the product has encountered.
Error Log	Web address pointing to the error log.
Reproducible	Whether or not the error is reproducible.
Steps to Replicate	Steps needed to replicate the error.
Impact	Impact: <b>Severe, Warning, Minor, or Cosmetic.</b>
Difficulty	Difficulty to fix: <b>Easy, Medium, or Hard.</b>
Estimated Time to Complete	Estimated time needed to fix the bug.
Resolution	Resolution: <b>Fixed, Not Fixed, Not Reproducible, Duplicate, or Process Fix.</b>
Resolution Summary	Description of what was done to fix the bug.

## PeopleSoft Enhancement Request Type

The PeopleSoft Enhancement request type logs and tracks requests for enhancements related to PeopleSoft instances. Once the appropriate customization has been approved and generated, it can be tested and migrated to the proper instance using PeopleSoft object migrators.

You should use the PeopleSoft Bug/Enhancement workflow (see [PeopleSoft Bug/Enhancement Workflow on page 85](#)) as the default workflow for this request type.

You can modify the Product Category and Product Line fields in this request type to include a list of the PeopleSoft applications and modules you are using. To do this, generate a new validation with a drop-down list that includes the application and module names. Reference the validation in the appropriate request type fields.

[Figure 4-11](#) shows the screen for creating a PeopleSoft Enhancement request based on the provided request type. [Table 4-13](#) contains field definitions for the request type.

Create New PeopleSoft Enhancement

Expand All Collapse All Submit Cancel

Header

Summary

Details

Enhancement

\*PeopleSoft Product  Impact

Product Category  Difficulty

Product Line  Estimated Time to Complete:

Product Release  Enhancement Type New Customization

\*Justification

Resolution

Resolution Summary

Report Name(s)

Program Name(s)

PeopleSoft Project Name

Notes

References

Submit Cancel

Figure 4-11. PeopleSoft Enhancement request creation

Table 4-13. PeopleSoft Enhancement request type field definitions

Field Name	Definition
<b>Summary</b> tab (not expanded in <a href="#">Figure 4-11</a> )	
Workflow	PeopleSoft Bug/Enhancement Workflow
<b>Enhancement</b> tab	
PeopleSoft Product	PeopleSoft product impacted by the enhancement.
Product Category	Product category impacted by the enhancement.
Product Line	Product line impacted by the enhancement.
Product Release	Product release impacted by the enhancement.
Justification	Justification for the enhancement.
Impact	Impact: <b>Severe, Warning, Minor, or Cosmetic.</b>
Difficulty	Difficulty to fix: <b>Easy, Medium, or Hard.</b>
Estimated Time to Complete	Estimated time needed to fix the bug.
Enhancement Type	Enhancement type: <b>New Customization, Basic Customization, or Additional Customization.</b>
Resolution	Resolution: <b>Fixed, Not Fixed, Not Reproducible, Duplicate, or Process Fix.</b>
Resolution Summary	Description of the changes made for the enhancement.
Report Name(s)	Reports generated or modified.
Program Name(s)	Programs generated or modified.
PeopleSoft Project Name	Project containing database objects that were generated or modified.

## PeopleSoft Patch Request Type

The PeopleSoft Patch request type logs and tracks requests for patches that need to be applied to a PeopleSoft instance. Once functional leads acknowledge the need for the patch, a Mercury Change Management package can be generated and the PeopleSoft Patch Applicator can be used to apply the required changes.

Use the PeopleSoft Patch Request workflow as the default workflow for this request type.

*Figure 4-12* shows the screen for creating a PeopleSoft Patch Request based on the provided request type. *Table 4-14* contains field definitions for the request type.

Create New PeopleSoft Patch Request Type

Expand All Collapse All Submit Cancel

Header

Summary

Details

Patch/Upgrade Details

Hotfix:

Upgrade Patch Description:

Special Note:

Type:

Report ID:

Date Entered:

Impacted Product Line:

Impacted Product:

Impacted Release:

Impacted Platforms:

Issue:  [View URL](#)

Notes

References

Submit Cancel

Figure 4-12. PeopleSoft Patch Request creation

Table 4-14. PeopleSoft Patch Request request type field definitions

Field Name	Definition
<b>Summary</b> tab (not expanded in <a href="#">Figure 4-12</a> )	
Workflow	PeopleSoft Patch Request Workflow
<b>Patch/Upgrade Details</b> tab	
Notify	Security group to be notified.
Upgrade/Patch Description	Description of the patch.
Special Note	Special note to be added to the patch.
Type	Patch type.
Report ID	Report ID that identifies the patch.
Date Entered	Date the request was entered.
Impacted Product Line	Product line impacted by the patch.
Impacted Product	Product release impacted by the bug.
Impacted Release	PeopleSoft release impacted by the patch.
Impacted Platforms	Platforms impacted by the patch.
Issue	Web address pointing to the error log.

## Workflows

This section provides reference information about the PeopleSoft-specific workflows provided in the Extension. These workflows are listed and defined in [Table 4-15](#).

### Overview of Workflows

Some workflows are associated with object types and Mercury Change Management, and include package execution. Other workflows are associated with request types and Mercury Demand Management, and describe decision-making processes. For more information about Mercury Change Management and Mercury Demand Management, see the *Mercury Change Management: Configuring a Deployment System* document and the *Mercury Demand Management: Configuring a Request Resolution System* document.

Migration and compilation of entities using workflows are driven by commands included within the entities or workflow steps. For more information about commands in the Mercury IT Governance Center environment, see the *Commands, Tokens, and Validations Guide and Reference*.

### Reference Workflows

Reference (REFERENCE) workflows cannot be edited, but you can copy and rename them and edit the copies to meet your needs. You can also use existing non-reference workflows as is or configure them further to meet your needs.

## List of Workflows

*Table 4-15* lists and defines the workflows included in the Extension. Each is described in subsequent sections.

*Table 4-15. Workflows included in the Extension*

Workflow Name	Product Scope	Definition
PeopleSoft Bug/Enhancement Workflow	Mercury Demand Management	Logs and tracks requests for bug fixes and software enhancements.
PeopleSoft Development Workflow	Mercury Change Management	Develops and tests enhancements to be made to PeopleSoft applications.
PeopleSoft Development Workflow w/ Project Rollback & Object Warning	Mercury Change Management	Processes the development and testing of enhancements to PeopleSoft applications.
PeopleSoft Patch Request Workflow	Mercury Demand Management	Used by the PeopleSoft Patch Request Type to make sure that the required patches are downloaded and packages are created from them.
PeopleSoft 8.1.x Patch Applicator	Mercury Change Management	Deploys and tracks patches for critical fixes to PeopleSoft Version 8.1.x applications.
PeopleSoft 8.4.x Patch Applicator	Mercury Change Management	Deploys and tracks patches for critical fixes to PeopleSoft Version 8.4.x applications.
PeopleSoft 8.44 Patch Applicator	Mercury Change Management	Deploys and tracks patches for critical fixes to PeopleSoft Version 8.44 applications.

## PeopleSoft Bug/Enhancement Workflow

The PeopleSoft Bug/Enhancement workflow is used in conjunction with the PeopleSoft Bug request type and the PeopleSoft Enhancement request type to make sure that requests related to PeopleSoft applications are resolved in a timely manner.

Figure 4-13 shows the workflow diagram. Steps in the workflow are described in the procedure that follows.

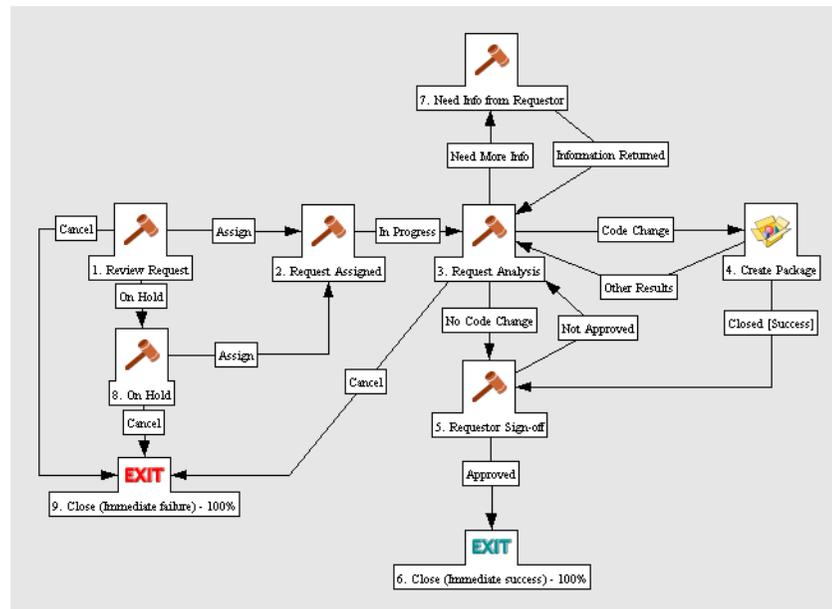


Figure 4-13. PeopleSoft Bug/Enhancement Workflow

The workflow steps are described as follows:

- 1. Review Request**—Requests are generated using either Mercury IT Governance Workbench or Mercury Demand Management. Once submitted, the request becomes eligible for review. From here, the request may be assigned, placed on hold, or cancelled.
- 2. Request Assigned**—Once the request is assigned, it can be set in progress.
- 3. Request Analysis**—The request then undergoes request analysis, which can result in code change, no code change, a need for more information, or cancellation of the request.
- 4. Create Package**—If there is a need for code change, then a package is generated. The object types and workflow needed to apply the changes to the

application instances depend on the changes made. For example, a package to apply a patch is generated if the request can be resolved by applying a patch.

**5. Requestor Sign-off**—Once the package has been successfully deployed or if there was no code change required, the request waits for the requestor to sign off on the request before it is closed. If the requestor does not approve the request, then the request returns to the Request Analysis step.

**6. Close (Immediate success)**—After sign-off, the request is closed with a status of success.

**7. Need Info from Requestor**—If a need for more information was identified in the Request Analysis step, then the request status changes to pending requestor information until the requestor adds more information and sets the status back to in progress.

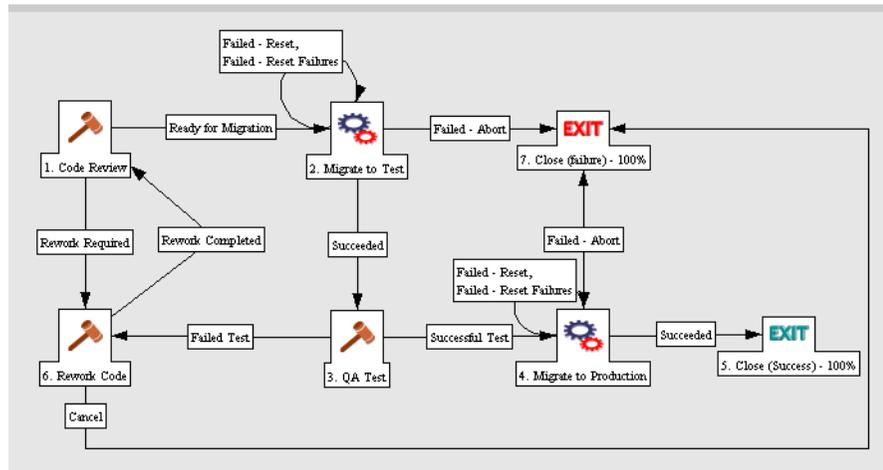
**8. On Hold**—If the request was placed on hold when it was first reviewed, it stays there until the request status is changed to assigned or cancelled.

**9. Closed (Immediate failure)**—If a request is cancelled at any point, it is closed with a status of failure.

## PeopleSoft Development Workflow

The PeopleSoft Development workflow develops and tests enhancements to be made to a PeopleSoft application. This can include changes made to programs, reports, and online objects.

*Figure 4-14* shows the workflow diagram. Steps in the workflow are described in the procedure that follows.



*Figure 4-14. PeopleSoft Development workflow*

The workflow steps are described as follows:

- 1. Code Review**—Packages are generated using the Mercury IT Governance Workbench. The package lines identify the entities to be migrated. Once submitted, the package becomes eligible for code review. From there, it can be identified as ready for migration or as requiring rework.
- 2. Migrate to Test**—Once the code review is approved and the changes are ready for migration, the entities are migrated to the test instance.
- 3. QA Test**—If the migration succeeds, it proceeds to this step.
- 4. Migrate to Production**—If the QA test passes, the changes are migrated to the production instance.
- 5. Close (Success)**—If migration to production is successful, the package line is closed with a status of success.

**6. Rework Code**—If the Code Review step or the QA Test step identified that code rework is required, the rework is conducted in this step. Once this step is approved, the package returns to the Code Review step.

**7. Close (failure)**—If code rework or migration to test or production is cancelled, the package line is closed with a status of failure.

## PeopleSoft Development Workflow w/ Project Rollback & Object Warning Workflow

The PeopleSoft Development Workflow w/ Project Rollback & Object Warning workflow processes the development and testing of enhancements to the PeopleSoft application. This can include changes made to programs, reports, and online objects. In conjunction with the PeopleSoft Project Object Warning SubWorkflow, this workflow is used to prevent conflicts in migrations of PeopleSoft objects.

Object locking, project rollback, and object warning functionality can be utilized when the workflow is used in conjunction with the PeopleSoft Project (ORACLE) object type.

*Figure 4-15*, *Figure 4-16*, and *Figure 4-17* show the workflow diagram for the PeopleSoft Development Workflow w/ Project Rollback & Object Warning workflow and two subworkflows. Steps in the workflows and subworkflows are described in the procedure that follows.



Note

A script must be run to use the object warning functionality. See [System Requirements](#) on page 26.

To utilize the object warning functionality of the PeopleSoft Project Object Warning Subworkflow, create a view in the Mercury IT Governance schema for all PeopleSoft instances:

```
CREATE OR REPLACE VIEW kacc_psft_psprojectitems_v
(environment_name
, projectname
, objecttype
, objectid1
, objectvalue1
, objectid2
, objectvalue2
, objectid3
, objectvalue3
, objectid4
, objectvalue4
, nodetype
)
AS
(select
'PeopleSoft environment1', projectname,
objecttype, objectid1, objectvalue1, objectid2, objectvalue2,
objectid3, objectvalue3, objectid4, objectvalue4, nodetype
from psprojectitem@dblinktoenvironment1)
union
```

```
(select 'PeopleSoft environment2', projectname,
objecttype,objectid1,objectvalue1,objectid2,objectvalue2,
objectid3,objectvalue3,objectid4,objectvalue4,nodetype
from psprojectitem@dblinktoenvironment2
```

For every PeopleSoft environment to be used for the object warning functionality, a union must be added to the view. This is the same view utilized by the PeopleSoft Project Migration Impact Analysis Report.

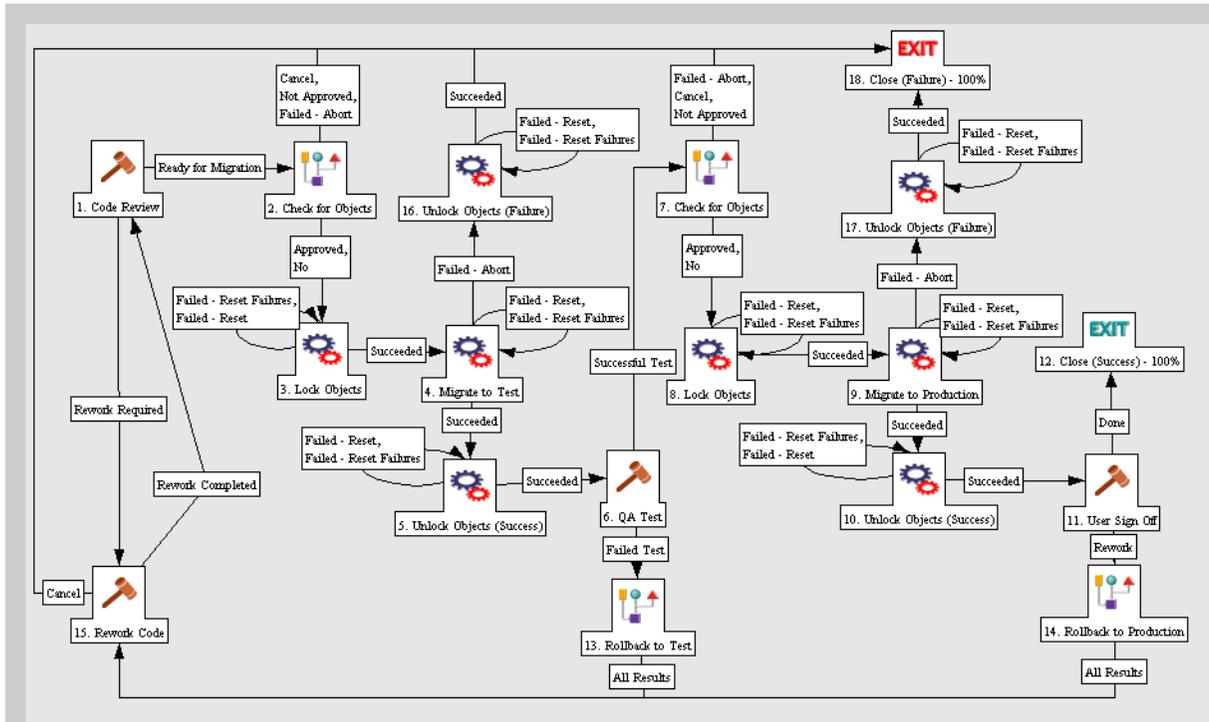


Figure 4-15. PeopleSoft Development Workflow w/ Project Rollback & Object Warning workflow

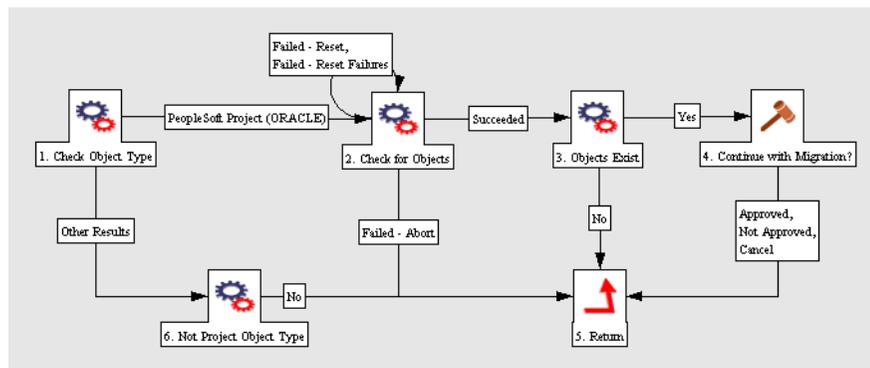


Figure 4-16. PeopleSoft Project Object Warning SubWorkflow

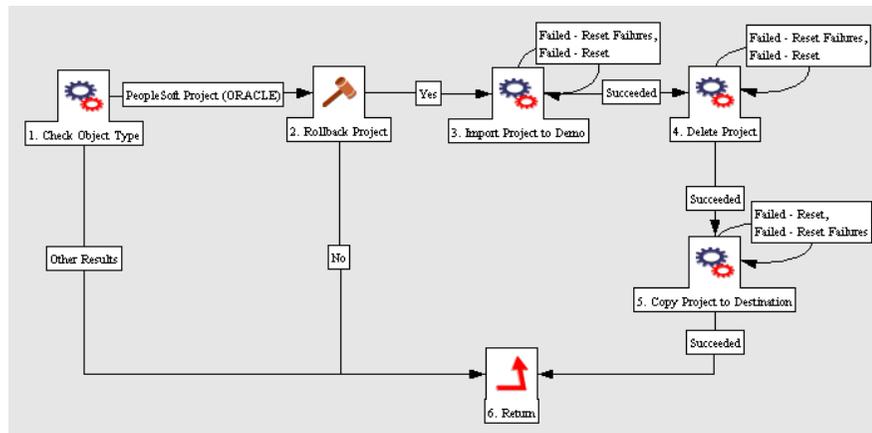


Figure 4-17. PeopleSoft Project Rollback SubWorkflow

The workflow steps are described as follows:

**1. Code Review**—Packages are generated using the Mercury IT Governance Workbench. The package lines identify the entities to be migrated. Once submitted, the package becomes eligible for code review. From there, it can be identified as ready for migration or as requiring rework.

**2. Check for Objects**—In this step, changes are identified as ready for migration. This initiates the PeopleSoft Project Object Warning SubWorkflow.

An object warning is provided if the status of any package line for the Migrate to Test step or Migrate to Production step is eligible, pending, or in progress.

PeopleSoft Project Object Warning SubWorkflow steps:

**2.1. Check Object Type**—This step determines whether or not the object type used is PeopleSoft Project (ORACLE).

**2.2. Check for Objects**—If the object type is identified as the PeopleSoft Project (ORACLE) object type, then all other package lines that have status of eligible or in progress and that use this object type are checked to determine whether the entities in the projects they are migrating exist in the project that is being migrated by the current package line.

**2.3. Objects Exist**—This step checks the outcome of the Checks for Objects step.

If no package lines that meet the described criteria are found, the step returns to the parent workflow.

**2.4. Continue with Migration?**—If package lines are found that meet the described criteria, the subworkflow prompts for approval of the migration in this step.

**2.5. Return**—If the migration is approved, this step returns to the parent workflow.

If the migration is cancelled or not approved, the step returns to the parent workflow, where the package line will be cancelled.

**2.6. Not Project Object Type**—If the Check Object Type step determines that the object type is not PeopleSoft Project (ORACLE), the subworkflow moves to this step and then returns to the parent workflow.

**3. Lock Objects**—If the lock for the project is approved, all the entities in the project are locked.

**4. Migrate to Test**—A backup of the existing project is taken from the destination and the project is migrated to the test instance.

If the migration succeeds, it goes to the Unlock Objects (Success) step, which unlocks all the entities (both source and destination) in the project.

If the migration fails, it can be restarted or cancelled. When the request is cancelled, it goes to the Unlock Objects (Failure) step.

