HP Service Manager

Software Version: 9.40

For the supported Windows® and Unix® operating systems

Process Designer Migration Guide

Document Release Date: January 2015
Software Release Date: January 2015

Legal Notices

Warranty

The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

Restricted Rights Legend

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notice

Trademark Notices

Adobe® is a trademark of Adobe Systems Incorporated.

Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation.

Oracle and Java are registered trademarks of Oracle and/or its affiliates.

UNIX® is a registered trademark of The Open Group.

Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries.

For a complete list of open source and third party acknowledgements, visit the HP Software Support Online web site and search for the product manual called HP Service Manager Open Source and Third Party License Agreements.

Documentation Updates

The title page of this document contains the following identifying information:

- Software Version number, which indicates the software version.
- Document Release Date, which changes each time the document is updated.
- Software Release Date, which indicates the release date of this version of the software.

To check for recent updates or to verify that you are using the most recent edition of a document, go to: **https://softwaresupport.hp.com**

This site requires that you register for an HP Passport and sign in. To register for an HP Passport ID, go to: **http://h20229.www2.hp.com/passport-registration.html**

Or click the **New users - please register** link on the HP Passport login page.

You will also receive updated or new editions if you subscribe to the appropriate product support service. Contact your HP sales representative for details.

Support

Visit the HP Software Support Online website at: **https://softwaresupport.hp.com**

This website provides contact information and details about the products, services, and support that HP Software offers.

HP Software online support provides customer self-solve capabilities. It provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued support customer, you can benefit by using the support website to:

- Search for knowledge documents of interest
- Submit and track support cases and enhancement requests
- Download software patches
- Manage support contracts
- Look up HP support contacts
- Review information about available services
- Enter into discussions with other software customers
- Research and register for software training

Most of the support areas require that you register as an HP Passport user and sign in. Many also require a support contract. To register for an HP Passport ID, go to:

http://h20229.www2.hp.com/passport-registration.html

To find more information about access levels, go to:

http://h20230.www2.hp.com/new_access_levels.jsp

HP Software Solutions Now accesses the HPSW Solution and Integration Portal website. This site enables you to explore HP Product Solutions to meet your business needs, includes a full list of Integrations between HP Products, as well as a listing of ITIL Processes. The URL for this website is **http://h20230.www2.hp.com/sc/solutions/index.jsp**

Contents

Chapter 1: Migration guide overview 6
Target audience 6
About the Service Manager 9.40 Classic and Codeless modes 6
Upgrade process 8
Tips and checklists for upgrading your customization information 11
Help Desk 12
Process Designer framework 14
Menu 15
Change 16
Service Level Agreements (SLAs) 16
Chapter 2: PD Process Migration 17
Service Desk 17
Incident Management 18
Problem Management 20
Problem migration 20
Known Error migration 22
Chapter 3: How to use the data migration tool 25
How to access the tool 27
How to run a migration script 28
Field mapping and value mapping 31
Batch Update mode 34
Out-of-box data migration settings 36
Interaction data migration 36
Incident data migration 38
Problem data migration 40
Task Planner data migration 44
Known Error data migration 46
Migrating from legacy PD known error records 51

Service Level Management data migration 52
Request data migration 54
Request 54
Migrate Quote to PD Request 55
Migrate Order to PD Request 59
Request Task 61
Migrate Quote Line Item to PD Request Task 62
Migrate Order Line Item to PD Request Task 64
Support Data for Request and Request Task 67
Request and Request Task Related Approval Migration 67
Request and Request Task Related Attachment Migration 69
Request Fulfillment Catalog 70
Migrate Independent Fulfillment Catalog Items 70
Migrate Packages 70
Menu of PD Data Migration 75
Service Catalog connector 80
Reconfigure the “Open New Request” connector 80
Reconfigure the “Open a Change” connector 81
Re-index the knowledge bases 82

Changes made by the Process Designer Enablement script 83

Security changes 83
Changes for PD Help Desk 86
Object changes for PD Help Desk 86
Datadict changes for PD Help Desk 87
Menu changes for PD Help Desk 89
SearchConfig 90
Queue displays 91
Related record links 91
KM related records 92
Format changes 92
Links 93
Knowledge Management knowledge base 95
Inbox 96
Service Level Agreements 100
Service Desk 100