**5. Unlock Objects (Success)**—This step unlocks all the entities (both source and destination) in the project, and then goes to the QA Test step.

**6. QA Test**—If the QA Test succeeds, the workflow proceeds to its Check for Object, Lock Objects, and Migrate to Production steps.

If the QA Test fails, the workflow goes to the Rollback to Test step, where it calls the PeopleSoft Project Rollback SubWorkflow.

**7. Check for Objects**—In this step, changes are identified as ready for migration. This initiates the PeopleSoft Project Object Warning SubWorkflow.

The subworkflow steps operate as described in step **2. Check for Objects**.

**8. Lock Objects**—If the lock for the project is approved, all the entities in the project are locked.

**9. Migrate to Production**—A backup of the existing project is taken from the destination and the project is migrated to the production instance.

If the migration succeeds, it goes to the Unlock Objects (Success) step, which unlocks all the entities (both source and destination) in the project.

If the migration fails, it can be restarted or cancelled. When the request is cancelled, it goes to the Unlock Objects (Failure) step.

**10. Unlock Objects (Success)**—This step unlocks all the entities (both source and destination) in the project, and then goes to the User Sign Off step.

**11. User Sign Off**—The user specifies whether the migration to production was successful.

**12. Close (Success)**—The package line is closed with successful completion.

**13. Rollback to Test**—This step calls the PeopleSoft Project Rollback SubWorkflow.

PeopleSoft Project Rollback SubWorkflow steps:

**13.1. Check Object Type**—If the object type used by the package line is PeopleSoft Project (ORACLE), the Rollback Project step becomes eligible. Otherwise, the subworkflow returns to the workflow's Rework Code step.

**13.2. Rollback Project**—At this point, the decision is made whether to roll back the project. If it is to be rolled back, proceed to the Import Project to Demo step. If the project is not to be rolled back, the subworkflow returns to the workflow's Rework Code step.

**13.3. Import Project to Demo**—To roll back the project, the project and object definitions that existed prior to the migration are imported to Demo from the backup taken during the migration in the main workflow. If this import succeeds, go to the Delete Project step.

**13.4. Delete Project**—If the project did not exist on the destination instance before the migration, the project and object definitions that were migrated are deleted from the destination. If this test succeeds, go to the Copy Project to Destination step.

**13.5. Copy Project to Destination**—The project is copied from the Demo to the destination and is returned to its previous state.

**13.6. Return**—Upon completion, the subworkflow returns to the workflow's Rework Code step.

**14. Rollback to Production**—This step calls the PeopleSoft Project Rollback SubWorkflow.

The subworkflow steps operate as described in step **13. Rollback to Test**.

**15. Rework Code**—This step is performed if rework was immediately identified as required in the Code Review step, or after the Rollback to Test or Rollback to Production step. Upon completion, the package returns to the initial Code Review step.

**16. Unlock Objects (Failure)**—When a request is cancelled after failure in the Migrate to Test step, it goes to this step, which unlocks all the entities (both source and destination) in the project.

**17. Unlock Objects (Failure)**—When a request is cancelled after failure in the Migrate to Production step, it goes to this step, which unlocks all the entities (both source and destination) in the project.

**18. Close (Failure)**—The package line is closed with a failure status.

*Table 4-16, Table 4-17, and Table 4-18* show the source and destination environments for the workflow steps in the PeopleSoft Development Workflow w/ Project Rollback & Object Warning workflow.

*Table 4-16. PeopleSoft Development Workflow w/ Project Rollback & Object Warning workflow*

Workflow Step Name	Source Environment	Destination Environment
Check for Objects (before Migrate to Test)	PeopleSoft DEV instance	PeopleSoft TEST environment or environment group
Lock Objects (before Migrate to Test)	PeopleSoft DEV instance	PeopleSoft TEST environment or environment group
Migrate to Test	PeopleSoft DEV instance	PeopleSoft TEST environment or environment group
Unlock Objects (Success) (after Migrate to Test)	PeopleSoft DEV instance	PeopleSoft TEST environment or environment group
Check for Objects (before Migrate to Production)	PeopleSoft TEST instance	PeopleSoft PROD environment or environment group
Lock Objects (before Migrate to Production)	PeopleSoft TEST instance	PeopleSoft PROD environment or environment group
Migrate to Production	PeopleSoft TEST instance	PeopleSoft PROD environment or environment group
Unlock Objects (Success) (after Migrate to Production)	PeopleSoft TEST instance	PeopleSoft PROD environment or environment group
Rollback to Test	PeopleSoft DEV instance	PeopleSoft TEST environment or environment group

Table 4-16. PeopleSoft Development Workflow w/ Project Rollback & Object Warning workflow [continued]

Workflow Step Name	Source Environment	Destination Environment
Rollback to Production	PeopleSoft TEST instance	PeopleSoft PROD environment or environment group
Unlock Objects (Failure) (after Migrate to Test)	PeopleSoft DEV instance	PeopleSoft TEST environment or environment group
Unlock Objects (Failure) (after Migrate to Production)	PeopleSoft TEST instance	PeopleSoft PROD environment or environment group

Table 4-17. PeopleSoft Project Object Warning SubWorkflow

Workflow Step Name	Source Environment (*See the Note below)	Destination Environment (**See the Note below)
Check Object Type	Not specified	Not specified
Check for Objects	Not specified	Not specified
Objects Exist	Not specified	Not specified

Table 4-18. PeopleSoft Project Rollback SubWorkflow

Workflow Step Name	Source Environment (*See the Note below)	Destination Environment (**See the Note below)
Check Object Type	Not specified	Not specified
Import Project to Demo	PeopleSoft Demo for 8.1.x Not specified for 8.4.x, 8.44	Not specified
Delete Project	Not specified	Not specified
Copy Project to Destination	PeopleSoft Demo for 8.1.x Not specified for 8.4.x, 8.44	Not specified

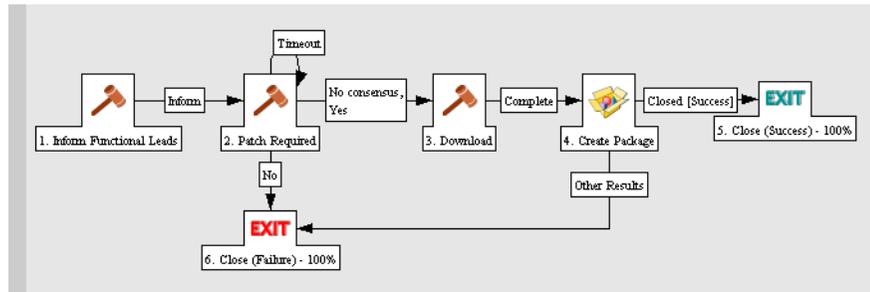
\* If there are no environments specified, this source environment makes use of the environment specified in the parent workflow step.

\*\* If there are no environments specified, this destination environment makes use of the environment specified in the parent workflow step.

## PeopleSoft Patch Request Workflow

The PeopleSoft Patch Request workflow is used by the PeopleSoft Patch request type to make sure that the required patches are downloaded and packages are created from them.

*Figure 4-18* shows the workflow diagram for the PeopleSoft Patch Request workflow. Steps in the workflow are described in the procedure that follows.



*Figure 4-18. PeopleSoft Patch Request workflow*

The workflow steps are described as follows:

- 1. Inform Functional Leads**—Requests are created using the Mercury IT Governance Workbench. Functional leads are informed of the available patches on the PeopleSoft site.
- 2. Patch Required**—The functional lead determines whether a patch is required.
- 3. Download**—If a patch is required, it is downloaded.
- 4. Create Package**—After the download is complete, a package is created.
- 5. Close (Success)**—If the package was created successfully, the request is closed with a status of success.
- 6. Close (Failure)**—If a patch was not required or if the package was not created successfully, the request is closed with a status of failure.

## PeopleSoft Patch Applicator Workflows

PeopleSoft Patch Applicator workflows include workflows and subworkflows for various PeopleSoft product releases:

- PeopleSoft Version 8.1.x
- PeopleSoft Version 8.4.x
- PeopleSoft Version 8.44

For more information about PeopleSoft Patch Applicator workflows, see [Chapter 5, \*Patch Applicator\*, on page 133](#).

## Report Types

This section provides reference information about the PeopleSoft-specific report types provided in the Extension. These report types are listed and defined in [Table 4-19](#).

### Overview of Report Types

The content or output of a report is controlled by what you specify in the fields for the report type, as described in this section. Some fields allow multiple entries. Some fields are hidden by default, but can be enabled to allow further control of the report output.

Execution of reports is driven by commands included within the report types. For more information about commands in the Mercury IT Governance Center environment, see the *Commands, Tokens, and Validations Guide and Reference*.

Report types are configured in the Workbench (in the shortcut bar, click **Configuration** and click the **Report Types** icon). Reports are run (submitted) from the standard interface (**Reports** menu).

#### **For More Information**

For information about running reports, see the *Reports Guide and Reference*.

### Reference Report Types

Reference (REFERENCE) report types cannot be edited, but you can copy and rename them and edit the copies to meet your needs. You can also use existing non-reference report types as is or configure them further to meet your needs.

## List of Report Types

*Table 4-19* lists the report types included in the Extension. Each is described in subsequent sections.

*Table 4-19. Report types included in the Extension*

Report Name	Definition
PeopleSoft Compare Custom Database Setup	Provides custom database comparisons of PeopleSoft databases.
PeopleSoft Comparison Report	Provides comparison of the contents of two PeopleSoft instances.
PeopleSoft Customized Objects Report	Provides information about PeopleSoft customized entities.
PeopleSoft File Comparison Report	Provides a comparison of PeopleSoft files.
PeopleSoft Object Customization Detail Report	Provides information about customizations of a PeopleSoft entity.
PeopleSoft Object Migrations Report	Lists migrations that were executed using any of the PeopleSoft object types.
PeopleSoft Patches Applied to an Environment	Lists patches that were applied to a specific environment.
PeopleSoft Project Migration Impact Analysis	Lists projects in specified environments that were affected by migrations from a “reference” environment that used the PeopleSoft Project (ORACLE) object type.
PeopleSoft Request Detail Report	Lists details of requests that use PeopleSoft request types, based on criteria you specify.
PeopleSoft Version Control - Baseline	Provides baseline version control for PeopleSoft entities.
PeopleSoft Version Control - Break Lock	Breaks a lock set by PeopleSoft Version Control.
PeopleSoft Version Control - Check In	Checks a file into PeopleSoft Version Control.
PeopleSoft Version Control - Check Out	Checks a file out of PeopleSoft Version Control.

### For More Information

For information about running reports, see the *Reports Guide and Reference*.

## PeopleSoft Compare Custom Database Setup Report

The PeopleSoft Compare Custom Database Setup Report is used to run custom database comparisons of PeopleSoft databases. Mercury-supplied data included with this report is used to compare PeopleSoft database objects. Custom comparison objects allow for the comparison of actual data within a database as well as within the data model.

Some standard features have been included with this report. For example, a filter can be applied to each object. Users are also given the option of viewing all objects in the category along with their associated attributes, or only objects and attributes that contain differences.

*Figure 4-19* shows the Submit Report window you use to specify the objects to be compared in the PeopleSoft Compare Custom Database Setup Report. *Table 4-20* provides definitions for the displayed fields.

Figure 4-19. PeopleSoft Compare Custom Database Setup Report

Table 4-20. PeopleSoft Compare Custom Database Setup Report field definitions

Field Name (*Required)	Definition
Comparison Name	Name you specify for the database comparison.
*Reference Environment	PeopleSoft environment to be compared with another environment.
*Compared Environment	PeopleSoft environment to be compared to the Reference Environment.
Reference AppCode	Application code used to override the parameters for the reference environment.
Compared AppCode	Application code used to override the parameters for the compared environment.
Only Report Differences	Whether or not to report only the differences between the two database setups.
Object 1 through Object 5	Object to be compared for the reference and compared environments. The drop-down lists show custom objects to be compared.
Object 1 Filter through Object 5 Filter	Filter applied to further define which objects are to be compared.

## PeopleSoft Comparison Report

The PeopleSoft Comparison Report introduced in release 6.0.0 allows you to compare PeopleSoft instances using the Mercury IT Governance Center standard interface. For example, it can be used to compare a previously customized PeopleSoft instance to a “vanilla” (non-customized) instance to which a new PeopleSoft patch has been applied.

See [System Requirements on page 26](#) for information about registry changes required to run this report.

The fields listed in the PeopleSoft Comparison Report vary according to the object types being compared. You can customize the fields based on the instructions in PeopleSoft documentation. By default, the report type uses only the basic parameters that are passed to the `pside` executable. Customers can make use of other parameters based on the instructions in PeopleSoft documentation.

[Figure 4-20](#) shows the Submit Report window you use to specify the instances to compare in the PeopleSoft Comparison Report. [Table 4-21](#) provides definitions for the displayed fields.

Figure 4-20. PeopleSoft Comparison Report

*Table 4-21. PeopleSoft Comparison Report field definitions*

Field Name (*Required)	Definition
*Source Environment	PeopleSoft environment to be compared to the Target Environment.
*Target Environment	PeopleSoft environment to be compared to the Source Environment.
*Project Name	Project whose object definitions are to be compared.
Object Types to Compare	Object types in the project that are to be compared.

## PeopleSoft Customized Objects Report

The PeopleSoft Customized Objects Report is used to list all PeopleSoft objects migrated through Mercury IT Governance Center into a specific environment. This report allows users to identify which objects have been customized since the last upgrade, and to identify which project was associated with the object at the time of customization.

The PeopleSoft Customized Objects Report lists the following fields:

- PeopleSoft Project Object Type
- PeopleSoft Project Object Name
- PeopleSoft Project Name
- Pkg Number
- Seq
- Object Type Used for Migration
- Last Migration Date

*Figure 4-21* shows the Submit Report window you use to specify the objects to be included in the PeopleSoft Customized Objects Report. *Table 4-22* provides definitions for the displayed fields.

The screenshot shows a web-based interface for submitting a report. The window title is "MERCURY" and the subtitle is "Submit Report: PeopleSoft Customized Objects Report". The interface includes a "Report Parameters" section with the following fields: "Environment Name", "PeopleSoft Object Type", "Transaction Date From", and "Transaction Date To". There are "Submit" and "Cancel" buttons at the top right and bottom right of the window. A "Restore Default" button is located in the "Report Parameters" section. The window also has a "Close Window" button in the top right corner.

*Figure 4-21. PeopleSoft Customized Objects Report*

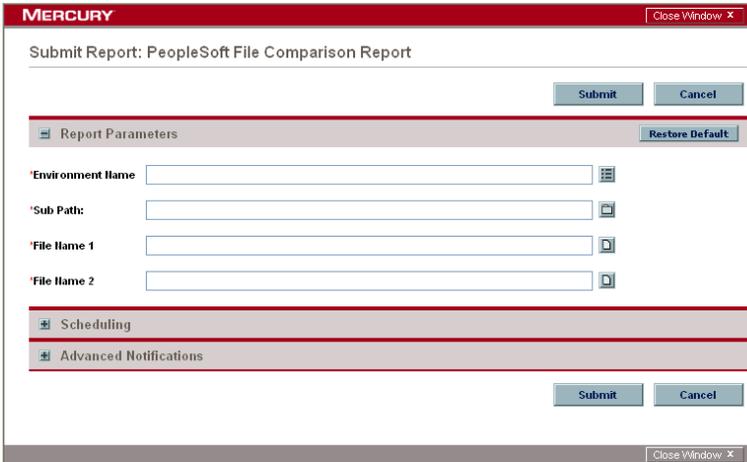
*Table 4-22. PeopleSoft Customized Objects Report field definitions*

Field Name (*Required)	Definition
*Environment Name	Environment containing the customized objects on which you want to report.
PeopleSoft Object Type	PeopleSoft object types on which to report.
Transaction Date From	Objects migrated on this date or later.
Transaction Date To	Objects migrated on this date or earlier.

## PeopleSoft File Comparison Report

The PeopleSoft File Comparison Report compares two ASCII source code files. This allows users to be able to identify differences between two COBOL programs, SQR files, or Data Mover scripts by providing a side-by-side comparison of the files. The output for this report is available in the execution log.

*Figure 4-22* shows the Submit Report window you use to specify the files to be compared by running the PeopleSoft File Comparison Report. *Table 4-23* provides definitions for the displayed fields.



*Figure 4-22. PeopleSoft File Comparison Report*

*Table 4-23. PeopleSoft File Comparison Report field definitions*

Field Name (*Required)	Definition
*Environment Name	Environment containing the files you want to compare.
*Sub Path	Sub path (relative to base path of the Environment Name) to the files.
*File Name 1	First file name to be used in the comparison.
*File Name 2	Second file name to be used in the comparison.

## PeopleSoft Object Customization Detail Report

The PeopleSoft Object Customization Detail Report lists all changes or customizations applied to a PeopleSoft online entity. This report provides users with a detailed historical view so they can determine what changes have been made to that particular entity and what project it was associated with when the changes were made.

The PeopleSoft Object Customization Detail Report lists the following fields:

- PeopleSoft Project Object Name
- PeopleSoft Project Object Type
- PeopleSoft Project Name
- Change Control Action
- Last Migration Date
- Operator Id
- Comment

The historical information gathered for the PeopleSoft online entities is retrieved from PeopleSoft tables. This information consists of comments PeopleSoft users enter whenever online objects are changed. (Comments are recorded only for changes made directly by users to a given instance, not for project copies.)

PeopleSoft Application Designer has an option to turn on change control. This prompts users to enter comments whenever they make changes to objects, by forcing them to lock the objects prior to making their changes.

To turn on change control, go to the **Tools > Change Control > Administrator** menu, and turn on these system-wide options:

- **Use change control locking**
- **Use change control history**

With both of these options on, whenever changes are made to the projects, users are prompted for a comment before saving their changes.

Selecting the **Use change control history only** option is also acceptable but users have to manually use the **Insert comment** option when they make changes, because they are not automatically prompted to do so.

Figure 4-23 shows the Submit Report window you use to specify the entity whose change history you want to review in the PeopleSoft Object Customization Detail Report output. Table 4-24 provides definitions for the displayed fields.

Figure 4-23. PeopleSoft Object Customization Detail Report

Table 4-24. PeopleSoft Object Customization Detail Report field definitions

Name (*Required)	Definition
*Environment Name	Environment containing the objects whose change history you want.
*PeopleSoft Project Object Name	PeopleSoft online object for which you want to obtain a change history.



Note

When this report is run from Mercury IT Governance Center on an Oracle 9i database, and PeopleSoft is on an Oracle 8i database, the following error is generated:

ORA-02064 - Distributed operation not supported.

You can resolve the problem in one of two ways:

- In table PSCHGCTHIST, convert the datatype from LONG to LOB.
- Upgrade the PeopleSoft database from 8i to 9i.

## PeopleSoft Object Migrations Report

The PeopleSoft Object Migrations Report provides a list of migrations that were executed using any of the PeopleSoft object types based on certain parameters. The report lists the following fields and their values for the selected object type:

- Pkg Number
- Seq
- Object
- App Code
- Workflow Step
- Last Migration Date
- User
- Dest Environment

*Figure 4-24* shows the Submit Report window you use to specify the object type and migrations to be used to generate the PeopleSoft Object Migrations Report output. *Table 4-25* provides definitions for the displayed fields.

The screenshot shows the 'MERCURY' window titled 'Submit Report: PeopleSoft Object Migrations Report'. The window has a 'Close Window X' button in the top right corner. Below the title bar, there are 'Submit' and 'Cancel' buttons. The main content area is divided into three sections: 'Report Parameters', 'Scheduling', and 'Advanced Notifications'. The 'Report Parameters' section is currently active and contains the following fields: 'PeopleSoft Object Type' (a dropdown menu), 'PeopleSoft Object Name' (a text input field), 'App Code' (a text input field), 'Destination Environment' (a dropdown menu), 'Transaction Date From' (a date input field with a calendar icon), 'Transaction Date To' (a date input field with a calendar icon), 'Performed by' (a text input field with a user selection icon), and 'Order by' (a dropdown menu). The 'Scheduling' and 'Advanced Notifications' sections are currently collapsed. At the bottom of the window, there are 'Submit' and 'Cancel' buttons and a 'Close Window X' button.

*Figure 4-24. PeopleSoft Object Migrations Report*

*Table 4-25. PeopleSoft Object Migrations Report field definitions*

Field Name (*Required)	Definition
*PeopleSoft Object Type	PeopleSoft object type used for the migrations to be listed.
PeopleSoft Object Name	Name of the migrated PeopleSoft object.
App Code	PeopleSoft Application Code.
Destination Environment	Name of the destination environment to which objects were migrated.
Transaction Date From	Migrations on this date or later.
Transaction Date To	Migrations on this date or earlier.
Performed by	User who performed the migration.
Order by	Criterion to use to sort the migrations in the report: <b>PeopleSoft Object Name, Destination Environment, or Package.</b>

## PeopleSoft Patches Applied to an Environment Report

The PeopleSoft Patches Applied to an Environment Report provides a list of patches that were applied to a specific environment. The report lists the following fields and their values for the selected object type:

- Pkg Number
- Seq
- PeopleSoft Product
- Product Category
- Product Line
- Product Release
- Report ID
- Patch Files
- App Code
- Step Name
- Transaction Date
- User

*Figure 4-25* shows the Submit Report window you use to specify the patches to include in the output of the PeopleSoft Patches Applied to an Environment Report. *Table 4-26* provides definitions for the displayed fields.

The screenshot shows a web-based form titled "Submit Report: PeopleSoft Patches Applied to an Environment" within the "MERCURY" application. The form is organized into several sections:

- Report Parameters:** This section contains the following fields:
  - PeopleSoft Environment: (text input)
  - Patch Workflow: (text input)
  - Patch Object Type: (text input)
  - PeopleSoft Product: (text input)
  - Product Category: (text input)
  - Product Line: (text input)
  - Product Release: (text input)
  - Report ID: (text input)
  - Patch File(s): (text input)
  - App Code: (text input)
  - Transaction Date From: (text input with calendar icon)
  - Transaction Date To: (text input with calendar icon)
  - Performed by: (text input with user selection icon)
  - Order by: (dropdown menu)
- Scheduling:** (Section header)
- Advanced Notifications:** (Section header)

Buttons for "Submit" and "Cancel" are located at the top right and bottom right of the form area. A "Close Window" button is visible in the top right corner of the application window.

Figure 4-25. PeopleSoft Patches Applied to an Environment Report

Table 4-26. PeopleSoft Patches Applied to an Environment Report field definitions

Field Name (*Required)	Definition
*PeopleSoft Environment	Name of the PeopleSoft environment to which patches were applied.
*Patch Workflow	Name of the workflow used to apply the patch.
*Patch Object Type	Name of the patch object type used to apply the patch.
PeopleSoft Product	PeopleSoft product impacted by the patch.
Product Category	Product category impacted by the patch.
Product Line	Product line impacted by the patch.
Product Release	Product release impacted by the patch.
Report ID	Name of the patch.
Patch File(s)	Files included in the patch.
App Code	PeopleSoft Application Code.
Transaction Date From	Patches applied on this date or later.
Transaction Date To	Patches applied on this date or earlier.
Performed by	User who applied the patches.
Order by	Criterion to use to sort the patches in the report: <b>PeopleSoft Product, Product Category, Product Line, Product Release, Report ID, or Package Line.</b>

## PeopleSoft Project Migration Impact Analysis Report

The PeopleSoft Project Migration Impact Analysis Report describes which projects in its Environments to Consider field have been affected by migrations from the environment in the Reference Environment Name field. The report describes only the migrations that used the PeopleSoft Project (ORACLE) object type (or a copy of it that could be customized, as long as the copy has the same fields in the same order).

The report lists the following fields:

- Pkg Number
- Seq
- Environment Name
- Impacted Project Name
- Last Migration Date

To use the report, create a view in the Mercury IT Governance schema for all PeopleSoft instances:

```
CREATE OR REPLACE VIEW kacc_psft_psprojectitems_v
(environment_name
, projectname
, objecttype
, objectid1
, objectvalue1
, objectid2
, objectvalue2
, objectid3
, objectvalue3
, objectid4
, objectvalue4
, nodetype
)
AS
(select
'PeopleSoft environment1', projectname,
objecttype,objectid1,objectvalue1,objectid2,objectvalue2,
objectid3,objectvalue3,objectid4,objectvalue4,nodetype
from psprojectitem@dblinktoenvironment1)
union
(select 'PeopleSoft environment2', projectname,
objecttype,objectid1,objectvalue1,objectid2,objectvalue2,
objectid3,objectvalue3,objectid4,objectvalue4,nodetype
from psprojectitem@dblinktoenvironment2)
```

For every PeopleSoft environment added for the impact analysis, you need to add a union to the view. This is the same view utilized by the Object Warning functionality of the PeopleSoft Development Workflow w/ Project Rollback & Object Warning workflow (see [PeopleSoft Development Workflow w/ Project Rollback & Object Warning Workflow](#) on page 89).

[Figure 4-26](#) shows the Submit Report window you use to specify the environments and object types used to generate the PeopleSoft Project Migration Impact Analysis Report. [Table 4-27](#) provides definitions for the displayed fields.

[Figure 4-26](#). PeopleSoft Project Migration Impact Analysis Report

[Table 4-27](#). PeopleSoft Project Migration Impact Analysis Report field definitions

Field Name (*Required)	Definition
*Project Name	Name of the PeopleSoft project to be analyzed.
*Reference Environment Name	PeopleSoft reference environment.
*PeopleSoft Object Type	<b>PeopleSoft Project (ORACLE)</b> or your copied and customized similar object type. If any other object type is selected, no data will be found.
*Environments To Consider	Impacted PeopleSoft environments.

## PeopleSoft Request Detail Report

The PeopleSoft Request Detail Report provides a detailed list of Mercury IT Governance Center requests that utilize PeopleSoft request types based on criteria you specify.

*Figure 4-27* shows the Submit Report window you use to specify which requests you want to appear in the PeopleSoft Request Detail Report and which information you want to appear for each request. *Table 4-28* provides definitions for the displayed fields.

The screenshot shows the 'Submit Report: PeopleSoft Request Detail Report' window. At the top, there are 'Submit' and 'Cancel' buttons. Below is a 'Report Parameters' section with a 'Restore Default' button. The parameters are organized into two columns:

- Left Column:** Request Numbers, PeopleSoft Request Type, Status, Assigned To, Created By, Department, Workflow, Contact, Creation Date From, Last Update Date From, Description Contains, Report Title (set to 'PeopleSoft Request Detail Report').
- Right Column:** Include Closed Requests (Yes/No), Priority, Assigned To Group, Request Sub Type, Application, Request Group, Company Name, Creation Date To, Last Update Date To, Order By (set to 'Request Number').

Below these are several 'Show' and 'Hide' options with radio buttons:

- Show Header Fields: Yes/No
- Hide Prompts for Empty Fields: Yes/No
- Show Field Audit History: Yes/No
- Show Notes: Yes/No
- Show Status: Yes/No
- Show Detail Fields: Yes/No
- Show Contents of Table Fields: Yes/No
- Filter Notes: Show all notes
- Show References: Yes/No

At the bottom, there are sections for 'Scheduling' and 'Advanced Notifications', followed by 'Submit' and 'Cancel' buttons.

Figure 4-27. PeopleSoft Request Detail Report

Table 4-28. PeopleSoft Request Detail Report field definitions

Field Name (*Required)	Definition
Request Numbers	Request numbers.
*Include Closed Requests	Whether or not to include requests that have been closed or cancelled.
PeopleSoft Request Type	Request type on which the report can be run.
Status	Requests having this status.
Priority	Requests having this priority,
Assigned To	Requests assigned to this user.
Assigned To Group	Requests assigned to this group.
Created By	Requests created by this user.
Request Sub Type	Requests having this sub-type.
Department	Requests logged against this department.
Application	Requests that are associated with this application.
Workflow	Requests that are associated with this workflow.
Request Group	Requests using request group(s) <b>Customization, Upgrade, or Setup.</b>
Contact	Requests having this contact name.
Company Name	Requests having this company name.
Creation Date From	Requests created on this date or later.
Creation Date To	Requests created on this date or earlier.
Last Update Date From	Requests last updated on this date or later.
Last Update Date To	Requests last updated on this date or earlier.
Description Contains	Requests having this description text string.
Report Title	Title of the report.
*Order By	Criterion to use to sort the requests in the report: <b>Application, Assigned To, Created By, Creation Date, Department, Last Update Date, Priority, Request Group, Request Number, Request Sub Type, Request Type, or Status.</b>
*Show Header Fields	Whether or not the report will show the full header fields of each request.
*Show Detail Fields	Whether or not the report will show the detail fields of each request.
*Hide Prompts for Empty Fields	Whether or not the report will hide prompts that have empty fields.

*Table 4-28. PeopleSoft Request Detail Report field definitions [continued]*

Field Name (*Required)	Definition
*Show Contents of Table Fields	Whether or not the report will show table fields for requests that have them.
*Show Field Audit History	Whether or not the report will show the transaction history of each request.
*Show Notes	Whether or not the report will show the notes attached to each request.
*Filter Notes	Whether the report will show all notes or only user notes.
*Show Status	Whether or not the report will show the workflow steps and current step status for each request.
*Show References	Whether or not the report will show the references associated with each request.

## PeopleSoft Version Control - Baseline Report

The PeopleSoft Version Control - Baseline Report allows users in the PS - Administrator security group to baseline online entities at their current state before developers start using Mercury IT Governance Center Version Control functionality to manage their development process. The following fields are displayed in the report output:

- Start Time
- Finish Time
- Number of files checked in
- User

This report is run from the standard interface menu option **PeopleSoft > Version Control > Baseline**. For more details about running this report, see [Creating a Baseline for Online Objects on page 174](#).

[Figure 4-28](#) shows the Submit Report window you use to specify what you want to baseline using the PeopleSoft Version Control - Baseline Report. [Table 4-29](#) provides definitions for the displayed fields.

Figure 4-28. PeopleSoft Version Control - Baseline Report

*Table 4-29. PeopleSoft Version Control - Baseline Report field definitions*

Field Name (*Required)	Definition
*Development Environment	Name of the development environment for which a baseline is to be established.
*Version Control Application	Application Code of the Version Control environment.
*Baseline	Whether to baseline <b>All Objects</b> or <b>Only Custom Objects</b> .
*Log Message	Log message you enter to describe the baseline.
Description	Description you enter for the files to be baselined.

## PeopleSoft Version Control - Break Lock Report

The PeopleSoft Version Control - Break Lock Report allows users in the PS - Administrator security group or the PS - Developer security group to break locks on entities checked out of the Mercury IT Governance Center Version Control repository. The following fields are displayed in the report output:

- PeopleSoft Project Object Type
- PeopleSoft Project Object Name
- Filename
- Revision
- Locked Status
- Last Updated By
- Last Updated Date
- Description
- Log Message
- Reference Id
- Reference Type

This report is run from the standard interface menu option **PeopleSoft > Version Control > Break Lock**. For more details about running this report, see [Breaking Locks on Objects on page 189](#).

[Figure 4-29](#) shows the Submit Report window you use to specify the entities for which you want the PeopleSoft Version Control - Break Lock Report to break locks. [Table 4-30](#) provides definitions for the displayed fields and for some fields that are hidden by default.

The screenshot displays a web-based form titled "Submit Report: PeopleSoft Version Control - Break Lock" within a "MERCURY" window. The form is organized into several sections:

- Report Parameters:** A section with a "Restore Default" button. It contains the following fields:
  - \*Version Control Application:** A text input field.
  - \*Development Environment:** A text input field.
  - \*Online Object?:** Radio buttons for "Yes" (selected) and "No".
  - \*Break Lock on:** A dropdown menu currently set to "Entire Project".
  - Project Name:** A text input field.
  - Project Revision Number:** A text input field.
  - \*Break Lock on Current Rev of All Objs in Project?:** Radio buttons for "Yes" (selected) and "No".
  - Project Object Type:** A text input field.
  - Project Object Name(s):** A text input field.
  - Filename(s):** A text input field.
- Scheduling:** A section with a plus icon and a text input field.
- Advanced Notifications:** A section with a plus icon and a text input field.

Buttons for "Submit" and "Cancel" are located at the top right and bottom right of the form area. The window title bar includes "MERCURY" and a "Close Window X" button.

Figure 4-29. PeopleSoft Version Control - Break Lock Report

Table 4-30. PeopleSoft Version Control - Break Lock Report field definitions

Field Name (*Required)	Definition
*Version Control Application	Vault where entities are stored.
*Development Environment	Development environment for which entities are to have locks broken.
*Online Object	<b>Yes</b> if breaking the lock on a PeopleSoft online object. <b>No</b> if breaking the lock on a PeopleSoft file.
*Break Lock on	Whether to break the lock on the <b>Entire Project</b> or on <b>Individual Project Objects Only</b> .
Project Name	Name of the project containing the online object for which to break locks.
Project Revision Number	Revision of project for which to break locks.
Project Object Type	Project object type for which to break locks.
*Break Lock on Current Rev of All Objs in Project	Whether or not to break the lock on the current revision of all objects in the project.
Project Object Name(s)	Project object name(s) for which to break locks.
Filename(s)	Name(s) of PeopleSoft files for which to break locks.
Version Control Env (Hidden by default)	Name of the Version Control environment. Default is <b>PeopleSoft VC</b> .

## PeopleSoft Version Control - Check In Report

The PeopleSoft Version Control - Check In Report allows users in the PS - Developer security group to check objects into the Mercury IT Governance Center Version Control repository. The following fields are displayed in the report output:

- PeopleSoft Project Object Type
- PeopleSoft Project Object Name
- Filename
- Revision
- Locked Status
- Last Updated By
- Last Updated Date
- Description
- Log Message
- Reference Id
- Reference Type

This report is run from the standard interface menu option **PeopleSoft > Version Control > Check In**. For more details about running this report, see [Checking In Objects on page 176](#).

[Figure 4-30](#) shows the Submit Report window you use to specify the objects you want the PeopleSoft Version Control - Check In Report to check in. [Table 4-31](#) provides definitions for the displayed fields and for some fields that are hidden by default.

**MERCURY** Close Window X

Submit Report: PeopleSoft Version Control - Check In

**Submit** **Cancel**

---

**Report Parameters** **Restore Default**

\*Development Environment:

\*Version Control Application:

\*Check if Current User Has Lock On Objects?  Yes  No

\*Online Object?  Yes  No

\*Project Name:

\*Check In:

Project Object Type:

Project Object Name(s):

Development Sub Path:

Filename:

\*Log Message:

\*Reference ID:

Description:

---

**Scheduling**

---

**Advanced Notifications**

**Submit** **Cancel**

Close Window X

Figure 4-30. PeopleSoft Version Control - Check In Report

Table 4-31. PeopleSoft Version Control - Check In Report field definitions

Field Name (*Required)	Definition
*Development Environment	Development environment containing object(s) to be checked into Version Control.
*Version Control Application	Vault where objects will be stored.
*Check If Current User Has Lock On Objects	<b>Yes</b> if objects are to be checked in only if the current user has a lock on them. <b>No</b> if objects are to be checked regardless of whether the current user has a lock on them.
*Online Object	<b>Yes</b> if checking in a PeopleSoft online object. <b>No</b> if checking in a PeopleSoft file.
*Project Name	Name of the project containing the online object to check in.
*Check In	Whether to check in the <b>Entire Project</b> or <b>Individual Project Objects Only</b> .
Project Object Type	Project object type to check in.
Project Object Name(s)	Project object name(s) to check in.
Development Sub Path	Sub Path for the PeopleSoft file.
Filename	Name of the PeopleSoft file to check in.
*Log Message	Log message you enter to describe the check-in.
*Reference ID	Mercury IT Governance Center package associated with the entity to be checked in.
Description	Description you enter for the object to be checked in.
Version Control Env (Hidden by default)	Name of the Version Control environment. Default is <b>PeopleSoft VC</b> .
File Type (Hidden by default)	Type of flat file. Default is <b>ASCII</b> .
File Location (Hidden by default)	Location of flat file. Default is <b>Server</b> .
Initial Rev Number (Hidden by default)	Initial revision number. Default is <b>1.0</b> .
Revision Number (Hidden by default)	Revision number to be assigned to the object.
Reference Type Code (Hidden by default)	Reference type code. Default is <b>1</b> for packages.

## PeopleSoft Version Control - Check Out Report

The PeopleSoft Version Control - Check Out Report allows users in the PS - Developer security group to check objects out of the Mercury IT Governance Center Version Control repository. The following fields are displayed in the report output:

- PeopleSoft Project Object Type
- PeopleSoft Project Object Name
- Filename
- Revision
- Locked Status
- Last Updated By
- Last Updated Date
- Description
- Log Message
- Reference Id
- Reference Type

This report is run from the standard interface menu option **PeopleSoft > Version Control > Check Out**. For more details about running this report, see [Checking Out Objects on page 182](#).

[Figure 4-31](#) shows the Submit Report window you use to specify the objects you want the PeopleSoft Version Control - Check Out Report to check out. [Table 4-32](#) provides definitions for the displayed fields and for some fields that are hidden by default.

Figure 4-31. PeopleSoft Version Control - Check Out Report

Table 4-32. PeopleSoft Version Control - Check Out Report field definitions

Field Name (*Required)	Definition
*Version Control Application	Vault where objects are stored.
*Development Environment	Development environment to which object(s) are to be checked out.
*Lock Objects	Whether or not to lock objects as they are checked out.
*Online Object	<b>Yes</b> if checking out a PeopleSoft online object. <b>No</b> if checking out a PeopleSoft file.
*Check Out	Whether to check out the <b>Entire Project</b> or <b>Individual Project Objects Only</b> .
Project Name	Name of the project containing the online object to check out.
Project Revision Number	Revision of the project for which to check out objects.
*Check Out Current Rev of All Objects in Project	<b>Yes</b> if you want to check out the latest revision of all of this project's objects, wherever the latest revision is found in the entire repository. <b>No</b> if you want to check out all of this project's objects at the revision the objects had when the project was at the revision specified in the Project Revision Number field.
Project Object Type	Project object type to check out.
Project Object Name(s)	Name(s) of the project object(s) to check out.
Filename(s)	Name(s) of the PeopleSoft files to check out.
Development Sub Path	Sub Path to which the file will be checked out.
Overwrite File if Exists	Whether or not to overwrite any existing copy of the file in the Development environment.
Version Control Env (Hidden by default)	Name of the Version Control environment. Default is <b>PeopleSoft VC</b> .
File Location (Hidden by default)	Location of the flat file. Default is <b>Server</b> .

## Special Commands

The Extension includes some special commands related to operation and migration. [Table 4-33](#) lists and defines these special commands.

### For More Information

For information about using and configuring commands, see the *Commands, Tokens, and Validations Guide and Reference*.

*Table 4-33. Special commands included in the Extension*

Special Command Name	Definition
ksc_psft_check_for_locked_objects	Objects contained in the project being migrated are checked on in both source and destination environments. If they are locked, the objects and the operator ID locking them as well as an error message are displayed.
ksc_psft_list_objects	Objects contained in the project being migrated are listed in the package line execution log.
ksc_psft_lock_objects	To prevent any overwriting while the project copy is in progress, objects contained in the project being migrated are locked on both source and destination databases.
ksc_psft_unlock_objects	Objects contained in the project being migrated are unlocked on both source and destination databases.
ksc_psft_run_sql_oracle	Executes an Oracle script against a PeopleSoft database using JDBC.

## Tokens

While configuring certain features in Mercury IT Governance Center, it is often necessary to refer to information that is undefined until it is used in a specific context. Instead of generating objects that are valid only in specific contexts, variables can be used to create general objects that can be applied to a variety of contexts. These variables are called tokens.

Tokens can be used in (but are not limited to) the following Mercury IT Governance entities:

- Object types
- Request types
- Validation commands and SQL statements
- Report types
- Workflow executions and notifications
- Workflow steps

Extension-specific tokens are available to be used only after you install the Extension. [Table 4-34](#) describes the standard environment tokens included with the Extension.

### **For More Information**

For information about using and configuring tokens, see the *Commands, Tokens, and Validations Guide and Reference*.

*Table 4-34. Tokens included in the Extension*

Prefix	Token	Description
ENV.AC	PS_USER_NAME	PeopleSoft User Name
ENV.AC	PS_PASSWORD	PeopleSoft Password
ENV.AC	PS_DATABASE_TYPE	PeopleSoft Database Type
ENV.AC	PS_BOOT_USER_NAME	Bootstrap User Name
ENV.AC	PS_BOOT_PASSWORD	Bootstrap Password
ENV.AC	PS_HOME	PeopleSoft Home Directory
ENV.AC	PS_MIGRATION_PATH	Migration Path
ENV.AC	PS_ENV_TYPE	Environment Type
ENV.AC	PS_PSFT_VERSION	PeopleSoft Version
USR.AC	PS_OPERATOR_ID	PeopleSoft Operator ID
USR.AC	PS_PASSWORD_ID	PeopleSoft Operator Password
USR.AC	PS_VC_USERNAME	Version Control Server Username
USR.AC	PS_VC_PASSWORD	Version Control Server Password

# Chapter 5 Patch Applicator

---

## In This Chapter:

- *Overview of Patch Applicator*
  - *Customization Considerations for All PeopleSoft Versions*
  - *Patches for PeopleSoft Version 8.1.x*
    - *PeopleSoft 8.1.x Patch Applicator Workflow*
    - *Requirements*
    - *Workflow Notification*
    - *Object Type*
    - *Applying Patches*
  - *Patches for PeopleSoft Version 8.4.x*
    - *PeopleSoft 8.4.x Patch Applicator Workflow*
    - *Requirements*
    - *Workflow Notification*
    - *Object Type*
    - *Applying Patches*
  - *Patches for PeopleSoft Version 8.44*
    - *PeopleSoft 8.44 Patch Applicator Workflow*
    - *Requirements*
    - *Workflow Notification*
    - *Object Type*
    - *Applying Patches*
-

## **Overview of Patch Applicator**

Patches available on the PeopleSoft Web site provide critical fixes for their applications. Maintaining a PeopleSoft application involves deploying and tracking these patches. Mercury Change Management Extension for PeopleSoft Enterprise includes PeopleSoft Patch Applicator, a useful tool for this kind of task.

PeopleSoft Patch Applicator for a particular PeopleSoft Version consists of the associated PeopleSoft Patch object type, PeopleSoft Patch Applicator workflow, and PeopleSoft Patch subworkflow. It can be used to apply patches involving Data Files, Direct DMS changes, COBOL programs, SQL scripts, SQR, Crystal Reports, nVision Layouts, and project files.