SD links 100
svcCatInterface 101
screlconfig 101
Validity 101
activityaction 102
Category related link 102
ModuleStatus 103
AlertDef 105
contextAction 106
extaccess 106
Changes for PD Change 107
Object changes for PD Change 107
Menu changes for PD Change 108
Datadict changes for PD Change 110
Process changes for PD Change 110
extaccess 110
Template 111
Data cleanup 113
Changes for PD Request 118
Object changes for PD Request 119
Menus changes for PD Request 119
SearchConfig 121
Display Options 121
svcCatInterface 122
Inbox 122
Links 124
extaccess 124
Globalist 125
Scheduler 125
Service Desk Environment 125
Assignment groups 125
Send Documentation Feedback 126

Chapter 1: Migration guide overview

The guide outlines the procedures to migrate your existing tailoring information to Process Designer (PD), as well as the procedures to migrate your legacy data to Process Designer based workflows.

Target audience

This guide is intended for the following Service Manager users:

- Service Manager 7.1x/9.2x/9.3x customers who have upgraded to HP Service Manager 9.40 Classic and want to migrate to HP Service Manager 9.40 Codeless by enabling the Process Designer applications.
- Service Manager 9.3x customers who have enabled any Process Designer modules prior to upgrading to version 9.40. These customers should read the entire *Process Designer Migration Guide* (this document), both for guidance on migrating any modules not previously moved to Process Designer and for the data migration steps required to update their legacy data to the updated 9.40 Process Designer workflows.

About the Service Manager 9.40 Classic and Codeless modes

Service Manager 9.40 can be deployed in two different modes:

- Service Manager Classic: This mode refers to a 9.40 system that is upgraded from an earlier Service Manager release and in which Process Designer is not enabled. In a Classic 9.40 system, the Knowledge Management and Service Level Management applications are implemented on Process Designer. All other modules will continue to use non-PD, traditionally tailored applications.
- Service Manager Codeless: This mode refers to a 9.40 system in which Process Designer is enabled. For users upgrading from an earlier Service Manager release, they need to upgrade to 9.40, enable Process Designer, and then complete the migration tasks required to configure the new Process Designer modules to meet their business requirements.

  For new installations of version 9.40 (not upgrading), The Codeless mode of Service Manager is automatically installed. The following table shows the difference of each module in the two modes.

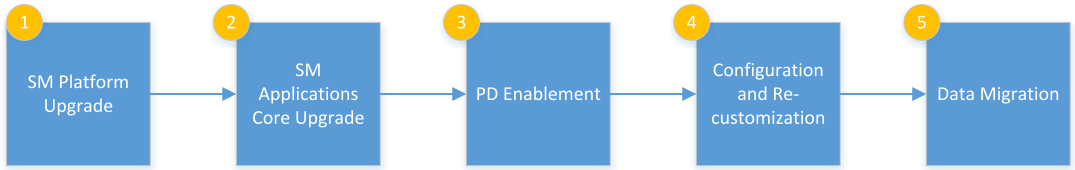
Modules	Classic mode	Codeless mode
Knowledge Management	Process Designer	Process Designer
Service Level Management	Process Designer	Process Designer
Service Desk	Classic Tailoring	Process Designer
Incident Management	Classic Tailoring	Process Designer
Problem Management	Classic Tailoring	Process Designer
Change Management	Classic Tailoring	Process Designer
Request Management	Classic Tailoring	Process Designer

You can use the following rules to determine in which mode you will be running Service Manager:

- Classic mode
  - If you are running a Service Manager 7.x, 9.2x, or 9.3x release, and have not installed an optional Process Designer content pack, you can use the Classic mode of Service Manager.
  - If you upgrade to Service Manager 9.40 and do not enable Process Designer, you will be running the Classic mode of Service Manager.
- Codeless mode (Recommended)
  - If you are running Service Manager 9.3x and have enabled any Process Designer Content Pack (Change, Help Desk, or both), you must enable Process Designer and run Service Manager in its Codeless mode.
  - If you upgrade from any other version, you can optionally enable Process Designer in Service Manager 9.40 to move to HP Service Manager 9.40 Codeless. When Process Designer is enabled, all modules will be moved to the Codeless mode. You do not have an option to enable Process Designer on a module-by-module basis.

Upgrade process

A complete upgrade process from a legacy Service Manager version to Process Designer based modules is outlined in the following figure.



Steps 1 and 2 are normal upgrade processes and explained by existing Service Manager tools and documentation as shown in the following table.