## Customization Considerations for All PeopleSoft Versions

Consider the following in using the Extension to apply patches to a PeopleSoft system at Version 8.1.x or later:

- Both the object type and workflow can be modified to apply patches to other instances. More instances can be added or existing ones can be removed.
- Additional security can be added if needed. The locks can also be removed if they are not needed.
- To apply patches at a scheduled time, the immediate steps in the workflow need to be changed to manual executions to allow for setting up a scheduled time for the step to run.
- A flag can be added to indicate whether any additional processing is needed to apply the patch. If this flag is set to **No**, the additional processing step is skipped.
- Notifications on the decision steps and manual execution steps alert users that a step needs to be executed.
- For UNIX batch servers, make sure the environment variables are set once the user logs onto the system. If one logon is used for all instances, add a line in the commands to set the environment variables before running any of the compilations. For example, in the Apply COBOL and SQR to Destination (UNIX) command, add the following instructions:

```
ksc_connect_dest_server
. DEST_ENV.DB_CONNECT_STRING <-enter the file that sets the
path in this variable in the Environment panel host tab
PS_HOME="DEST_ENV.AC.PS_HOME"
sh PKG.NUMBER PKGL.SEQ applypatch
\rm -rf PKG.NUMBER PKGL.SEQ *
if [ -f DEST_ENV.SERVER_BASE_PATH/PKGL.P.PS_LOG_FILE_NAME_
cobol_unix.log ]; then logfile=Y; else logfile=N; fi
echo $logfile
ksc_exit
```

## Patches for PeopleSoft Version 8.1.x

The following sections discuss the following subjects in connection with applying patches to PeopleSoft Version 8.1.x:

- PeopleSoft 8.1.x Patch Applicator workflow (and PeopleSoft 8.1.x Patch SubWorkflow)
- Requirements
- Workflow notification
- Object type
- Customization considerations
- Applying patches

### PeopleSoft 8.1.x Patch Applicator Workflow

The PeopleSoft 8.1.x Patch Applicator Workflow is used in the process of deploying a patch downloaded from PeopleSoft to one or more destination instances. For each instance, the workflow calls the PeopleSoft 8.1.x Patch SubWorkflow. The process of deploying the patch is the same for all destination instances. For configuration requirements to use the workflow, see [Requirements on page 140](#).

*Figure 5-1* shows the workflow diagram for the PeopleSoft 8.1.x Patch Applicator workflow. *Figure 5-2* shows the PeopleSoft 8.1.x Patch SubWorkflow. Steps in the workflow and subworkflow are described in the procedure that follows.

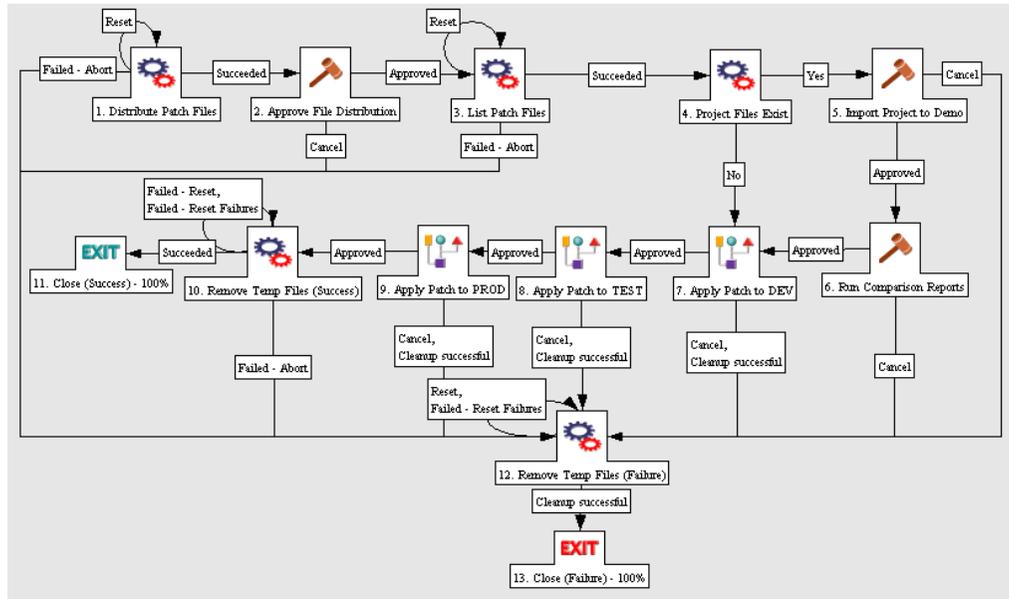


Figure 5-1. PeopleSoft 8.1.x Patch Applicator workflow

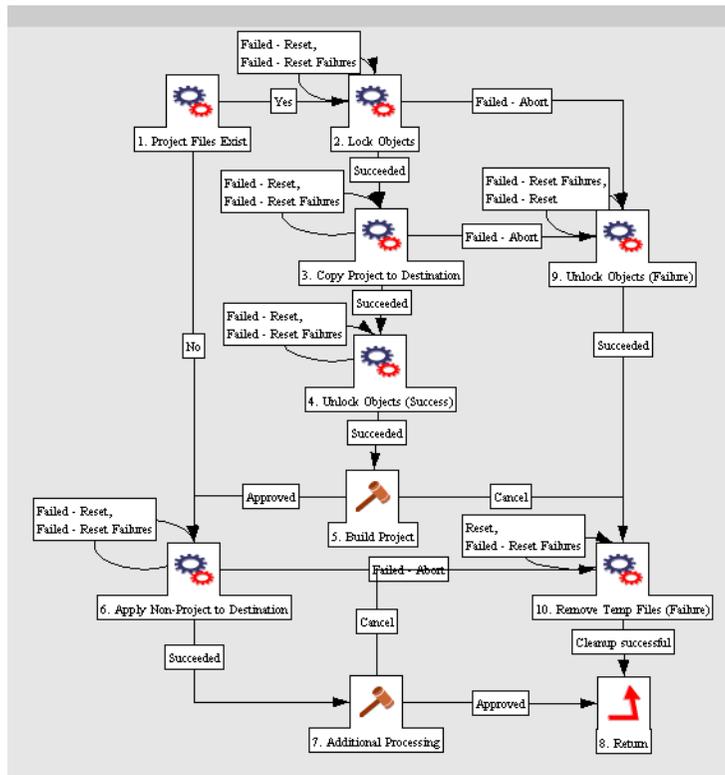


Figure 5-2. PeopleSoft 8.1.x Patch SubWorkflow

The workflow steps are described as follows:

**1. Distribute Patch Files**—A directory under the Patch Stage Base Path is created for the patch being applied. The patch files are distributed into different subdirectories under it depending on their file types, based on their file extensions. Any zipped files, including project files, are unzipped and distributed. A preliminary list of the file distribution is displayed in the execution log.

**2. Approve File Distribution**—The user verifies that the files were distributed correctly. The user also checks for file types that are not supported for migration or deployment—such files require manual handling in the Additional Processing step in the subworkflow.

**3. List Patch Files**—The final list of patch files is generated, displayed in the package line execution log, and placed in a `log` subdirectory under the patch folder. Project, non-project, and UNIX changes are identified for use later in the workflow to identify what type of processing must be performed in the object type execution on the destination.

**4. Project Files Exist**—The projects folder of the patch is checked to see whether it has any project files. If so, proceed to the Approve Demo Lock step; if not, go to the Apply Patch to DEV step.

**5. Import Project to Demo**—In this manual step, the user imports the patch project(s) into the Demo database.

**6. Run Comparison Reports**—In this step, the user runs comparison reports between the Demo and destination instances to evaluate impacts on customized objects and take any actions required before final migration.

**7. Apply Patch to DEV**—The PeopleSoft 8.1.x Patch SubWorkflow is called to deploy the patch to the DEV instance.

PeopleSoft 8.1.x Patch SubWorkflow steps:

**7.1. Project Files Exist**—This step determines whether the patch includes any project files. If so, proceed to the Lock Objects step. If not, go to the Apply Non-Project to Destination step.

**7.2. Lock Objects**—This step locks all the objects that are in patch projects in both the source and destination databases before migration. However, if any of these objects were previously locked by another process, an error is generated and the subworkflow goes to the Unlock Objects (Failure) step.

**7.3. Copy Project to Destination**—Projects are copied from the Demo database to the destination database, using the PeopleSoft command line interface. If there is a failure, the subworkflow goes to the Unlock Objects (Failure) step.

**7.4. Unlock Objects (Success)**—If the copy is successful, the objects on the source and destination are unlocked.

**7.5. Build Project**—The user builds the project at the destination.

**7.6. Apply Non-Project to Destination**—At the destination, this step deploys (migrates, or migrates and executes) all non-project changes for supported file types.

**7.7. Additional Processing**—In this step, the user performs any additional processing for file types that are not supported for automated deployment.

**7.8. Return**—This step returns to the parent workflow, where it goes to the Apply Patch to TEST step.

**7.9. Unlock Objects (Failure)**—After a failure in the Copy Project to Destination step, this step unlocks the project objects in the source and destination.

**7.10. Remove Temp Files (Failure)**—Any failure or cancellation during the subworkflow leads to this step, where all temporary and log files created in the deployment process are deleted, and the subworkflow goes to the Return step.

**8. Apply Patch to TEST**—The PeopleSoft 8.1.x Patch SubWorkflow is called to deploy the patch to the TEST instance. See the details of this subworkflow in the preceding Apply Patch to DEV step.

**9. Apply Patch to PROD**—The PeopleSoft 8.1.x Patch SubWorkflow is called to deploy the patch to the PROD instance. See the details of this subworkflow in the preceding Apply Patch to DEV step.

**10. Remove Temp Files (Success)**—This step removes temporary files generated during the migration and deployment of patches.

**11. Close (Success)**—Upon successful completion, the package line is closed.

**12. Remove Temp Files (Failure)**—If any step in the patch process fails or is cancelled, this step removes temporary files generated during the migration and deployment of patches.

**13. Close (Failure)**—At the conclusion of any failed process, close the package line.

## Requirements

To use the patch applicator, the requirements described in the following sections must be met:

### *Software Requirements*

The following software requirements must be met:

- PeopleTools version 8.1.x
- PeopleSoft-endorsed COBOL compiler
- Mercury IT Governance Center release 6.0

### *File Structure Requirements*

The file storage structure must be identical on all PeopleSoft instances except Demo. See [System Requirements on page 26](#).

### *Environment Requirements*

The following environments must be specified:

- PeopleSoft Patch Stage—the environment where all the patch files will be stored
- Demo instance of PeopleSoft—the environment where all the project changes will be applied
- Development instance of PeopleSoft
- Test instance of PeopleSoft
- Production instance of PeopleSoft

## Access Requirements

To log on to the environments, users need access to:

- sqlplus.exe
- pstools.exe
- psdmt.exe

## Workflow Setup Requirements

You must set up the PeopleSoft 8.1.x Patch Applicator workflow steps to include the source and destination environments listed in [Table 5-1](#).

Table 5-1. PeopleSoft 8.1.x Patch Applicator workflow setup

Workflow Step Name	Source Environment	Destination Environment
Distribute Patch Files	PeopleSoft Patch Stage	PeopleSoft Patch Stage
List Patch Files	PeopleSoft Patch Stage	
Project Files Exist		
Apply Patch to DEV	PeopleSoft Patch Stage	PeopleSoft DEV environment or environment group
Apply Patch to TEST	PeopleSoft Patch Stage	PeopleSoft TEST environment or environment group
Apply Patch to PROD	PeopleSoft Patch Stage	PeopleSoft PROD environment or environment group
Remove Temp Files (Success)	PeopleSoft Patch Stage	
Remove Temp Files (Failure)	PeopleSoft Patch Stage	PeopleSoft Demo

Set up the PeopleSoft 8.1.x Patch SubWorkflow steps to include the source and destination environments listed in [Table 5-2](#).

Table 5-2. PeopleSoft 8.1.x Patch SubWorkflow setup

Subworkflow Step Name	Source Environment	Destination Environment
Lock Objects	PeopleSoft Demo	** For all of the destination environments, see the note below.
Copy Projects to Dest	PeopleSoft Demo	
Unlock Objects (Success)	PeopleSoft Demo	
Apply Non-Project to Destination	* See the note below.	
Unlock Objects (Failure)	PeopleSoft Demo	
Remove Temp Files (Failure)	* See the note below.	

\* This source environment makes use of the environment specified in the parent workflow step. For example, if a subworkflow step is called by the Apply Patch to DEV step, the subworkflow uses the same source environment as the Apply Patch to DEV step, which is the PeopleSoft Patch Stage.

\*\* This destination environment makes use of the environment specified in the parent workflow step. For example, if any of the subworkflow steps are called by the Apply Patch to DEV step, the subworkflow uses the same destination environment as the Apply Patch to DEV step, which is the PeopleSoft DEV instance.

## Workflow Notification

You must select which users are notified for the following workflow steps:

- PeopleSoft 8.1.x Patch Applicator workflow steps
  - Distribute Patch Files
  - Approve File Distribution
  - Import Project to Demo
  - Run Comparison Reports
- PeopleSoft 8.1.x Patch SubWorkflow steps
  - Build Project
  - Additional Processing

## Object Type

*Figure 5-3* shows sample data for the PeopleSoft 8.1.x Patch object type when adding a package line. *Table 5-3* lists and defines the field names.

*Figure 5-3. PeopleSoft 8.1.x Patch object type sample data*

*Table 5-3. PeopleSoft 8.1.x Patch object type field definitions*

Field Name	Definition
PeopleSoft Product	PeopleSoft product impacted by the patch.
Product Category	Product category impacted by the patch.
Product Line	Product Line impacted by the patch.
Product Release	Product release impacted by the patch.
Report ID	Report ID that identifies the patch.
Patch File(s)	Files included in the patch.
Log File Name	Name to be used by patch log files.
Delete Source Files	Whether or not to delete patch files from the source.
File Location (Hidden by default)	File location: <b>Client</b> or <b>Server</b> .

Table 5-3. PeopleSoft 8.1.x Patch object type field definitions [continued]

Field Name	Definition
Log Exist (Hidden by default)	Flag indicating whether log files exist.
Project(s) (Hidden by default)	Name(s) of project(s), if any, contained in the applied patch.
NonProject Files Exist (Hidden by default. Was called NonAU Files Exist before release 6.0.0.)	Flag indicating whether Data, DMS, SQL Script, nVision Layout, or Crystal Reports files exist in the patch.
Cobol or SQR Files Exist (Hidden by default)	Flag indicating whether COBOL or SQR files exist in the patch.
PatchStage (Hidden by default)	Identifies which environment is the patch stage.
Project Files Exist (Hidden by default)	Flag indicating whether there are projects to be copied as part of the patch.
Sub Path (Hidden by default)	Sub Path for patch files. Defaults to "." for current directory.

## Applying Patches

To apply a patch to PeopleSoft Version 8.1.x:

1. Download the patch from the PeopleSoft Customer Connection Web site to the Mercury IT Governance Center environment defined as the Patch Stage environment.
2. In the Mercury IT Governance Workbench, generate a new package.
3. Select the PeopleSoft 8.1.x Patch Applicator workflow.
4. Click **New Line** to generate a new package line. This opens the Add Line window.
5. Select the PeopleSoft 8.1.x Patch object type. This dynamically updates the screen.
6. Choose the PeopleSoft application to which you want to apply the patch.

7. Choose the PeopleSoft Product from the drop-down list.
8. Choose the Product Category from the drop-down list.
9. Enter the Product Line and Product Release information.
10. Enter the Report ID of the patch.
11. In the Patch File(s) field, enter the patch file(s) included in the patch. Use the multi-file chooser for multiple files.
12. Enter the Log File Name for the patch. The default is the same as the Report ID.
13. Indicate whether or not you want the patch files deleted at the source.
14. Click **OK** to accept the package line fields and close the window.
15. Click **Submit** to submit the package line. The patch is processed through the PeopleSoft 8.1.x Patch Applicator workflow, which notifies users of required decisions, actions, and approvals. Users will be asked to:
  - a. Submit the package line.
  - b. Execute the file distribution step.
  - c. Verify file type distribution.
  - d. Approve any lock(s) for the Demo database, if necessary.
  - e. Run comparison reports.
  - f. Respond to the prompt to lock project objects.
  - g. Build project(s) for the destination instance.
  - h. Apply any additional processing needed that is not performed by the patch applicator.
16. Repeat [step 15](#) for all other destination instances.

## Patches for PeopleSoft Version 8.4.x

The following sections discuss the following subjects in connection with applying patches to PeopleSoft Version 8.4.x:

- PeopleSoft 8.4.x Patch Applicator Workflow (and PeopleSoft 8.4.x Patch SubWorkflow)
- Requirements
- Workflow notification
- Object type
- Customization considerations
- Applying patches

### PeopleSoft 8.4.x Patch Applicator Workflow

The PeopleSoft 8.4.x Patch Applicator Workflow is used in the process of deploying a patch downloaded from PeopleSoft to one or more destination instances. For each instance, the workflow calls the PeopleSoft 8.4.x Patch SubWorkflow. The process of deploying the patch is the same for all destination instances. For configuration requirements to use the workflow, see [Requirements on page 151](#).

*Figure 5-4* shows the workflow diagram for the PeopleSoft 8.4.x Patch Applicator workflow. *Figure 5-5* shows the PeopleSoft 8.4.x Patch SubWorkflow. Steps in the workflow and subworkflow are described in the procedure that follows.

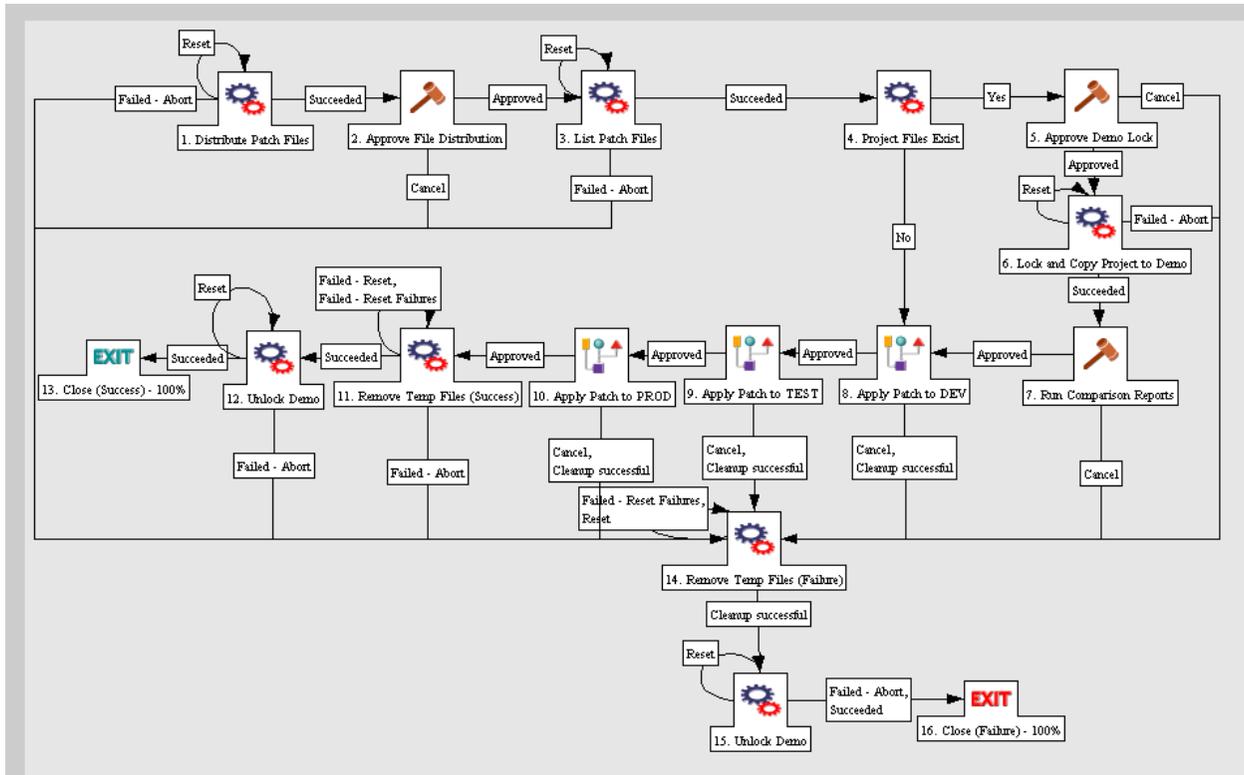


Figure 5-4. PeopleSoft 8.4.x Patch Applicator workflow

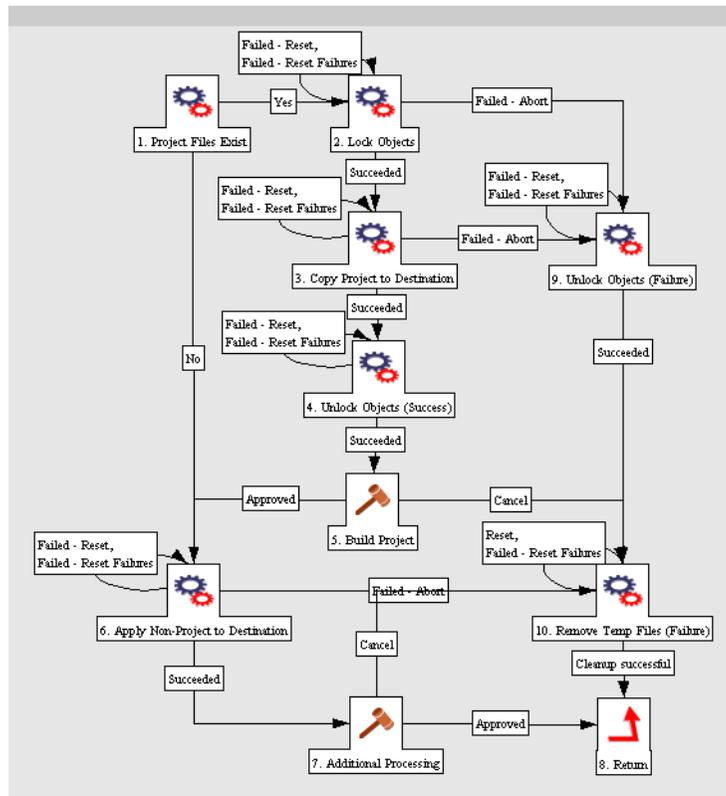


Figure 5-5. PeopleSoft 8.4.x Patch SubWorkflow

The workflow steps are described as follows:

- 1. Distribute Patch Files**—A directory under the Patch Stage Base Path is created for the patch being applied. The patch files are distributed into different subdirectories under it depending on their file types, based on their file extensions. Any zipped files, including project files, are unzipped and distributed. A preliminary list of the file distribution is displayed in the execution log.
- 2. Approve File Distribution**—The user verifies that the files were distributed correctly. The user also checks for file types that are not supported for migration or deployment—such files require manual handling in the Additional Processing step in the subworkflow.
- 3. List Patch Files**—The final list of patch files is generated, displayed in the package line execution log, and placed in a `log` subdirectory under the patch folder. Project, non-project, and UNIX changes are identified for use later in the workflow to identify what type of processing must be performed in the object type execution on the destination.