Migration Step	**Tool**	**Document(s)**
1. SM Platform Upgrade	*Binary replacement plus some lightweight server-side reconfiguration*	*Service Manager 9.40 Installation and Upgrade Documentation Center > Interactive Installation Guide*
2. SM Applications Core Upgrade	Service Manager Upgrade tool (for upgrading from an earlier version to 9.40). The Service Manager applications core upgrade usually involves conflict resolution between out-of-box data and customized data.	*Service Manager 9.40 Installation and Upgrade Documentation Center > Applications Upgrade Guide*

Migration Step	Tool	Document(s)
3. Process Designer (PD) Enablement	• Server configuration utility (for new installations only) **Note:** Process Designer is automatically enabled if you load the Service Manager 9.40 applications data by running the server configuration utility. • The "enablepd" command (for systems upgraded from an earlier application version only) **Note:** After upgrading to the Service Manager Classic applications (not PD-based), existing customers are provided with an option to enable Process Designer by running this command. When you run this command, all modules (Service Desk, Incident Management, Problem Management, Change Management, and Request Fulfillment) are upgraded to use the Process Designer workflows. There is not an option to migrate these modules to Process Designer on a module-by-module basis.	• *Service Manager 9.40 Installation and Upgrade Documentation Center* • Appendix A in this document

Migration Step	Tool	Document(s)
4. Configuration and Re-Customization	For configuration: • Enable Tool • Administration and Configuration For re-customization: • Process Designer workflows • Process Designer RuleSets and Actions • Process Designer Transition • Process Designer approval/alerts	• The *Application Setup* section in the online help provides instructions on how to implement your customizations within the Process Designer framework. • *Process Designer Tailoring Best Practices Guide* • This document
5. Data Migration	The Data Migration tool. It supports the following two modes: • Batch update mode: Support volume migrations, such as Interactions and Incidents. • Copy or update mode: Supports individual table copies.	• This document

This Process Designer *9.40 Migration Guide* is intended for both administrators and developers. As such, different sections may be more useful to you depending on your particular role.

Tips and checklists for upgrading your customization information

Considering the complexity of the application code structure in the Process Designer framework, a code migration script is not provided for the Process Designer based modules. The following table provides a summary of how process migration and transactional data migration are performed for different modules.

Module	Process Migration	Transactional Data Migration
Change Management	Automatically migrated by the PD enablement script	Automatically migrated by the PD enablement script
Service Desk	Manually migrated (see "PD Process Migration" on page 17)	By the data migration tool (see "How to use the data migration tool" on page 25) - update the “incidents” table
Incident Management	Manually migrated (see "PD Process Migration" on page 17)	By the data migration tool (see "How to use the data migration tool" on page 25) - update the“probsummary” table
Problem Management and Problem/Problem Task	Manually migrated (see "PD Process Migration" on page 17)	By the data migration tool (see "How to use the data migration tool" on page 25) - update the “rootcause” and “rootcausetask” tables
Problem Management and Known Error/Known Error Task	Manually (see "PD Process Migration" on page 17)	By the data migration tool (see "How to use the data migration tool" on page 25) - either update the “knownerror” and “knownerrortask” tables, or move data from the “knownerror” to the “rootcause” table, and related attachments

Module	Process Migration	Transactional Data Migration
Request Fulfillment	Not recommended	By the data migration tool (see "How to use the data migration tool" on page 25) - move data from the “ocmq”/”ocmo”/”ocml” to the “request”/”requestTask” table and related approvals, attachments
Service Level Management	N/A	By the data migration tool (see "How to use the data migration tool" on page 25) - update the “sla” table

Review the tips and checklists in the following sections before you upgrade your customization information to the Process Designer framework.

Help Desk

For the Process Designer Help Desk module, no code migration script is provided. This is because there is no workflow concept in the Service Desk and Incident modules, and these concepts are only introduced in Service Manager 9.40 with Process Designer enabled.

In the new Help Desk modules, new formats, display screens, and display options are used to reduce the chance of conflicts during the installation process. If some legacy formats, formatctrls, display options and display screens are still needed, you should re-implement the customization as a new format, display screen, or display option.

Most processes can be re-used, so we recommend that you merge the customization during the conflict resolution steps in the Service Manager 9.40 applications upgrade process.

Changes to Categories in Process Designer Help Desk

The legacy Incident, Interaction and Problem modules share the same category, subcategory, and producttype files. However, in new Process Designer Help Desk modules, these categories are managed separately. Therefore, if a new category is created in a legacy category table, you must migrate to the corresponding category files (such as the imCategory/imSubcateory/imArea for the probsummary file). The mapping relationship is category to imCategory, subcategory to imSubcategory and producttype to imArea and so on for the other Help Desk modules. For more information on how to use the migration tool to migrate category data, see "How to use the data migration tool" on page 25.

PD Help Desk also introduces the concept of shared categories. This concept exists because there may be many legacy Help Desk scenarios in which categories between Incident and Interaction do not match (such as when an end user submits a request for a support catalog item, and the categories must be shared to prevent Service Manager from failing to automatically create the Incident). If the customized category is also shared by two or three modules, you must migrate the data for each module category file that has the “Apply To” flag marked.

In the new category file, some legacy business logic that is defined on the category level are obsoleted. The following table provides suggestions for any customization based on those features:

Feature	Upgrade Suggestion (if there is still customization)
Category Alerts	This is a legacy, light-weight SLA feature. We recommend that you re-implement the Alerts using the Alert Definition functionality. Alternatively, you may use the Service Level Management module for those customers who have the SLM module installed. **Note:** For Re-assignment Thresholds, we recommend that you replace them with an Alert and a Notification to inform the Reassignment Group configured in the Assignment Table.
Formats	Display Formats settings should be re-mapped to the Format setting in a Process Designer workflow. If you customized a format in the legacy out-of-box, you must re-map the format to workflow-based format. **Note:** We recommend that you move any legacy Print Formats setting to client printing. This is because the Print Formats setting is related to server side printing, which is obsolete and replaced by client printing.
Assignment Group name/Assignment expression	The Assignment Group name and the Assignment expression have two use cases: 1. Used together with SLA alerts and defined on legacy category (for Incident only) as a Stage 1 alert group. In this use case, we recommend that you move these definitions to the SLM module. 2. Used as default assignment group when an Incident is created. In this use case, we recommend that you replace them with a workflow rule set, by creating a JavaScript-based assignment lookup rule.

Open options	We recommend that you replace Open scripts with an OnAdd RuleSets at the workflow level. We recommend that you replace the Copy/Open Link by using the copy record functionality.
Old style print options	Old print options are related to server side printing, which is obsolete and replaced by client printing.

Note: Service Desk approvals are moved to a new sdCategory file. If you have any additional customized approval definitions, you must re-define them in the new sdCategory file.

Your implementation may also have some business control logic that is defined in the Problem phases. If so, we recommend that you move these to RuleSets.

Process Designer framework

Process Designer Tailoring Best Practices Guide

We recommend that you use the new Process Designer tailoring methods provided by the Process Designer framework. It is ideal to move all business logic to the new Process Designer methods.

For information about best practices for using the Process Designer framework tailoring methods, see the *Process Designer Tailoring Best Practices Guide*.

However, you can still leverage the legacy tailoring methods used in previous versions (such as format control, data policy, and triggers). In other words, you can use both the new and legacy tailoring methods in a hybrid mode.

Leveraging legacy tailoring

The following is an example of how to leverage legacy tailoring.

You have upgraded an existing system, and in Incident Management you created a “Legacy Workflow” with the following phases: Logging, Update, Resolved, and Closure. After that, you assign the existing forms to each phase (open to Logging, update to Update, and close to Resolved and

Closure). Next, you migrate the category to imCategory file. You selectively add the rulesets needed to the initialization variables, but the business logic remains in FC. Last, you open a new Incident record, update, resolve and close it. All legacy FC is called and functions normally.

Menu

After Process Designer (PD) is enabled and the new PD- based modules are available in the left-side menu of Service Manager, the old-style menu entry points for the legacy Service Desk, Incident, Problem, Change and Request modules are prefixed by “Legacy.” To avoid confusion for your users, we recommend that you either remove or rename these legacy modules after the code upgrade is finished. Also, after you run the Enablement script, these legacy entries point to legacy functionality which may no longer work because the legacy code base and the new Process Designer code base are not guaranteed to co-exist.

You may need to move any customized menu link from a legacy sub-menu tree to a new location. For example, you may want to move a customized sub-menu tree from the legacy Incident Management to the new Incident Management.

Additionally, new or customized inbox queries in Favorites and Dashboards may need to be re-defined if the query is based on now-obsolete field values (for example, a mismatched ticket status value or an obsolete file name such as knownerror).