**4. Project Files Exist**—The projects folder of the patch is checked to see whether it has any project files. If so, proceed to the Approve Demo Lock step; if not, go to the Apply Patch to DEV step.

**5. Approve Demo Lock**—To keep anyone from overwriting data at the source or destination during the project copy process, all the objects in the project are locked in the Demo database before the project is copied. In this step, the user approves the lock.

**6. Lock and Copy Project to Demo**—The patch project files are copied to the Demo database. Objects in the patch project(s) that already exist in the Demo database are locked. If any objects have been locked by another process, an error is generated. Log files are created in the Report ID folder.

**7. Run Comparison Reports**—In this step, the user runs comparison reports between the Demo and destination instances to evaluate impacts on customized objects and take any actions required before final migration.

**8. Apply Patch to DEV**—The PeopleSoft 8.4.x Patch SubWorkflow is called to deploy the patch to the DEV instance.

PeopleSoft 8.4.x Patch SubWorkflow steps:

**8.1. Project Files Exist**—This step determines whether the patch includes any project files. If so, proceed to the Lock Objects step. If not, go to the Apply Non-Project to Destination step.

**8.2. Lock Objects**—This step locks all the objects that are in patch projects in both the source and destination databases before migration. However, if any of these objects were previously locked by another process, an error is generated and the subworkflow goes to the Unlock Objects (Failure) step.

**8.3. Copy Project to Destination**—Projects are copied from the Demo database to the destination database, using the PeopleSoft command line interface. If there is a failure, the subworkflow goes to the Unlock Objects (Failure) step.

**8.4. Unlock Objects (Success)**—If the copy is successful, the objects on the source and destination are unlocked.

**8.5. Build Project**—The user builds the project at the destination.

**8.6. Apply Non-Project to Destination**—At the destination, this step deploys (migrates, or migrates and executes) all non-project changes for supported file types.

**8.7. Additional Processing**—In this step, the user performs any additional processing for file types that are not supported for automated deployment.

**8.8. Return**—This step returns to the parent workflow, where it goes to the Apply Patch to TEST step.

**8.9. Unlock Objects (Failure)**—After a failure in the Copy Project to Destination step, this step unlocks the project objects in the source and destination.

**8.10. Remove Temp Files (Failure)**—Any failure or cancellation during the subworkflow leads to this step, where all temporary and log files created in the deployment process are deleted, and the subworkflow goes to the Return step.

**9. Apply Patch to TEST**—The PeopleSoft 8.4.x Patch SubWorkflow is called to deploy the patch to the TEST instance. See the details of this subworkflow in the preceding Apply Patch to DEV step.

**10. Apply Patch to PROD**—The PeopleSoft 8.4.x Patch SubWorkflow is called to deploy the patch to the PROD instance. See the details of this subworkflow in the preceding Apply Patch to DEV step.

**11. Remove Temp Files (Success)**—This step removes temporary files generated during the migration and deployment of patches.

**12. Unlock Demo**—This step unlocks the Demo database.

**13. Close (Success)**—Upon successful completion, the package line is closed.

**14. Remove Temp Files (Failure)**—If any step in the patch process fails or is cancelled, this step removes temporary files generated during the migration and deployment of patches.

**15. Unlock Demo**—This step unlocks the Demo database.

**16. Close (Failure)**—At the conclusion of any failed process, close the package line.

## Requirements

To use the patch applicator, the requirements described in the following sections must be met:

### *Software Requirements*

The following software requirements must be met:

- PeopleTools version 8.4.x
- PeopleSoft-endorsed COBOL compiler
- Mercury IT Governance Center release 6.0

### *File Structure Requirements*

The file storage structure must be identical on all PeopleSoft instances except Demo. See [System Requirements on page 26](#).

### *Environment Requirements*

The following environments must be specified:

- PeopleSoft Patch Stage—the environment where all the patch files will be stored
- Demo instance of PeopleSoft—the environment where all the project changes will be applied
- Development instance of PeopleSoft
- Test instance of PeopleSoft
- Production instance of PeopleSoft

### *Access Requirements*

To log on to the environments, users need access to:

- `sqlplus.exe`
- `pside.exe`
- `psdmt.exe`

## ***Workflow Setup Requirements***

You must set up the PeopleSoft 8.4.x Patch Applicator workflow steps to include the source and destination environments listed in [Table 5-4](#).

*Table 5-4. PeopleSoft 8.4.x Patch Applicator workflow setup*

Workflow Step Name	Source Environment	Destination Environment
Distribute Patch Files	PeopleSoft Patch Stage	PeopleSoft Patch Stage
List Patch Files	PeopleSoft Patch Stage	
Project Files Exist		
Lock and Copy Project to Demo	PeopleSoft Patch Stage	PeopleSoft Demo
Apply Patch to DEV	PeopleSoft Patch Stage	PeopleSoft DEV environment or environment group
Apply Patch to TEST	PeopleSoft Patch Stage	PeopleSoft TEST environment or environment group
Apply Patch to PROD	PeopleSoft Patch Stage	PeopleSoft PROD environment or environment group
Remove Temp Files (Success)	PeopleSoft Patch Stage	
Remove Temp Files (Failure)	PeopleSoft Patch Stage	PeopleSoft Demo
Unlock Demo		

Set up the PeopleSoft 8.4.x Patch SubWorkflow steps to include the source and destination environments listed in [Table 5-5](#).

*Table 5-5. PeopleSoft 8.4.x Patch SubWorkflow setup*

Subworkflow Step Name	Source Environment	Destination Environment
Project Files Exist	* See the note below.	** For all of the destination environments, see the note below.
Lock Objects	PeopleSoft Demo	
Copy Projects to Dest	PeopleSoft Demo	
Unlock Objects (Success)	PeopleSoft Demo	
Apply Non-Project to Destination	* See the note below.	
Unlock Objects (Failure)	PeopleSoft Demo	
Remove Temp Files (Failure)	* See the note below.	

\* This source environment makes use of the environment specified in the parent workflow step. For example, if a subworkflow step is called by the Apply Patch to DEV step, the subworkflow uses the same source environment as the Apply Patch to DEV step, which is the PeopleSoft Patch Stage.

\*\* This destination environment makes use of the environment specified in the parent workflow step. For example, if any of the subworkflow steps are called by the Apply Patch to DEV step, the subworkflow uses the same destination environment as the Apply Patch to DEV step, which is the PeopleSoft DEV instance.

## Workflow Notification

You must select which users are notified for the following workflow steps:

- PeopleSoft 8.4.x Patch Applicator workflow steps
  - Distribute Patch Files
  - Approve File Distribution
  - Approve Demo Lock
  - Run Comparison Reports

- PeopleSoft 8.4.x Patch SubWorkflow steps
  - Build Project
  - Additional Processing

## Object Type

Figure 5-6 shows sample data for the PeopleSoft 8.4.x Patch object type when adding a package line. Table 5-6 contains field definitions.

Figure 5-6. PeopleSoft 8.4.x Patch object type sample data

Table 5-6. PeopleSoft 8.4.x Patch object type field definitions

Field Name	Definition
PeopleSoft Product	PeopleSoft Product impacted by the patch.
Product Category	Product category impacted by the patch.
Product Line	Product Line impacted by the patch.
Product Release	Product release impacted by the patch.
Report ID	Report ID identifying the patch.
Patch File(s)	Files included in the patch.

Table 5-6. PeopleSoft 8.4.x Patch object type field definitions [continued]

Field Name	Definition
Log File Name	Name to be used by patch log files.
Delete Source Files	Whether or not to delete patch files from the source.
File Location (Hidden by default)	File location: <b>Client</b> or <b>Server</b> .
Log Exist (Hidden by default)	Flag indicating whether log files exist.
Project(s) (Hidden by default)	Name(s) of project(s), if any, contained in the applied patch.
NonProject Files Exist (Hidden by default. Was called NonAU Files Exist before release 6.0.0.)	Flag indicating whether Data, DMS, SQL Script, nVision Layout, or Crystal Reports files exist in the patch.
Cobol or SQR Files Exist (Hidden by default)	Flag indicating whether COBOL or SQR files exist in the patch.
PatchStage (Hidden by default)	Identifies which environment is the patch stage.
Project Files Exist (Hidden by default)	Flag indicating whether there are projects to be copied as part of the patch.
Sub Path (Hidden by default)	Sub Path for patch files. Defaults to "." for current directory.

## Applying Patches

To apply a patch to PeopleSoft Version 8.4.x:

1. Download the patch from the PeopleSoft Customer Connection Web site to the Mercury IT Governance Center environment defined as the Patch Stage environment.
2. In the Mercury IT Governance Workbench, generate a new package.
3. Select the PeopleSoft 8.4.x Patch Applicator workflow.
4. Click **New Line** to generate a new package line. This opens the Add Line window.

5. Select the PeopleSoft 8.4.x Patch object type. This dynamically updates the screen.
6. Choose the PeopleSoft application to which you want to apply the patch.
7. Choose the PeopleSoft Product from the drop-down list.
8. Choose the Product Category from the drop-down list.
9. Enter the Product Line and Product Release information.
10. Enter the Report ID of the patch.
11. In the Patch File(s) field, enter the patch file(s) included in the patch. Use the multi-file chooser for multiple files.
12. Enter the Log File Name for the patch. The default is the same as the Report ID.
13. Indicate whether or not you want the patch files deleted at the source.
14. Click **OK** to accept the package line fields and close the window.
15. Click **Submit** to submit the package line. The patch is processed through the PeopleSoft 8.4.x Patch Applicator workflow, which notifies users of required decisions, actions, and approvals. Users will be asked to:
  - a. Submit the package line.
  - b. Execute the file distribution step.
  - c. Verify file type distribution.
  - d. Approve any lock(s) for the Demo database, if necessary.
  - e. Run comparison reports.
  - f. Respond to the prompt to lock project objects.
  - g. Build project(s) for the destination instance.
  - h. Apply any additional processing needed that is not performed by the patch applicator.
16. Repeat [step 15](#) for all other destination instances.

## Patches for PeopleSoft Version 8.44

Release 6.0.0 of the Extension adds support for PeopleSoft Version 8.44, including the following new entities:

- PeopleSoft 8.44 Patch object type
- PeopleSoft 8.44 Patch Applicator workflow.
- PeopleSoft 8.44 Patch SubWorkflow.

The following sections discuss the following subjects in connection with applying patches to PeopleSoft Version 8.44:

- PeopleSoft 8.44 Patch Applicator Workflow (and PeopleSoft 8.44 Patch SubWorkflow)
- Requirements
- Workflow notification
- Object type
- Customization considerations
- Applying patches

### PeopleSoft 8.44 Patch Applicator Workflow

The PeopleSoft 8.44 Patch Applicator Workflow is used in the process of deploying a patch downloaded from PeopleSoft to one or more destination instances. For each instance, the workflow calls the PeopleSoft 8.44 Patch SubWorkflow. The process of deploying the patch is the same for all destination instances. For configuration requirements to use the workflow, see [Requirements on page 162](#).

[Figure 5-7](#) shows the workflow diagram for the PeopleSoft 8.44 Patch Applicator workflow. [Figure 5-8](#) shows the PeopleSoft 8.44 Patch SubWorkflow. Steps in the workflow and subworkflow are described in the procedure that follows.



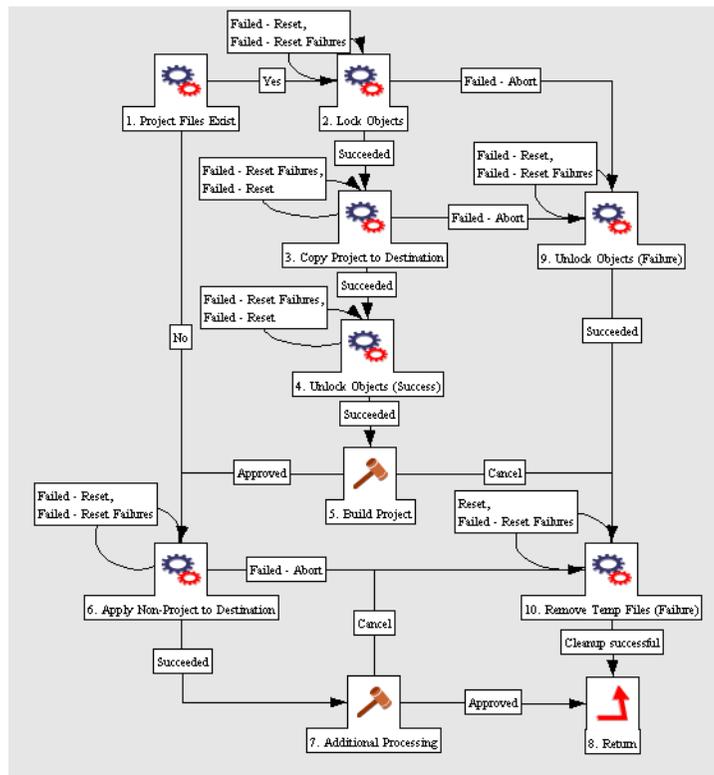


Figure 5-8. PeopleSoft 8.44 Patch SubWorkflow

The workflow steps are described as follows:

- 1. Create Patch Folder**—A patch folder is created in Patch Stage, based on the Report ID the user entered.
- 2. Run Patch Install Setup**—In this manual step, the user downloads the patch, unzips it, and runs the setup, specifying the Report ID folder name as the destination of the patch.
- 3. Approve File Distribution**—The user verifies that the files were distributed correctly. The user also checks for file types that are not supported for migration or deployment—such files require manual handling in the Additional Processing step in the subworkflow. The user must also delete any projects in the `projects` directory in the patch that are not listed in the patch instruction or that do not need to be copied to the destination.

- 4. List Supported Patch Files**—The final list of patch files is generated, displayed in the package line execution log, and placed in a `log` subdirectory under the patch folder. Project, non-project, and UNIX changes are identified for use later in the workflow to identify what type of processing must be performed in the object type execution on the destination.
- 5. Project Files Exist**—The projects folder of the patch is checked to see whether it has any project files. If so, proceed to the Approve Demo Lock step; if not, go to the Apply Patch to DEV step.
- 6. Approve Demo Lock**—To keep anyone from overwriting data at the source or destination during the project copy process, all the objects in the project are locked in the Demo database before the project is copied. In this step, the user approves the lock.
- 7. Lock and Copy Project to Demo**—The patch project files are copied to the Demo database. Objects in the patch project(s) that already exist in the Demo database are locked. If any objects have been locked by another process, an error is generated. Log files are created in the Report ID folder.
- 8. Run Comparison Reports**—In this step, the user runs comparison reports between the Demo and destination instances to evaluate impacts on customized objects and take any actions required before final migration.
- 9. Apply Patch to DEV**—The PeopleSoft 8.44 Patch SubWorkflow is called to deploy the patch to the DEV instance.

PeopleSoft 8.44 Patch SubWorkflow steps:

- 9.1. Project Files Exist**—This step determines whether the patch includes any project files. If so, proceed to the Lock Objects step. If not, go to the Apply Non-Project to Destination step.
- 9.2. Lock Objects**—This step locks all the objects that are in patch projects in both the source and destination databases before migration. However, if any of these objects were previously locked by another process, an error is generated and the subworkflow goes to the Unlock Objects (Failure) step.
- 9.3. Copy Project to Destination**—Projects are copied from the Demo database to the destination database, using the PeopleSoft command line interface. If there is a failure, the subworkflow goes to the Unlock Objects (Failure) step.
- 9.4. Unlock Objects (Success)**—If the copy is successful, the objects on the source and destination are unlocked.

**9.5. Build Project**—The user builds the project at the destination.

**9.6. Apply Non-Project to Destination**—At the destination, this step deploys (migrates, or migrates and executes) all non-project changes for supported file types.

**9.7. Additional Processing**—In this step, the user performs any additional processing for file types that are not supported for automated deployment.

**9.8. Return**—This step returns to the parent workflow, where it goes to the Apply Patch to TEST step.

**9.9. Unlock Objects (Failure)**—After a failure in the Copy Project to Destination step, this step unlocks the project objects in the source and destination.

**9.10. Remove Temp Files (Failure)**—Any failure or cancellation during the subworkflow leads to this step, where all temporary and log files created in the deployment process are deleted, and the subworkflow goes to the Return step.

**10. Apply Patch to TEST**—The PeopleSoft 8.44 Patch SubWorkflow is called to deploy the patch to the TEST instance. See the details of this subworkflow in the preceding Apply Patch to DEV step.

**11. Apply Patch to PROD**—The PeopleSoft 8.44 Patch SubWorkflow is called to deploy the patch to the PROD instance. See the details of this subworkflow in the preceding Apply Patch to DEV step.

**12. Remove Temp Files (Success)**—This step removes temporary files generated during the migration and deployment of patches.

**13. Unlock Demo**—This step unlocks the Demo database.

**14. Close (Success)**—Upon successful completion, the package line is closed.

**15. Remove Temp Files (Failure)**—If any step in the patch process fails or is cancelled, this step removes temporary files generated during the migration and deployment of patches.

**16. Unlock Demo**—This step unlocks the Demo database.

**17. Close (Failure)**—At the conclusion of any failed process, close the package line.

## Requirements

To use the patch applicator, the requirements described in the following sections must be met:

### *Software Requirements*

The following software requirements must be met:

- PeopleTools version 8.44
- PeopleSoft-endorsed COBOL compiler
- Mercury IT Governance Center release 6.0

### *File Structure Requirements*

The file storage structure must be identical on all PeopleSoft instances except Demo. See [System Requirements on page 26](#).

### *Environment Requirements*

The following environments must be specified:

- PeopleSoft Patch Stage—the environment where all the patch files will be stored
- Demo instance of PeopleSoft—the environment where all the project changes will be applied
- Development instance of PeopleSoft
- Test instance of PeopleSoft
- Production instance of PeopleSoft

### *Access Requirements*

To log on to the environments, users need access to:

- `sqlplus.exe`
- `pside.exe`
- `psdmt.exe`

## ***Workflow Setup Requirements***

You must set up the PeopleSoft 8.44 Patch Applicator workflow steps to include the source and destination environments listed in [Table 5-7](#).

*Table 5-7. PeopleSoft 8.44 Patch Applicator workflow setup*

Workflow Step Name	Source Environment	Destination Environment
Create Patch Folder	PeopleSoft Patch Stage	PeopleSoft Patch Stage
List Supported Patch Files	PeopleSoft Patch Stage	
Project Files Exist		
Lock and Copy Project to Demo	PeopleSoft Patch Stage	PeopleSoft Demo
Apply Patch to DEV	PeopleSoft Patch Stage	PeopleSoft DEV environment or environment group
Apply Patch to TEST	PeopleSoft Patch Stage	PeopleSoft TEST environment or environment group
Apply Patch to PROD	PeopleSoft Patch Stage	PeopleSoft PROD environment or environment group
Remove Temp Files (Success)	PeopleSoft Patch Stage	
Remove Temp Files (Failure)	PeopleSoft Patch Stage	PeopleSoft Demo
Unlock Demo		

Set up the PeopleSoft 8.44 Patch SubWorkflow steps to include the source and destination environments listed in [Table 5-8](#).

*Table 5-8. PeopleSoft 8.44 Patch SubWorkflow setup*

Subworkflow Step Name	Source Environment	Destination Environment
Project Files Exist	* See the note below.	** For all of the destination environments, see the note below.
Lock Objects	PeopleSoft Demo	
Copy Projects to Dest	PeopleSoft Demo	
Unlock Objects (Success)	PeopleSoft Demo	
Apply Non-Project to Destination	* See the note below.	
Unlock Objects (Failure)	PeopleSoft Demo	
Remove Temp Files (Failure)	* See the note below.	

\* This source environment makes use of the environment specified in the parent workflow step. For example, if a subworkflow step is called by the Apply Patch to DEV step, the subworkflow uses the same source environment as the Apply Patch to DEV step, which is the PeopleSoft Patch Stage.

\*\* This destination environment makes use of the environment specified in the parent workflow step. For example, if any of the subworkflow steps are called by the Apply Patch to DEV step, the subworkflow uses the same destination environment as the Apply Patch to DEV step, which is the PeopleSoft DEV instance.

## Workflow Notification

You must select which users are notified for the following workflow steps:

- PeopleSoft 8.44 Patch Applicator workflow steps
  - Distribute Patch Files
  - Run Patch Install Setup
  - Approve File Distribution
  - Approve Demo Lock
  - Run Comparison Reports

- PeopleSoft 8.44 Patch SubWorkflow steps
  - Build Project
  - Additional Processing

## Object Type

*Figure 5-9* shows sample data for the PeopleSoft 8.44 Patch object type when adding a package line. *Table 5-9* contains field definitions.

The screenshot shows a dialog box titled "Add Line" with a close button in the top right corner. The dialog is divided into two main sections: "Object Type Information" and "Parameters".

**Object Type Information:**

- Object Type:** PeopleSoft 8.44 Patch Object Type
- Sequence:** 1
- Application Code:** None

**Parameters:** (User Data)

- PeopleSoft Product:** (Dropdown menu)
- Product Category:** (Dropdown menu)
- Product Line:** (Text field)
- Product Release:** (Text field)
- Report ID:** (Text field)
- Patch File(s):** (Text field with a file selection icon)
- Log File Name:** (Text field)
- Delete Source Files:** Radio buttons for Yes and No (No is selected)

At the bottom of the dialog, there are four buttons: Clear, OK, Add, and Cancel. A status bar at the very bottom reads: "PeopleSoft 8.44 Patch Object Type' parameters loaded."