The legacy menu tree’s security control is based on legacy profile variables and capability words. Because Process Designer maintains backward compatibility for security, and legacy profile variables are still populated, the legacy menu security control can still work. However, this does not mean that the menu link itself will function. Instead, it determines merely whether or not the control for this menu’s visibility will still work. However, it is strongly recommended that you use the Process Designer security control methods to unify the controls, especially for newly-added menu links.

As a best practice, we recommend that you start from the new out-of-box menus and add any customized elements on a case-by-case basis . Trying to find all the legacy menu items that may not work after the PD enablement would be more difficult.

Change

In the Change module, customizations on format, formatctrl, display option, display screen, process still require manual code upgrade from legacy code to the Process Designer framework. For more information, see the *Process Designer Tailoring Best Practices Guide*.

Service Level Agreements (SLAs)

The legacy Response Service Level Target (SLT) definitions are based on the legacy status (or phases for the Problem module). After Process Designer is enabled and the Help Desk data migration is complete, you must manually update the legacy Response SLO definitions according to how you migrated the old statuses and problem phases, and your business needs for SLA based on new Service Desk, Incident, Problem, and Change statuses and phases. Use the following table as a guideline:

Information Type:	**Where to Access:**	**Action:**
SLT Catalog definition	**Service Level Management** > **Agreements** > **SLT Catalog**	Update the initial/final state of legacy SLT catalog which are defined based on old statuses (or old phases) to new reasonable statuses (or new phases).
SLT definition	**Service Level Management** > **Agreements** > **Process Targets**	Update the initial/final state of legacy Response SLTs which are defined based on old statuses (or old phases) to new reasonable statuses (or new phases).

Note: If the Batch Updated mode is used during data migration on non-closed Incident, Interaction, and Problem tickets, the corresponding Response SLT calculations may be inaccurate. This is because, in Batch Mode, the SLA triggers for these tickets may not be triggered.

Therefore, we recommend that you redefine the SLT and SLA definitions (based on the previous table) first, and then migrate the open Incident, Interaction, and Problem tickets in non-Batch Update mode. For closed tickets, you can still migrate data by using the Batch Update mode to gain the performance advantage.

Chapter 2: PD Process Migration

Service Manager 9.40 does not provide an out-of-box process migration script for Help Desk modules. This section provides guidelines on how to migrate your legacy processes to Process Designer (PD) workflows.

Because of the variety of customer environments, the guidelines may not work well in your environment, and therefore extra effort may be required on a case-by-case basis.

In the Change Management module, the legacy processes are migrated to PD workflows automatically by the PD enablement script. For this reason, the Change process migration is not described in this chapter. After PD enablement, all Change processes are run as PD workflows - workflows that are associated to the Emergency Change, Standard Change, Normal Change, or Change Proposal category are newly introduced by PD Change, while other workflows are migrated from legacy Change processes.

The Request Fulfillment module is re-implemented on Process Designer, and many of its features are different from those of the legacy Request Management module; additionally, new tables and document engine objects were introduced in PD Request Fulfillment. For these reasons, we do not recommend existing customers to migrate legacy Request Management processes to PD workflows.

However, some out-of-box data migration utilities are provided for the Request Fulfillment module, which will help customers keep their existing investment to the most extent. For more information, see "Request data migration" on page 54.

Service Desk

Based on the legacy Service Desk process, you can create a corresponding workflow with three phases in Process Designer.

The following table provides information about each phase in the PD workflow.

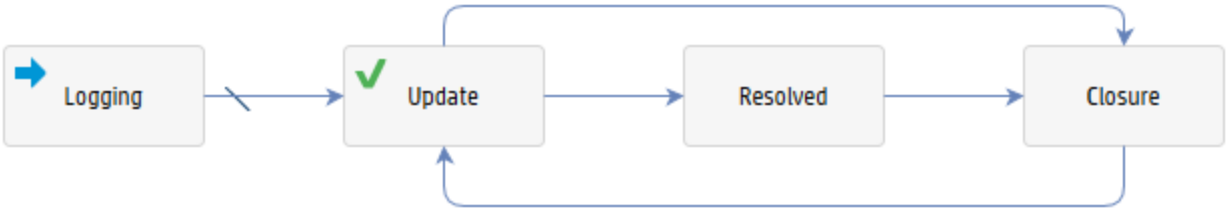
PD Workflow Phase	PD Workflow Form	Format Control	Alert in PD Workflow	PD Workflow Transition(s)
Logging	SD.open.interaction	SD.open.interaction	Migrate from the “category” table	Click **Close Interaction** to go to the Closure phase for the first call resolution; Click **Escalate** to Escalation phase
Escalation	SD.update.interaction	SD.update.interaction	Migrate from the “category” table	Click **Close Interaction** to go to the Closure phase
Closure	SD.update.interaction	SD.update.interaction	Migrate from the “category” table	N/A

We recommend that you use the following steps for your manual migration:

1. Migrate category values from the “category” to the “sdCategory” table, including the Service Desk Approval definition.
2. Create a workflow as shown above in Process Designer.
3. Follow the code migration suggestions described in "Tips and checklists for upgrading your customization information" on page 11.
4. Migrate the transactional data in your legacy process to the migrated PD workflow, by using the data migration tool described in this document. To do so, you need to create your own migration settings in the data migration tool. For more information, see "How to use the data migration tool" on page 25.

Incident Management

Based on the legacy Incident Management process, you can create a workflow with four phases in Process Designer.



The following table provides information about each phase in the PD workflow.

PD Workflow Phase	PD Workflow Phase Form	Format Control	Alert in PD Workflow	PD Workflow Transition(s)
Logging	IM.open.incident	IM.open.incident	Migrate from "category" table	Clicking **Save** goes to the Update phase by default
Update	IM.update.incident	IM.update.incident	Migrate from "category" table	If the two-step closure functionality is enabled, click **Resolve** to go to the Resolved phase; If the two-step closure functionality is disabled, click **Close Incident** to go to the Closure phase
Resolved	IM.close.incident	IM.close.incident	Migrate from "category" table	Click **Close Incident** to go to the Closure phase
Closure	IM.close.incident	IM.close.incident	Migrate from "category" table	Click **Reopen** to revert to the Update phase.

We recommend you use the following steps for your manual migration:

1. Migrate category values from the "category" to the "imCategory" table.
2. Create a workflow as shown above in Process Designer.
3. Follow the code migration suggestions described in "Tips and checklists for upgrading your customization information" on page 11.
4. Migrate the transactional data in your legacy process to the migrated PD workflow, by using the

data migration tool described in this document. To do so, you need to create your own migration settings in the data migration tool. For more information, see "How to use the data migration tool" on page 25.

Problem Management

Problem Management includes Problems and Known Errors.

Problem migration

Problem

Based on a legacy Problem workflow, you can create a corresponding PD workflow. The legacy problem workflow information is saved in the Problem category table (rootcausecat), and the details of the legacy workflow phase can be retrieved from the Problem phase table (rootcausephase).

The following figure shows the PD problem workflow.

The following table provides information about each phase in the PD workflow.

PD Workflow Phase	PD Workflow Phase Form	Format Control	Alert in PD Workflow	PD Workflow Transition (s)
Problem Detection, Logging and Categorization	Migrate from the “rootcausephase” table	Form-specific	Migrate from the “rootcausephase” table	Manually move to the next phase
Problem Prioritization and Planning	Migrate from the“rootcausephase” table	Form-specific	Migrate from the “rootcausephase” table	Manually move to the next phase
Problem Investigation and Diagnosis	Migrate from the “rootcausephase” table	Form-specific	Migrate from the “rootcausephase” table	Manually move to the next phase

PD Workflow Phase	PD Workflow Phase Form	Format Control	Alert in PD Workflow	PD Workflow Transition (s)
Problem Resolution	Migrate from the “rootcausephase” table	Form-specific	Migrate from the “rootcausephase” table	Manually move to the next phase
Problem Closure and Review	Migrate from the “rootcausephase” table	Form-specific	Migrate from the “rootcausephase” table	Click **Close Problem** to the Closure phase
Closure	Same as the form of Problem Closure and Review phase	Read-only	N/A	Click **Reopen** to return to the Problem Closure and Review phase

We recommend you use the following steps for your manual migration:

1. Migrate category values from the “rootcausecat” to the “pbmCategory” table.
2. Create a PD workflow as shown above.
3. Follow the code migration suggestions described in "Tips and checklists for upgrading your customization information" on page 11.
4. Migrate the transactional data in your legacy process to the migrated PD workflow, by using the data migration tool described in this document. To do so, you need to create your own migration settings in the data migration tool. For more information, see "How to use the data migration tool" on page 25.

Problem Task

The legacy Problem Task module does not support workflows, and so you can create a simple two-phase workflow in Process Designer.