*Figure 5-9. PeopleSoft 8.44 Patch object type sample data*

Table 5-9. PeopleSoft 8.44 Patch object type field definitions

Field Name	Definition
PeopleSoft Product	PeopleSoft Product impacted by the patch.
Product Category	Product category impacted by the patch.
Product Line	Product Line impacted by the patch.
Product Release	Product release impacted by the patch.
Report ID	Report ID identifying the patch.
Patch File(s)	Files included in the patch.
Log File Name	Name to be used by patch log files.
Delete Source Files	Whether or not to delete patch files from the source.
File Location (Hidden by default)	File location: <b>Client</b> or <b>Server</b> .
Log Exist (Hidden by default)	Flag indicating whether log files exist.
Project(s) (Hidden by default)	Name(s) of project(s), if any, contained in the applied patch.
NonProject Files Exist (Hidden by default)	Flag indicating whether Data, DMS, SQL Script, nVision Layout, or Crystal Reports files exist in the patch.
Cobol or SQR Files Exist (Hidden by default)	Flag indicating whether COBOL or SQR files exist in the patch.
PatchStage (Hidden by default)	Identifies which environment is the patch stage.
Project Files Exist (Hidden by default)	Flag indicating whether there are projects to be copied as part of the patch.
Sub Path (Hidden by default)	Sub Path for patch files. Defaults to "." for current directory.
SQL Files Exist (Hidden by default and new in release 6.0.0)	Flag indicating whether SQL Script files exist in the patch.
NVISION Files Exist (Hidden by default and new in release 6.0.0)	Flag indicating whether nVision Layout files exist in the patch.

Table 5-9. PeopleSoft 8.44 Patch object type field definitions [continued]

Field Name	Definition
Crystal Reports Exist (Hidden by default and new in release 6.0.0)	Flag indicating whether Crystal Reports files exist in the patch.
DATA Files Exist (Hidden by default and new in release 6.0.0)	Flag indicating whether Data files exist in the patch.
DMS Files Exist (Hidden by default and new in release 6.0.0)	Flag indicating whether DMS files exist in the patch.
SQR Files Exist (Hidden by default and new in release 6.0.0)	Flag indicating whether SQR files exist in the patch.
Cobol Files Exist (Hidden by default and new in release 6.0.0)	Flag indicating whether Cobol files exist in the patch.

## Applying Patches

To apply a patch to PeopleSoft Version 8.44:

1. Download the patch from the PeopleSoft Customer Connection Web site to the Mercury IT Governance Center environment defined as the Patch Stage environment.
2. In the Mercury IT Governance Workbench, generate a new package.
3. Select the PeopleSoft 8.44 Patch Applicator workflow.
4. Click **New Line** to generate a new package line. This opens the Add Line window.
5. Select the PeopleSoft 8.44 Patch object type. This dynamically updates the screen.
6. Choose the PeopleSoft application to which you want to apply the patch.
7. Choose the PeopleSoft Product from the drop-down list.
8. Choose the Product Category from the drop-down list.

9. Enter the Product Line and Product Release information.
10. Enter the Report ID of the patch.
11. In the Patch File(s) field, enter the name of the PeopleSoft patch executable you downloaded in [step 1](#).
12. Enter the Log File Name for the patch. The default is the same as the Report ID.
13. Indicate whether or not you want the patch files deleted at the source.
14. Click **OK** to accept the package line fields and close the window.
15. Click **Submit** to submit the package line. The patch is processed through the PeopleSoft 8.44 Patch Applicator workflow, which notifies users of required decisions, actions, and approvals. Users will be asked to:
  - a. Submit the package line.
  - b. Execute the file distribution step.
  - c. Verify file type distribution.
  - d. Approve any lock(s) for the Demo database, if necessary.
  - e. Run comparison reports.
  - f. Respond to the prompt to lock project objects.
  - g. Build project(s) for the destination instance.
  - h. Apply any additional processing needed that is not performed by the patch applicator.
16. Repeat [step 15](#) for all other destination instances.

# Chapter 6 Version Control

---

## In This Chapter:

- *Overview of Version Control*
- *Requirements*
- *Viewing Version Control Repository Contents*
- *Version Control Menu Items*
- *Creating a Baseline for Online Objects*
- *Checking In Objects*
  - *Checking In Objects on the Project Level*
  - *Checking In Objects on the Project Object Level*
  - *Checking In Flat Files*
- *Checking Out Objects*
  - *Checking Out Objects on the Project Level*
  - *Checking Out Objects on the Project Object Level*
  - *Checking Out Flat Files*
- *Breaking Locks on Objects*
  - *Breaking Locks on the Project Level*
  - *Breaking Locks on the Project Object Level*
  - *Breaking Locks on Flat Files*
- *Configuring Users*
- *Configuring Environments*
  - *Mercury IT Governance Server Database Environment*
  - *PeopleSoft VC Environment*
  - *PeopleSoft Development Environment*
- *PeopleSoft Version Control Pre-configured Dashboard Page*
  - *Overview of Pages and Portlets*
  - *Adding Pre-configured Dashboard Pages*

- *PSVC - Object Status Portlet*
  - *PSVC - Locked Objects Summary Portlet*
  - *PSVC - Object Revision Summary Portlet*
  - *PSVC - Object History Portlet*
- 

## **Overview of Version Control**

Mercury Change Management Extension for PeopleSoft Enterprise provides version control functionality that stores PeopleSoft object definitions in a central repository. These object definitions can then be extracted from the repository and restored in any PeopleSoft instance at any given time.

A module showing the contents of the Version Control repository is provided. It contains different portlets that display different sets of information about the status of the objects in the repository. Menu options are provided to allow users to execute check in and check out for both online objects and flat files.

Options for locking the objects are provided to make sure that only the developer who has the objects checked out can modify them. Then other developers cannot overwrite any modifications being made. An option for breaking locks is also available to enable users such as system administrators to override the locks if the need arises.

## Requirements

The following are required for the PeopleSoft version control functionality:

- Version control server with RCS.
- PeopleSoft development instances.
- Mercury IT Governance instance with Mercury Change Management Extension for PeopleSoft Enterprise installed.
- The line `UTL_FILE_DIR=*` should be included in the `init SID.ora` file of the Mercury IT Governance database, so that the `UTL_FILE` built-in package can be used to read and write to the operating system directories through PL/SQL. To specify a specific directory for accessing the files, replace `*` with the full directory path to be used. The Oracle database needs to be restarted after this change is made.

## Viewing Version Control Repository Contents

When you enable users to access PeopleSoft modules, they can add the PeopleSoft Version Control pre-configured Dashboard page. This page includes the following portlets that contain information about the objects in the Version Control repository:

**PeopleSoft Version Control Object Status.** This portlet provides the current status of PeopleSoft objects in the Version Control database. The portlet data can be filtered using fields such as Application, Updated By, Object Type, and Project Name. The portlet is useful in identifying the current revision number of an object, when the current revision was created, and by whom and whether the object is locked by a user.

**Locked Objects Summary.** This portlet allows the developer to identify objects that are locked in the Version Control repository, along with the user name of the developer who locked the object and the revision of the object. This portlet can be further filtered by fields such as Application and Locked by.

**Object Summary Revision.** This portlet provides the developer with a summary of the number of revisions available for the PeopleSoft objects. The portlet is filtered by fields such as Application and Object Type.

**PeopleSoft Version Control Object History.** This portlet provides developers with a history of modifications made to a particular object. The portlet can be filtered by fields such as Application, Updated By, Object Type, and Object Name.

These portlets can be further configured to filter data relevant to individual needs.

Figure 6-1 shows an example of a PeopleSoft Version Control preconfigured Dashboard page.

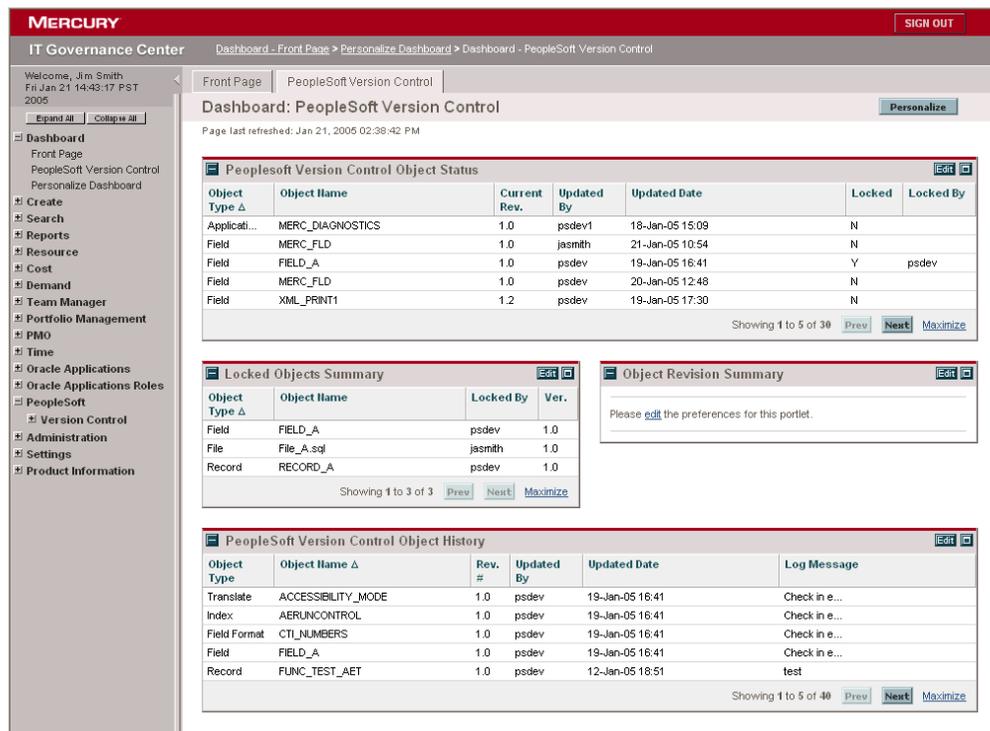


Figure 6-1. PeopleSoft Version Control pre-configured Dashboard page

## Version Control Menu Items

For users who are members of the PS - Developer or the PS - Administrator security group, the standard interface menu bar includes a **PeopleSoft** menu with a **Version Control** submenu that provides quick access to certain menu items (Version Control functions) listed in *Table 6-1*. These functions are described in the following sections.

A different report is invoked for each menu item. For these functions to work, the associated reference report types in the Workbench must be copied and renamed exactly as is but without “(REFERENCE)”. Users are actually submitting reports when they use these convenient **Version Control** menu items.

*Table 6-1. Menu items in PeopleSoft menu, Version Control submenu*

Menu Item	Security Groups That Have Menu Item Available	Function
Baseline	PS - Administrator	Direct access to Baseline functionality for online objects and files.
Break Lock	PS - Administrator PS - Developer	Direct access to Break Lock functionality for online objects and files.
Check In	PS - Developer	Direct access to Check In functionality for online objects and files.
Check Out	PS - Developer	Direct access to Check Out functionality for online objects and files.

The Baseline option appears in this menu for the first time in release 6.0.0.

## Creating a Baseline for Online Objects

To create a baseline of PeopleSoft online objects, you use the **PeopleSoft** menu to submit a report based on the PeopleSoft Version Control - Baseline report type (see *PeopleSoft Version Control - Baseline Report* on page 119). Baselining can be a long, server-intensive process, because it creates an initial check-in for all online objects. It is executed once against an empty Version Control repository.

To create a baseline for customized objects or all objects (see the example in *Figure 6-2*):

1. In the menu bar, click **PeopleSoft > Version Control > Baseline**.



### Note

If you receive an error message that the report type PeopleSoft Version Control - Baseline is not found, do the following: In the Workbench, click **Configuration** and click the **Report Types** icon. Then copy the (REFERENCE) PeopleSoft Version Control - Baseline report type and rename it PeopleSoft Version Control - Baseline. The copied report type must have this exact name for the **PeopleSoft > Version Control > Baseline** standard interface menu item to work.

2. Select the Development Environment where the online objects are located. Only Development environments are listed in the auto-complete list.
3. Select the Version Control Application where the objects are to be stored. The auto-complete list includes all the vaults defined on the Version Control server.
4. In the Baseline field, select **Only Custom Objects** or **All Objects** from the drop-down list.



### Warning

Mercury strongly recommends that you use the **Only Custom Objects** option, because this significantly reduces the number of objects being checked into Version Control. The “vanilla” (non-customized) instance can always be recreated using the PeopleSoft installation software, and the customizations can be applied to it from the Version Control repository. The **All Objects** option checks in both PeopleSoft delivered objects and customized objects, which increases the time required for this operation significantly.

5. In the Log Message field, enter a log message for the baseline.
6. Enter a Description for the objects being checked in.

The screenshot shows a web-based form titled "Submit Report: PeopleSoft Version Control - Baseline" within a "MERCURY" application window. The form is organized into several sections:

- Report Parameters:** This section contains several input fields:
  - Development Environment:** A text box containing "PeopleSoft PT844DEV".
  - Version Control Application:** A text box containing "vault".
  - Baseline:** A dropdown menu currently set to "Only Custom Objects".
  - Log Message:** A text box containing "Baseline Version".
  - Description:** An empty text box.
- Scheduling:** A section header with a plus icon, currently collapsed.
- Advanced Notifications:** A section header with a plus icon, currently collapsed.

At the top right of the form area, there are "Submit" and "Cancel" buttons. At the bottom right, there are also "Submit" and "Cancel" buttons. A "Restore Default" button is located to the right of the "Report Parameters" section header. The window title bar at the top shows "MERCURY" and a "Close Window X" button.

Figure 6-2. Baselining PeopleSoft online objects

## Checking In Objects

Different versions of PeopleSoft objects can be maintained using the provided Check In functionality. This allows users to store a copy of their revisions into a Version Control repository. Objects are copied from a specified Development environment into a Version Control repository. The copies are assigned revision numbers and any locks on the objects are released.

To check in objects, you use the **PeopleSoft** menu to submit a report based on the PeopleSoft Version Control - Check In report type (see [PeopleSoft Version Control - Check In Report on page 124](#)). Log on as a developer and check in any of the following as described in the following sections:

- Objects on the project level
- Objects on the project object level
- Flat files

### Checking In Objects on the Project Level

To check in online objects on the project level (see the example in [Figure 6-3](#)):

1. In the menu bar, click **PeopleSoft > Version Control > Check In**.



If you receive an error message that the report type PeopleSoft Version Control - Check In is not found, do the following: In the Workbench, click **Configuration** and click the **Report Types** icon. Then copy the (REFERENCE) PeopleSoft Version Control - Check In report type and rename it PeopleSoft Version Control - Check In. The copied report type must have this exact name for the **PeopleSoft > Version Control > Check In** standard interface menu item to work.

2. Select the Development Environment where the projects are located. Only Development environments are listed in the auto-complete list.
3. Select the Version Control Application where the projects are to be stored. The auto-complete list includes all the vaults defined on the Version Control server.
4. To check if the project and its objects are locked by the current user, set Check If Current User Has Lock On Objects to **Yes**. Then you know that the user checked out the project and its objects and has exclusive rights to make changes to it before making the modifications. To override this verification, set Check If Current User Has Lock On Objects to **No**.

5. Set Online Object to **Yes**.
6. Select the Project Name. The auto-complete list includes all projects available in the specified Development environment.
7. Set the Check In field to **Entire Project**.
8. In the Log Message field, describe the changes made.
9. In the Reference ID field, select a package number to associate with this change.
10. Enter a Description for the objects being checked in.
11. Click **Submit**. This executes the Check In functionality.

A report is displayed showing the status of the objects checked into Version Control.

The screenshot shows a web form titled "Submit Report: PeopleSoft Version Control - Check In" within a "MERCURY" window. The form is organized into several sections:

- Report Parameters:** Contains a "Restore Default" button.
- Development Environment:** Text input field with "PeopleSoft PT844DEV".
- Version Control Application:** Text input field with "vault".
- Check If Current User Has Lock On Objects?:** Radio buttons for "Yes" (selected) and "No".
- Online Object?:** Radio buttons for "Yes" (selected) and "No".
- Project Name:** Text input field with "MERC\_PROJ".
- Check In:** Dropdown menu with "Entire Project" selected.
- Project Object Type:** Text input field.
- Project Object Name(s):** Text input field.
- Development Sub Path:** Text input field.
- Filename:** Text input field.
- Log Message:** Text input field with "Adding new project".
- Reference ID:** Text input field with "30006".
- Description:** Text input field.

At the bottom of the form, there are sections for "Scheduling" and "Advanced Notifications", each with a plus icon. "Submit" and "Cancel" buttons are located at the bottom right of the form area.

Figure 6-3. Checking in PeopleSoft objects on the project level

## Checking In Objects on the Project Object Level

To check in online objects on the project object level (see the example in [Figure 6-4](#)):

1. In the menu bar, click **PeopleSoft > Version Control > Check In**.



Note

If you receive an error message that the report type PeopleSoft Version Control - Check In is not found, do the following: In the Workbench, click **Configuration** and click the **Report Types** icon. Then copy the (REFERENCE) PeopleSoft Version Control - Check In report type and rename it PeopleSoft Version Control - Check In. The copied report type must have this exact name for the **PeopleSoft > Version Control > Check In** standard interface menu item to work.

2. Select the Development Environment where the project objects are located. Only Development environments are listed in the auto-complete list.
3. Select the Version Control Application where the project objects are to be stored. The auto-complete list includes all the vaults defined on the Version Control server.
4. To check if the project objects are locked by the current user, set Check If Current User Has Lock On Objects to **Yes**. Then you know that the user checked out the project objects and has exclusive rights to make changes to them before making the modifications. To override this verification, set Check If Current User Has Lock On Objects to **No**.
5. Set Online Object to **Yes**.
6. Select the Project Name. The auto-complete list includes all projects available in the specified Development environment.
7. Set the Check In field to **Individual Project Objects Only**.
8. Select the Project Object Type. The auto-complete list includes all object types available.
9. Select the Project Object Name(s). The auto-complete list includes all the project object names available for the specified project and object type in the specified Version Control Application.
10. In the Log Message field, describe the changes made.

11. In the Reference ID field, select a package number to associate with this change.
12. Enter a Description for the objects being checked in.
13. Click **Submit**. This executes the Check In functionality.

A report is displayed showing the status of the objects checked into Version Control.

The screenshot shows a web-based form titled "Submit Report: PeopleSoft Version Control - Check In". The form is organized into several sections:

- Report Parameters:** This section contains the following fields:
  - Development Environment:** Text input field with value "PeopleSoft PT844DEV".
  - Version Control Application:** Text input field with value "vault".
  - Check if Current User Has Lock On Objects?:** Radio button group with "Yes" selected.
  - Online Object?:** Radio button group with "Yes" selected.
  - Project Name:** Text input field with value "MERC\_PROJ".
  - Check In:** Dropdown menu with "Individual Project Objects Only" selected.
  - Project Object Type:** Text input field with value "Field".
  - Project Object Name(s):** Text input field with value "MERC\_FLD".
  - Development Sub Path:** Text input field (empty).
  - Filename:** Text input field (empty).
  - Log Message:** Text input field with value "Replaced MERC\_FLD field".
  - Reference ID:** Text input field with value "30005".
  - Description:** Text input field (empty).
- Scheduling:** Section with a plus icon and a text input field (empty).
- Advanced Notifications:** Section with a plus icon and a text input field (empty).

Buttons for "Submit" and "Cancel" are located at the top right and bottom right of the form area. A "Restore Default" button is also present in the Report Parameters section. The window title bar includes "MERCURY" and "Close Window X".

Figure 6-4. Checking in PeopleSoft objects on the project object level

## Checking In Flat Files

To check in flat files such as SQR and COBOL (see the example in [Figure 6-5](#)):

1. In the menu bar, click **PeopleSoft > Version Control > Check In**.



Note

If you receive an error message that the report type PeopleSoft Version Control - Check In is not found, do the following: In the Workbench, click **Configuration** and click the **Report Types** icon. Then copy the (REFERENCE) PeopleSoft Version Control - Check In report type and rename it PeopleSoft Version Control - Check In. The copied report type must have this exact name for the **PeopleSoft > Version Control > Check In** standard interface menu item to work.

2. Select the Development Environment where the flat files are located. Only Development environments are listed in the auto-complete list.
3. Select the Version Control Application where the files are to be stored. The auto-complete list includes all the vaults defined on the Version Control server.
4. To check if the files are locked by the current user, set Check If Current User Has Lock On Objects to **Yes**. Then you know that the user checked out the files and has exclusive rights to make changes to them before making the modifications. To override this verification, set Check If Current User Has Lock On Objects to **No**.
5. Set Online Object to **No**.
6. Set the Development Sub Path field to the file location.
7. In the Filename field, enter the name of the file to be checked in.
8. In the Log Message field, describe the changes made.
9. In the Reference ID field, select a package number to associate with this change.
10. Enter a Description for the file being checked in.
11. Click **Submit**. This executes the Check In functionality.

A report is displayed showing the status of the files checked into Version Control.

The screenshot shows a web-based form titled "Submit Report: PeopleSoft Version Control - Check In" within a window labeled "MERCURY". The form is organized into several sections:

- Report Parameters:** A section with a "Restore Default" button. It contains the following fields:
  - Development Environment:** Text input with value "PeopleSoft PT844DEV".
  - Version Control Application:** Text input with value "vault".
  - Check if Current User Has Lock On Objects?:** Radio buttons for "Yes" (selected) and "No".
  - Online Object?:** Radio buttons for "Yes" and "No" (selected).
  - Project Name:** Text input field.
  - Check In:** Dropdown menu with "Entire Project" selected.
  - Project Object Type:** Text input field.
  - Project Object Name(s):** Text input field.
  - Development Sub Path:** Text input with value "jsq/".
  - Filename:** Text input with value "datetime.sq".
  - Log Message:** Text input with value "Added new format".
  - Reference ID:** Text input with value "30006".
  - Description:** Text input field.
- Scheduling:** A section with a plus icon and the text "Scheduling".
- Advanced Notifications:** A section with a plus icon and the text "Advanced Notifications".

At the top right and bottom right of the form area, there are "Submit" and "Cancel" buttons. A "Close Window X" button is visible in the top right corner of the window frame.

Figure 6-5. Checking in flat files

## Checking Out Objects

Users can retrieve a copy of their object revisions from a Version Control repository and establish locks on those objects so that only they can make changes to those objects.

To check out objects, you use the **PeopleSoft** menu to submit a report based on the PeopleSoft Version Control - Check Out report type (see [PeopleSoft Version Control - Check Out Report on page 127](#)). Log on as a developer and check out any of the following as described in the following sections:

- Objects on the project level
- Objects on the project object level
- Flat files

### Checking Out Objects on the Project Level

To check out online objects on the project level (see the example in [Figure 6-6](#)):

1. In the menu bar, click **PeopleSoft > Version Control > Check Out**.



Note

If you receive an error message that the report type PeopleSoft Version Control - Check Out is not found, do the following: In the Workbench, click **Configuration** and click the **Report Types** icon. Then copy the (REFERENCE) PeopleSoft Version Control - Check Out report type and rename it PeopleSoft Version Control - Check Out. The copied report type must have this exact name for the **PeopleSoft > Version Control > Check Out** standard interface menu item to work.

2. Select the Version Control Application containing the checked-in projects. The auto-complete list includes all the vaults defined on the Version Control server.
3. Select the Development Environment to which the project needs to be checked out. Only Development environments are listed in the auto-complete list.

4. The locking function provides a developer exclusive access to a particular version of a project so that no other developer can make changes to that version of the project. Set Lock Objects to **Yes** to enable the lock or set it to **No** to check out objects without locking. The project will be locked in the Version Control Application, in the PeopleSoft database, and in the Mercury IT Governance Version Control tables.
5. Set Online Object to **Yes**.
6. Set the Check Out field to **Entire Project**.
7. Select the Project Name. The auto-complete list includes all projects available in the specified Version Control Application.
8. Select the Project Revision Number. The auto-complete list includes all revision numbers available for the specified project in the specified Version Control Application.
9. Set Check Out Current Rev of All Objects in Project to **Yes** if you want to check out the latest revision of all of this project's objects, wherever the latest revision is found in the entire repository.

Set Check Out Current Rev of All Objects in Project to **No** if you want to check out all of this project's objects at the revision the objects had when the project was at the revision specified in the Project Revision Number field.

10. Click **Submit**. This executes the Check Out functionality.

A report is displayed showing the status of the objects checked out of Version Control.

**MERCURY** Close Window X

Submit Report: PeopleSoft Version Control - Check Out

---

\*Version Control Application:

\*Development Environment:

\*Lock Objects?  Yes  No

\*Online Object?  Yes  No

\*Check Out:

Project Name:

Project Revision Number:

\*Check Out Current Rev of All Objects in Project?  Yes  No

Project Object Type:

Project Object Name(s):

Filename(s):

Development Sub Path:

Over write File if Exists?  Yes  No

---

Close Window X

Figure 6-6. Checking out PeopleSoft objects on the project level

## Checking Out Objects on the Project Object Level

To check out online objects on the project object level (see the example in [Figure 6-7](#)):

1. In the menu bar, click **PeopleSoft > Version Control > Check Out**.



### Note

If you receive an error message that the report type PeopleSoft Version Control - Check Out is not found, do the following: In the Workbench, click **Configuration** and click the **Report Types** icon. Then copy the (REFERENCE) PeopleSoft Version Control - Check Out report type and rename it PeopleSoft Version Control - Check Out. The copied report type must have this exact name for the **PeopleSoft > Version Control > Check Out** standard interface menu item to work.

2. Select the Version Control Application containing the checked-in project objects. The auto-complete list includes all the vaults defined on the Version Control server.
3. Select the Development Environment to which the project objects need to be checked out. Only Development environments are listed in the auto-complete list.
4. The locking function provides a developer exclusive access to a particular version of a project so that no other developer can make changes to that version of the project. Set Lock Objects to **Yes** to enable the lock or set it to **No** to check out objects without locking. The project will be locked in the Version Control Application, in the PeopleSoft database, and in the Mercury IT Governance Version Control tables.
5. Set Online Object to **Yes**.
6. Set the Check Out field to **Individual Project Objects Only**.
7. To limit the list of object types and names to a specific project, select the Project Name. The auto-complete list includes all projects available in the specified Version Control Application.
8. To limit the list of object types and names to a specific project revision, select the Project Revision Number. The auto-complete list includes all revision numbers available for the specified project in the specified Version Control Application.

9. Set Check Out Current Rev of All Objects in Project to **Yes** if you want to check out the latest revision of all of this project’s objects, wherever the latest revision is found in the entire repository.

Set Check Out Current Rev of All Objects in Project to **No** if you want to check out all of this project’s objects at the revision the objects had when the project was at the revision specified in the Project Revision Number field.

10. Select the Project Object Type. The auto-complete list includes all object types available in the specified Version Control Application.

11. Select the Project Object Name(s). The auto-complete list includes all project object names and their revision numbers available in the specified Version Control Application.

12. Click **Submit**. This executes the Check Out functionality.

A report is displayed showing the status of the objects checked out of Version Control.

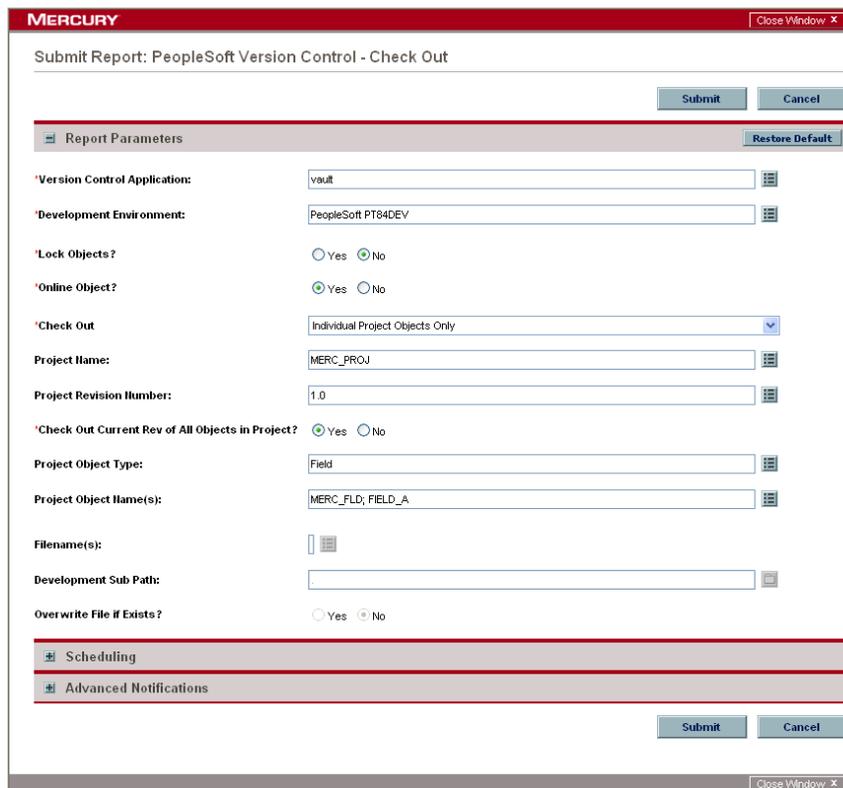


Figure 6-7. Checking out PeopleSoft objects on the project object level

## Checking Out Flat Files

To check out flat files such as SQR or COBOL (see the example in [Figure 6-8](#)):

1. In the menu bar, click **PeopleSoft > Version Control > Check Out**.



### Note

If you receive an error message that the report type PeopleSoft Version Control - Check Out is not found, do the following: In the Workbench, click **Configuration** and click the **Report Types** icon. Then copy the (REFERENCE) PeopleSoft Version Control - Check Out report type and rename it PeopleSoft Version Control - Check Out. The copied report type must have this exact name for the **PeopleSoft > Version Control > Check Out** standard interface menu item to work.

2. Select the Version Control Application containing the checked-in flat files. The auto-complete list includes all the vaults defined on the Version Control server.
3. Select the Development Environment to which the flat files need to be checked out. Only Development environments are included in the auto-complete list.
4. The locking function provides a developer exclusive access to a particular file so that no other developer can make changes to that file. Set Lock Objects to **Yes** to enable the lock. The flat files will be locked in the Version Control Application, in the PeopleSoft database, and in the Mercury IT Governance Version Control tables.
5. Set Online Object to **No**.
6. Select the Filename(s). The auto-complete list includes all files and their revision numbers available in the specified Version Control Application.
7. Set the Development Sub Path field to the file location.
8. If you want the checked-out file to overwrite the existing file in the Development environment, set Overwrite File if Exists to **Yes**.
9. Click **Submit**. This executes the Check Out functionality.

A report is displayed showing the status of the files checked out of Version Control.

**MERCURY** Close Window X

Submit Report: PeopleSoft Version Control - Check Out

Submit Cancel

**Report Parameters** Restore Default

\*Version Control Application: vault

\*Development Environment: PeopleSoft PT84DEV

\*Lock Objects?  Yes  No

\*Online Object?  Yes  No

Check Out: Entire Project

Project Name: |

Project Revision Number: |

Check Out Current Rev of All Objects in Project?  Yes  No

Project Object Type: |

Project Object Name(s): |

\*Filename(s): File\_A.sql

\*Development Sub Path: .jsqf

\*Overwrite File if Exists?  Yes  No

**Scheduling**

**Advanced Notifications**

Submit Cancel

Close Window X

Figure 6-8. Checking out flat files

## Breaking Locks on Objects

During the normal Version Control process, locks obtained by a developer on objects are released when the objects are checked in. To handle exceptional cases, locks can be broken using the Break Lock functionality. This allows users to remove locks on objects they have checked out at any time without the need to check in the objects. This functionality also allows users with system administrator privileges to remove locks obtained by other users.

To break locks on objects, you use the **PeopleSoft** menu to submit a report based on the PeopleSoft Version Control - Break Lock report type (see [PeopleSoft Version Control - Break Lock Report on page 121](#)). Log on as a developer and break the lock for any of the following as described in the following sections:

- Objects on the project level
- Objects on the project object level
- Flat files

## Breaking Locks on the Project Level

To break locks on online objects on the project level (see the example in [Figure 6-9](#)):

1. In the menu bar, click **PeopleSoft > Version Control > Break Lock**.

Note

If you receive an error message that the report type PeopleSoft Version Control - Break Lock is not found, do the following: In the Workbench, click **Configuration** and click the **Report Types** icon. Then copy the (REFERENCE) PeopleSoft Version Control - Break Lock report type and rename it PeopleSoft Version Control - Break Lock. The copied report type must have this exact name for the **PeopleSoft > Version Control > Break Lock** standard interface menu item to work.

2. Select the Version Control Application containing the locked objects. The auto-complete list includes all the vaults defined on the Version Control server.
3. Select the Development Environment where the projects are located. Only Development environments are listed in the auto-complete list.
4. Set Online Object to **Yes**.

5. Set the Break Lock on field to **Entire Project**.
6. Select the Project Name. The auto-complete list includes all projects available in the specified Version Control Application.
7. Select the Project Revision Number. The auto-complete list includes all revision numbers available for the specified project in the specified Version Control Application.
8. Set Break Lock on Current Rev of All Objs in Project to **Yes** if you want to break locks on the latest versions of the objects in the Version Control Application for the project.

Set Break Lock on Current Rev of All Objs in Project to **No** if you want to break locks on the version of the objects associated with the project and the specified project revision.

9. Click **Submit**. This executes the Break Lock functionality.

A report is displayed showing the status of the object locks that were broken.

The screenshot shows a web-based form titled "Submit Report: PeopleSoft Version Control - Break Lock". The form is divided into several sections. At the top right, there is a "Close Window X" button. Below the title, there are "Submit" and "Cancel" buttons. The main section is titled "Report Parameters" and contains the following fields: "Version Control Application" (text box with "vault"), "Development Environment" (text box with "PeopleSoft PT84DEV"), "Online Object?" (radio buttons for "Yes" and "No", with "Yes" selected), "Break Lock on" (dropdown menu with "Entire Project" selected), "Project Name" (text box with "MERC\_PROJ"), "Project Revision Number" (text box with "1.0"), "Break Lock on Current Rev of All Objs in Project?" (radio buttons for "Yes" and "No", with "Yes" selected), "Project Object Type" (text box), "Project Object Name(s)" (text box), and "Filename(s)" (text box). Below the "Report Parameters" section, there are two expandable sections: "Scheduling" and "Advanced Notifications". At the bottom right, there are "Submit" and "Cancel" buttons. At the very bottom, there is another "Close Window X" button.

Figure 6-9. Breaking locks on PeopleSoft objects on the project level

---

## Breaking Locks on the Project Object Level

To break locks on online objects on the project object level (see the example in [Figure 6-10](#)):

1. In the menu bar, click **PeopleSoft > Version Control > Break Lock**.



### Note

If you receive an error message that the report type PeopleSoft Version Control - Break Lock is not found, do the following: In the Workbench, click **Configuration** and click the **Report Types** icon. Then copy the (REFERENCE) PeopleSoft Version Control - Break Lock report type and rename it PeopleSoft Version Control - Break Lock. The copied report type must have this exact name for the **PeopleSoft > Version Control > Break Lock** standard interface menu item to work.

2. Select the Version Control Application containing the locked objects. The auto-complete list includes all the vaults defined on the Version Control server.
3. Select the Development Environment where the project objects are located. Only Development environments are listed in the auto-complete list.
4. Set Online Object to **Yes**.
5. Set the Break Lock on field to **Individual Project Objects Only**.
6. To limit the list of object types and names to a specific project, select the Project Name. The auto-complete list includes all projects available in the specified Version Control Application.
7. To limit the list of object types and names to a specific project revision, select the Project Revision Number. The auto-complete list includes all revision numbers available for the specified project in the specified Version Control Application.
8. Select the Project Object Type. The auto-complete list includes all object types available for any specified project and project revision in the specified Version Control Application.

9. Select the Project Object Name(s). The auto-complete list includes all project object names and their revision numbers available for any specified project and project revision and the specified object type in the specified Version Control Application.

10. Click **Submit**. This executes the Break Lock functionality.

A report is displayed showing the status of the object locks that were broken.

The screenshot shows a web-based interface for Mercury software. The title bar reads 'MERCURY' and 'Close Window'. The main window title is 'Submit Report: PeopleSoft Version Control - Break Lock'. Below the title bar are 'Submit' and 'Cancel' buttons. A 'Report Parameters' section is visible, containing the following fields and controls:

- 'Version Control Application': Text input field with 'vault' entered.
- 'Development Environment': Text input field with 'PeopleSoft PT844DEV' entered.
- 'Online Object?': Radio buttons for 'Yes' (selected) and 'No'.
- 'Break Lock on': Dropdown menu with 'Individual Project Objects Only' selected.
- 'Project Name': Text input field.
- 'Project Revision Number': Text input field.
- 'Break Lock on Current Rev of All Objs in Project?': Radio buttons for 'Yes' (selected) and 'No'.
- 'Project Object Type': Text input field with 'Field' entered.
- 'Project Object Name(s)': Text input field with 'MERC\_FLD; FIELD\_A' entered.
- 'Filename(s)': Text input field.

Below the 'Report Parameters' section are sections for 'Scheduling' and 'Advanced Notifications', each with a plus icon. At the bottom of the window are 'Submit' and 'Cancel' buttons, and a 'Close Window' button in the title bar.

Figure 6-10. Breaking locks on PeopleSoft objects on the project object level

---

## Breaking Locks on Flat Files

To break locks on flat files such as SQR, COBOL (see the example in [Figure 6-11](#)):

1. In the menu bar, click **PeopleSoft > Version Control > Break Lock**.



### Note

If you receive an error message that the report type PeopleSoft Version Control - Break Lock is not found, do the following: In the Workbench, click **Configuration** and click the **Report Types** icon. Then copy the (REFERENCE) PeopleSoft Version Control - Break Lock report type and rename it PeopleSoft Version Control - Break Lock. The copied report type must have this exact name for the **PeopleSoft > Version Control > Break Lock** standard interface menu item to work.

2. Select the Version Control Application containing the locked objects. The auto-complete list includes all the vaults defined on the Version Control server.
3. Select the Development Environment where the flat files are located. Only Development environments are listed in the auto-complete list.
4. Set Online Object to **No**.
5. Select the Filename(s). The auto-complete list includes all files and their revision numbers available in the specified Version Control Application.
6. Click **Submit**. This executes the Break Lock functionality.

A report is displayed showing the status of the file locks that were broken.

The screenshot shows a web-based interface for submitting a report. The title bar reads "MERCURY" and "Close Window X". The main title is "Submit Report: PeopleSoft Version Control - Break Lock". Below the title are "Submit" and "Cancel" buttons. The "Report Parameters" section is expanded, showing several input fields: "Version Control Application" with the value "vaut", "Development Environment" with "PeopleSoft PT844DEV", "Online Object?" with radio buttons for "Yes" and "No" (selected), "Break Lock on" with a dropdown menu, "Project Name", "Project Revision Number", "Break Lock on Current Rev of All Objs in Project?" with radio buttons for "Yes" and "No" (selected), "Project Object Type", "Project Object Name(s)", and "Filename(s)" with the value "File\_A.sql". Below this section are "Scheduling" and "Advanced Notifications" sections, each with a plus icon. At the bottom right of the form area are "Submit" and "Cancel" buttons. The footer of the window contains "Close Window X".

Figure 6-11. Breaking locks on flat files

## Configuring Users

Use Mercury IT Governance Workbench to configure user information for developers and system administrators who access Version Control utilities. [Table 6-2](#) lists user information that needs to be configured.

*Table 6-2. User information for Version Control*

Field	Value
<b>User Information tab</b>	
Username	Username for developer or system administrator.
<b>Security Groups tab</b>	
Group Name	<b>PS - Developer</b> or <b>PS - Administrator</b>
<b>Extension Data tab, PeopleSoft subtab</b>	
PeopleSoft Operator ID	Operator ID for developer or administrator to interact with the PeopleSoft Application server.
Password	Password for PeopleSoft Operator ID.
Version Control Server Username	Version Control server username for developer or administrator to connect to the Version Control server.
Password	Password for Version Control server username.

### For More Information

For more information about configuring users, see [Configuring PeopleSoft Users](#) on page 51.

## Configuring Environments

The following sections describe how to configure the following environments:

- Mercury IT Governance Server database server environment
- PeopleSoft Version Control (VC) environment
- PeopleSoft Development environment

### For More Information

For more information about configuring environments, see [Configuring Environments on page 44](#).

## Mercury IT Governance Server Database Environment

ITG\_SERVER\_DATABASE is the Mercury IT Governance Server database server where Version Control object information is stored. [Table 6-3](#) lists the environment information that needs to be configured in the Environment window for ITG\_SERVER\_DATABASE.

Table 6-3. Environment window required fields for ITG\_SERVER\_DATABASE

Field Name	Value
Environment Name	<b>ITG_SERVER_DATABASE</b>
Enabled	<b>Yes</b>
<b>Host tab, Server section</b>	
Enable Server	Enabled (select this checkbox)
Name	Name of the server where the database resides.
Type	Type of server: <b>Windows</b> or <b>UNIX</b> .
Username	Username for access to the database.
Password	Password for username.
NT Domain	(Windows only) Domain name.
Base Path	Base path for the database. This path should be accessible by the UTL_FILE_DIR parameter in the <code>initSID.ora</code> file of the database.
Connection Protocol	Connection protocol to use.
Transfer Protocol	Transfer protocol to use.

Table 6-3. Environment window required fields for ITG\_SERVER\_DATABASE [continued]

Field Name	Value
<b>Host tab, Client section</b>	
Enable Client	Disabled (deselect this checkbox)
<b>Host tab, Database section</b>	
Enable Database	Enabled (select this checkbox)
Server Type	<b>Oracle Server</b>
Host Name	Name of server where the database resides.
Connect String	Connect string for the database.
Username	Username for the database.
Password	Password for the database.
Oracle SID	SID for the database.

## PeopleSoft VC Environment

PeopleSoft VC is the Version Control Server where all vaults and repositories are located. [Table 6-4](#) lists the environment information that needs to be configured for PeopleSoft VC.

Table 6-4. Environment window required fields for PeopleSoft VC environment

Field Name	Value
Environment Name	<b>PeopleSoft VC.</b>
Enabled	<b>Yes.</b>
<b>Host tab, Server section</b>	
Enable Server	Enabled (select this checkbox).
Name	Name of server where Version Control directories reside.
Type	Type of server: <b>Windows</b> or <b>UNIX</b> .
Username	Username for access to Version Control server.
Password	Password for username.
NT Domain	Domain name if Windows.
Base Path	Base path for Version Control server.

Table 6-4. Environment window required fields for PeopleSoft VC environment [continued]

Field Name	Value
Connection Protocol	Connection protocol to use.
Transfer Protocol	Transfer protocol to use.
<b>Host tab, Client section</b>	
Enable Client	Disabled (deselect this checkbox).
<b>Host tab, Database section</b>	
Enable Database	Disabled (deselect this checkbox).
<b>Applications tab</b>	
Application Code	Application Code for Version Control vault.
Application Name	Application Name for Version Control vault.
Server Base Path	Base Path for Version Control directory.
<b>Extension Data tab, PeopleSoft subtab</b>	
Set as default Extension	<b>Yes</b>
Enabled	<b>Yes</b>
PeopleSoft Database Type	<b>Oracle</b>
Environment Type	<b>Version Control</b>

## PeopleSoft Development Environment

If PeopleSoft is set up so that the file server and the batch server are located on separate machines, then the environments for the file server and batch server must be defined separately. *Table 6-5* lists the environment information that needs to be configured for the PeopleSoft development instances that will be used.

*Table 6-5. Environment window required fields for PeopleSoft Development environment*

Field Name	Value
Environment Name	Name of PeopleSoft Development environment.
Enabled	<b>Yes</b>
<b>Host tab, Server section</b>	
Enable Server	Enabled (select this checkbox).
Name	Name of development server.
Type	Type of server: <b>Windows</b> or <b>UNIX</b> .
Username	Username for access to development server.
Password	Password for username.
NT Domain	(Windows only) Domain name.
Base Path	Base path for development server.
Connection Protocol	Connection protocol to use.
Transfer Protocol	Transfer protocol to use.
<b>Host tab, Client section</b>	
Enable Client	Enabled (select this checkbox).
<b>Host tab, Database section</b>	
Enable Database	Enabled (select this checkbox).
Server Type	Database type: <b>Oracle Server</b> .
Host Name	Name of server where the database resides.
Connect String	Connect string for the database.
Username	Username for the database.
Password	Password for the database.
Oracle SID	(Oracle only) SID for the database.

Table 6-5. Environment window required fields for PeopleSoft Development environment

Field Name	Value
Port Number	Port number.
DB Link	(Oracle only) Database link from Mercury IT Governance instance to the PeopleSoft database.
<b>Extension Data</b> tab, <b>PeopleSoft</b> subtab	
Enabled	<b>Yes.</b>
Set as default extension	<b>Yes.</b>
PeopleSoft Database Type	<b>Oracle Server.</b>
Environment Type	<b>Development.</b>
PeopleSoft Username	Username for PeopleSoft operator.
PeopleSoft Password	Password for PeopleSoft operator.
Bootstrap Username	Username for system administrator.
Bootstrap Password	Password for system administrator.
PS Home	PS_HOME directory for the instance.
Migration Path	Alternative to using the base path for connect and copy commands in cases wherein the path for the files is accessed through a mapped drive instead of through Telnet or SSH.

## PeopleSoft Version Control Pre-configured Dashboard Page

This section discusses pre-configured Mercury IT Governance Dashboard pages (referred to hereafter as pages, but also called modules and previously known as templates) and their portlets, in general and as they relate to the Extension.

### Overview of Pages and Portlets

Users can add pre-configured pages to their Dashboards. Then, each page is accessed by clicking the tab for that page. The added pages consist of a set of portlets related to a specific function, and they are generally pre-configured for common usage. Once added, users can personalize the pages to suit their specific business needs.

The pre-configured pages for the Extension are available from the **Dashboard > Personalize Dashboard** menu item in the Mercury IT Governance standard interface. Users can access only those pages for which they have been granted access by their Mercury IT Governance Center administrator. Adding a page also adds the portlets configured for that page.

The pages shown in this document are samples. The content and configuration of pages may vary depending on the Mercury products you have purchased or how the Extension is used by your business.

## Adding Pre-configured Dashboard Pages

Mercury Change Management Extension for PeopleSoft Enterprise includes one preconfigured Dashboard page called PeopleSoft Version Control, as shown in *Figure 6-12*. It is accessible to users who belong to the PS - Developer or the PS - Administrator security group.

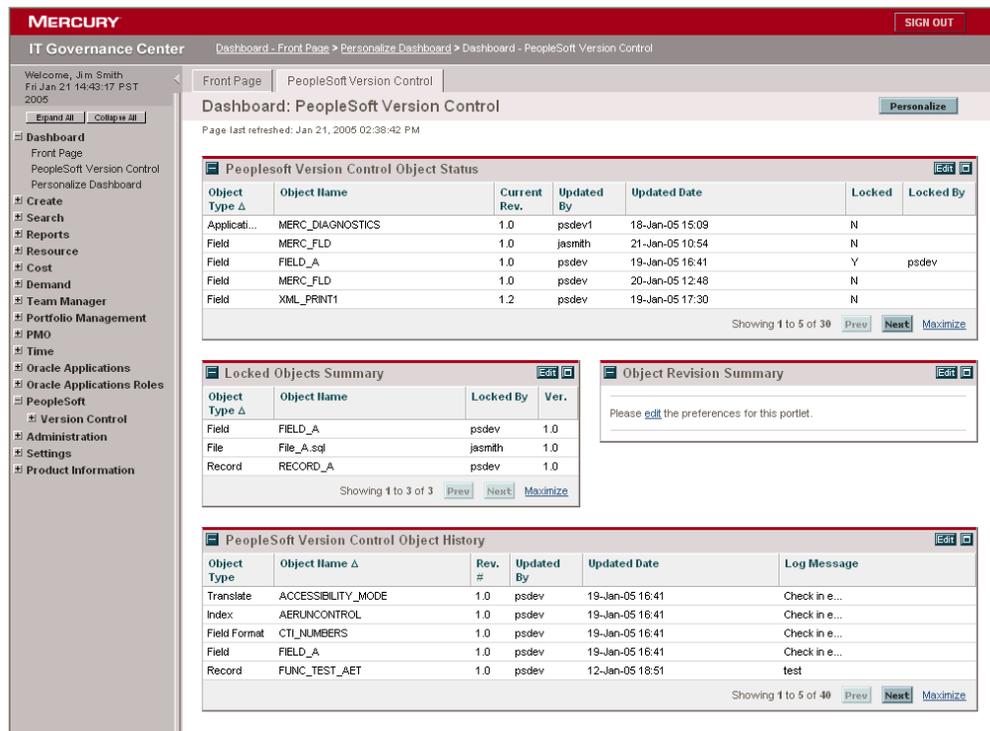
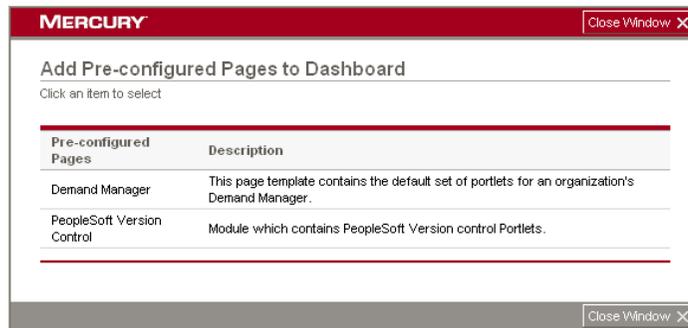


Figure 6-12. PeopleSoft Version Control pre-configured Dashboard page

To add the PeopleSoft Version Control pre-configured Dashboard page:

1. Log on to the Mercury IT Governance Center.
2. In the menu bar on the left, click **Dashboard > Personalize Dashboard**.  
The Personalize Dashboard page appears.
3. Click **Add Pre-configured Pages**.

The Add Pre-configured Pages to Dashboard window opens.



Note

Depending on your security configuration, you might see different page names than the ones in the example.

4. From the list of page names, select **PeopleSoft Version Control**.

The selected page is added to your Dashboard and displayed on a new tab.

5. You can now personalize the portlets as you would any portlets in the Dashboard.

To save the added page to your Dashboard, click **Save**.

To save the added page to your Dashboard and populate the portlets with data as in [Figure 6-12](#), click **Done**.

As described in the following sections, the PeopleSoft Version Control page includes the following portlets by default:

- PSVC - Object Status portlet
- PSVC - Locked Objects Summary portlet
- PSVC - Object Revision Summary portlet
- PSVC - Object History portlet

## PSVC - Object Status Portlet

The PSVC - Object Status portlet provides the current status of PeopleSoft objects in the Version Control database. The portlet is useful in determining:

- The current revision number of the object
- When the current revision was created
- Whether and by whom the object was locked

*Table 6-6* lists the fields displayed in this portlet.

*Table 6-6. PSVC - Object Status portlet displayed fields*

Field Name	Description
Object Type	Category to which the object belongs.
Object Name	Name of the object.
Current Rev.	Current revision number in RCS.
Updated By	Name of user who last updated the object.
Updated Date	Date and time when the object was last updated.
Locked	Whether the object is locked in RCS.
Locked By	Username if the object is locked.

*Table 6-7* lists the filter fields in this portlet.

*Table 6-7. PSVC - Object Status portlet filter fields*

Field Name	Description
Application	Filters the list for objects in the selected application.
Updated By	Filters the list for objects updated by the selected user.
Object Type	Filters the list for objects of selected object type.
Project Name	Filters the list for objects in the selected project.

## PSVC - Locked Objects Summary Portlet

The PSVC - Locked Objects Summary portlet provides a lock summary of PeopleSoft objects in Version Control. The portlet shows users each object that is locked in the Version Control repository along with the name of the user who has locked the object and the revision of the object.

*Table 6-8* lists the fields displayed in this portlet.

*Table 6-8. PSVC - Locked Objects Summary portlet displayed fields*

Field Name	Description
Object Type	Category to which the object belongs.
Object Name	Name of the object.
Locked By	Name of the user who has locked the object.
Ver.	Version number of the locked object.

*Table 6-9* lists the filter fields in this portlet.

*Table 6-9. PSVC - Locked Object Summary portlet filter fields*

Field Name	Description
Application	Filters the list for objects in the selected application.
Locked By	Filters the list for objects locked by the selected user.

## PSVC - Object Revision Summary Portlet

The PSVC - Object Revision Summary portlet provides historical information on PeopleSoft objects in the Version Control database. The portlet is useful in determining:

- The history of revisions made to a particular object
- The reasons for revisions
- The number of revisions made

*Table 6-10* lists the fields displayed in this portlet.

*Table 6-10. PSVC - Object Revision Summary portlet displayed fields*

Field Name	Description
Object Type	Category to which the object belongs.
Object Name	Name of the object.
# of Revs.	How many times the object has been changed.

*Table 6-11* lists the filter fields in this portlet.

*Table 6-11. PSVC - Object Revision Summary portlet filter fields*

Field Name	Description
Application	Filters the list for objects in the selected application.
Object Type	Filters the list for objects of the selected object type.

## PSVC - Object History Portlet

The PSVC - Object History portlet provides historical information on PeopleSoft objects in the Version Control database. The portlet is useful in determining:

- The history of revisions made to a particular object
- The reasons for revisions
- The number of revisions made

*Table 6-12* lists the fields displayed in this portlet.

*Table 6-12. PSVC - Object History portlet displayed fields*

Field Name	Description
Object Type	Category to which the object belongs.
Object Name	Name of the object.
Rev. #	Current revision number in RCS.
Updated By	Name of the user who last updated the object.
Updated Date	Date and time when the object was last updated.
Log Message	Whether or not the object is locked in RCS.

*Table 6-13* lists the filter fields in this portlet.

*Table 6-13. PSVC - Object History portlet filter fields*

Field Name	Description
Application	Filters the list for objects in the selected application.
Updated By	Filters the list for revisions created by the user selected.
Object Type	Filters the list for objects of the selected object type.
Object Name	Filters the list for the selected object.



## Numerics

- 5.0.0 release, upgrading from **29, 30**
- 5.5.0 release, upgrading from **29, 31**
- 6.0.0 release, what's new in **20**
- 7.5 release (PeopleSoft), desupport for **21**
- 8.44 release (PeopleSoft), support for **20**

## B

- baseline, creating for online objects **174**
- breaking locks on objects **189**
- Build Project field **68**

## C

- cblbin directory **58, 63**
- changing passwords **53**
- checking in
  - flat files **180**
  - objects **176**
- checking out
  - flat files **187**
  - objects **182**
- commands
  - ksc\_psft\_check\_for\_locked\_objects **130**
  - ksc\_psft\_list\_objects **130**
  - ksc\_psft\_lock\_objects **130**

- ksc\_psft\_run\_sql\_oracle **130**
- ksc\_psft\_unlock\_object **130**

- configuring
  - environment groups **50**
  - environments **44, 196**
  - PeopleSoft instances **48**
  - PeopleSoft Patch Stage environment **47**
  - PeopleSoft users **51**
  - user information for version control **195**

## D

- Dashboard pages **31, 201, 202**
- data mover tool **58**
- ddlDriver.#####.log **42**
- desupported products **21**
- development environment (PeopleSoft) **199**
- documents
  - prerequisite **17**
  - related **18**
- Download Center (Mercury IT Governance), URL for **38**
- downloading the installation files **38**

## E

- environment groups, configuring **50**
- Environment window **44**

environments, configuring 44, 196  
Execute Build Script field 68  
Extension Data tab, Environment window 44  
Extension overview 19

## F

flat files  
    breaking locks on 193  
    checking in 180  
    checking out 187

## I

installing  
    collecting information for 37  
    downloading the files 38  
    installation summary report 40  
    logs generated during installation 41  
    overview of steps to follow 26  
    post-installation procedures 42  
    preparing for 37  
    procedure for 39  
instances, configuring 48  
ITG\_DEPLOYMENT\_HOME environment variable 37  
ITG\_Home directory 37  
ITG\_Home/Deployment\_Home directory 37

## J

jarxvf.#####.log 42

## K

kDeploy.sh script 39, 40  
kPSFTEnvUpdatePassword.sh script 53  
ksc\_psft\_check\_for\_locked\_objects command 130  
ksc\_psft\_list\_objects command 130  
ksc\_psft\_lock\_objects command 130  
ksc\_psft\_run\_sql\_oracle command 130  
ksc\_psft\_unlock\_objects command 130

## L

license agreement for the Extension 38  
Locked Objects Summary portlet 171  
logs generated during installation  
    ddlDriver.#####.log 42  
    jarxvf.#####.log 42  
    packageDriver.#####.log 42  
    postXMLDriver.#####.log 42  
    preXMLDriver.#####.lo 42

## M

mass password change functionality 53  
Microsoft SQL Server, desupport for 21  
mitg\_extensions.sh script 39  
mitg-600-PeopleSoft.jar file 37, 39

## O

Object Summary Revision portlet 172  
object types  
    PeopleSoft 8.1.x Patch 59, 143  
    PeopleSoft 8.4.x Patch 59, 154  
    PeopleSoft 8.44 Patch 59, 165  
    PeopleSoft COBOL Copybook 60  
    PeopleSoft COBOL DMS 62  
    PeopleSoft COBOL Program 63  
    PeopleSoft Data File 65  
    PeopleSoft Load DMS 66  
    PeopleSoft Patch 134  
    PeopleSoft Project 89  
    PeopleSoft Project (ORACLE) 21, 68, 91, 93  
    PeopleSoft SQC 71  
    PeopleSoft SQL Script 72  
    PeopleSoft SQR 74  
objects  
    breaking locks on 189  
    checking in 176  
    checking out 182  
online objects, creating a baseline for 174  
overview of the Extension 19

**P**

- packageDriver.#####.log 42
- passwords, changing 53
- PeopleSoft 8.1.x Patch Applicator Workflow 136
- PeopleSoft 8.1.x Patch object type 59, 143
- PeopleSoft 8.1.x Patch SubWorkflow 136
- PeopleSoft 8.4.x Patch Applicator Workflow 146
- PeopleSoft 8.4.x Patch object type 59, 154
- PeopleSoft 8.4.x Patch SubWorkflow 146
- PeopleSoft 8.44 Patch Applicator Workflow 157
- PeopleSoft 8.44 Patch object type 59, 165
- PeopleSoft 8.44 Patch SubWorkflow 157
- PeopleSoft Application Designer 107
- PeopleSoft Bug request type 77
- PeopleSoft Bug/Enhancement workflow 77, 79, 85
- PeopleSoft COBOL Copybook object type 60
- PeopleSoft COBOL DMS object type 62
- PeopleSoft COBOL Program object type 63
- PeopleSoft Compare Custom Database Setup Report 100
- PeopleSoft Comparison Report 21
- PeopleSoft Customization Applicator 77
- PeopleSoft Customized Objects Report 104
- PeopleSoft Data File object type 65
- PeopleSoft development environment 199
- PeopleSoft Development workflow 87
- PeopleSoft Development Workflow with Project Rollback & Object Warning 68, 89
- PeopleSoft Enhancement request type 79
- PeopleSoft File Comparison Report 106
- PeopleSoft Load DMS object type 66
- PeopleSoft Object Customization Detail Report 107
- PeopleSoft Object Migrations Report 109
- PeopleSoft Patch Applicator 77
- PeopleSoft Patch Applicator tool 134
- PeopleSoft Patch Applicator Workflow for Version 8.1.x 136
- PeopleSoft Patch Applicator Workflow for Version 8.4.x 146
- PeopleSoft Patch Applicator Workflow for Version 8.44 157
- PeopleSoft Patch Applicator workflows 97, 134
- PeopleSoft Patch object type 134
- PeopleSoft Patch request type 81, 96
- PeopleSoft Patch Request workflow 81, 96
- PeopleSoft Patch Stage environment 47
- PeopleSoft Patch subworkflow 134
- PeopleSoft Patches Applied to an Environment Report 111
- PeopleSoft Project (ORACLE) object type 21, 68, 91, 93
- PeopleSoft Project Migration Impact Analysis Report 90, 114
- PeopleSoft Project object type 89
- PeopleSoft Project Object Warning SubWorkflow 91, 92
- PeopleSoft Project Rollback subworkflow 21, 92
- PeopleSoft Request Detail Report 116
- PeopleSoft SQC object type 71
- PeopleSoft SQL Script object type 72
- PeopleSoft SQR object type 74
- PeopleSoft subtab, Environment window 44
- PeopleSoft VC environment 197
- PeopleSoft Version Control - Baseline Report 119
- PeopleSoft Version Control - Break Lock Report 121
- PeopleSoft Version Control - Check In Report 124
- PeopleSoft Version Control - Check Out Report 127
- PeopleSoft Version Control Object History

portlet **172**  
PeopleSoft Version Control Object Status portlet **171**  
portlets  
    Locked Objects Summary **171**  
    Object Summary Revision **172**  
    PeopleSoft Version Control Object History **172**  
    PeopleSoft Version Control Object Status **171**  
    PSVC - Locked Objects Summary **205**  
    PSVC - Object History **207**  
    PSVC - Object Revision Summary **206**  
    PSVC - Object Status **204**  
postXMLDriver.#####.log **42**  
preconfigured Dashboard pages **31**  
prerequisites  
    documents **17**  
    knowledge and experience **23**  
preXMLDriver.#####.log **42**  
PS\_BOOT\_PASSWORD token **132**  
PS\_BOOT\_USER\_NAME token **132**  
PS\_DATABASE\_TYPE token **132**  
PS\_ENV\_TYPE token **132**  
PS\_HOME token **132**  
PS\_MIGRATION\_PATH token **132**  
PS\_OPERATOR\_ID token **132**  
PS\_PASSWORD token **132**  
PS\_PASSWORD\_ID token **132**  
PS\_PSFT\_VERSION token **132**  
PS\_USER\_NAME token **132**  
PS\_VC\_PASSWORD token **132**  
PS\_VC\_USERNAME token **132**  
psdmt.exe **45, 141, 151, 162**  
Pstools **68**  
pstools.exe **45, 141, 151, 162**  
PSVC - Locked Objects Summary portlet **205**  
PSVC - Object History portlet **207**  
PSVC - Object Revision Summary portlet **206**  
PSVC - Object Status portlet **204**

## R

release 5.0.0, upgrading from **29, 30**  
release 5.5.0, upgrading from **29, 31**  
release 6.0.0, what's new in **20**  
release 7.5 (PeopleSoft), desupport for **21**  
release 8.44 (PeopleSoft), support for **20**  
report types  
    overview **99**  
    PeopleSoft Compare Custom Database Setup **100**  
    PeopleSoft Comparison **21**  
    PeopleSoft Customized Objects **104**  
    PeopleSoft File Comparison **106**  
    PeopleSoft Object Customization Detail **107**  
    PeopleSoft Object Migrations **109**  
    PeopleSoft Patches Applied to an Environment **111**  
    PeopleSoft Project Migration Impact Analysis **90, 114**  
    PeopleSoft Request Detail **116**  
    PeopleSoft Version - Break Lock **121**  
    PeopleSoft Version Control - Baseline **119**  
    PeopleSoft Version Control - Check In **124**  
    PeopleSoft Version Control - Check Out **127**  
repository contents, viewing **171**  
request types  
    PeopleSoft Bug **77**  
    PeopleSoft Enhancement **79**  
    PeopleSoft Patch **81, 96**

## S

scripts  
    kDeploy.sh **39, 40**  
    mitg\_extensions.sh **39**  
SQL Server, desupport for **21**  
sqlplus.exe **45, 141, 151, 162**  
sqr directory flags **74**  
sqr.ini file **74**  
subworkflows

---

PeopleSoft 8.1.x Patch SubWorkflow **136**  
PeopleSoft 8.4.x Patch SubWorkflow **146**  
PeopleSoft 8.44 Patch SubWorkflow **157**  
PeopleSoft Patch **134**  
PeopleSoft Project Object Warning **91, 92**  
PeopleSoft Project Rollback **21, 92**

system requirements **26**

## T

tokens

PS\_BOOT\_PASSWORD **132**  
PS\_BOOT\_USER\_NAME **132**  
PS\_DATABASE\_TYPE **132**  
PS\_ENV\_TYPE **132**  
PS\_HOME **132**  
PS\_MIGRATION\_PATH **132**  
PS\_OPERATOR\_ID **132**  
PS\_PASSWORD **132**  
PS\_PASSWORD\_ID **132**  
PS\_PSFT\_VERSION **132**  
PS\_USER\_NAME **132**  
PS\_VC\_PASSWORD **132**  
PS\_VC\_USERNAME **132**

## U

upgrading

from release 5.0.0 **29**  
from release 5.5.0 **29**  
impacts and guidelines **29**

users, configuring **51, 195**

## V

version control

menu items **173**  
overview **170**  
preconfigured Dashboard page **201**  
requirements **171**  
viewing repository contents **171**

## W

workflows

PeopleSoft Bug/Enhancement **77, 79, 85**  
PeopleSoft Development **87**  
PeopleSoft Development Workflow with  
Project Rollback & Object Warning **68, 89**  
PeopleSoft Patch Applicator **97, 134**  
PeopleSoft Patch Applicator Workflow for  
Version 8.1.x **136**  
PeopleSoft Patch Applicator Workflow for  
Version 8.4.x **146**  
PeopleSoft Patch Applicator Workflow for  
Version 8.44 **157**  
PeopleSoft Patch Request **81, 96**