The following table provides information about each phase in the PD workflow.

PD Workflow Phase	PD Workflow Phase Form	Format Control	PD Workflow Transition(s)
Logging	PM.task	Migrate from the “rootcausetask” format control	Click **Close Task** to the Closure phase
Closure	PM.task	Migrate from the “rootcausetask” format control	Click **Reopen** to revert to the Logging phase

We recommend you use the following steps for your manual migration:

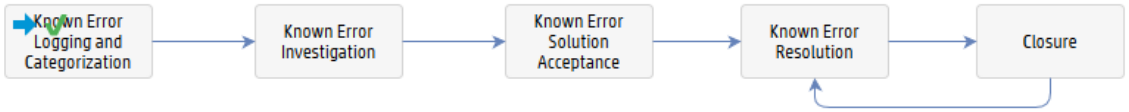
1. Migrate category value from the “rootcausetaskcat” table to the “pbmTaskCat” table .
2. Create above workflow in PD.
3. Follow the code migration suggestions described in "Tips and checklists for upgrading your customization information" on page 11.
4. Migrate the transactional data in your legacy process to the migrated PD workflow, by using the data migration tool described in this document. To do so, you need to create your own migration settings in the data migration tool. For more information, see "How to use the data migration tool" on page 25.

Known Error migration

As in PD Help Desk, Known Error records are saved in the same table of Problem (rootcause). If you want to keep your existing Known Error and Known Error Task processes, you may follow the instructions below to migrate your processes to PD. In this case, since the migrated Known Error and Known Error Task records are still saved in the original tables (knownerror and knownerrortask), you might need to revamp the related records feature between the migrated Known Error/Known Error Task and other processes. If you don’t want to keep your existing Known Error and Known Error Task processes, you may use the out-of-box settings in the data migration tool to migrate your existing Known Error records to the out-of-box PD Known Error workflow.

Known Error

Based on legacy the Known Error workflow, the corresponding PD workflow can be created. The legacy Known Error workflow information is saved in the Known Error category table (knownerrorcat), and the details of the legacy workflow phase can be retrieved from the Known Error phase table (knownerrorphase).



PD Workflow Phase	PD Workflow Phase Form	Format Control	Alert in PD Workflow	PD Workflow Transition (s)
Known Error Logging and Categorization	Migrate from the “knownerrorphase” table	Form-specific	Migrate from the “knownerrorphase” table	Manually move to the next phase
Known Error Investigation	Migrate from the “knownerrorphase” table	Form-specific	Migrate from the “knownerrorphase” table	Manually move to the next phase
Known Error Solution Acceptance	Migrate from the “knownerrorphase” table	Form-specific	Migrate from the “knownerrorphase” table	Manually move to the next phase
Known Error Resolution	Migrate from the “knownerrorphase” table	Form-specific	Migrate from the “knownerrorphase” table	Click **Close Known Error** to Closure phase
Closure	Same as the form of Known Error Resolution phase	Read-only	N/A	Click **Reopen** to revert to the Known Error Resolution phase

The following are the suggested steps you might want to follow for your manual migration.

1. Enable PD workflow and security in the “knownerror” object.
2. Create the above workflow in PD.
3. Follow the code migration suggestions described in "Tips and checklists for upgrading your customization information" on page 11.
4. Through the data migration tool, migrate the transactional data in your legacy process to the migrated PD workflow. To achieve it, you need to create your own migration settings in the data migration tool. For more information, refer to "How to use the data migration tool" on page 25.

Known Error Task

Legacy Known Error Task doesn’t support workflow, so a simple two-phase workflow will be created in PD.

PD Workflow Phase	PD Workflow Phase Form	Format Control	PD Workflow Transition(s)
Logging	PM.error.task	knownerrortask	Click “Close Task” to Closure phase
Closure	PM.error.task	knownerrortask	Click “Reopen” to revert to the Logging phase

The following are the suggested steps you might want to follow for your manual migration:

1. Enable PD workflow and security in the “knownerrortask” object.
2. Create the above workflow in PD.
3. Follow the description in "Tips and checklists for upgrading your customization information" on page 11.
4. Through the data migration tool, migrate the transactional data in your legacy process to the migrated PD workflow. To achieve it, you need to create your own migration settings in the data migration tool. For more information, refer to "How to use the data migration tool" on page 25.

Chapter 3: How to use the data migration tool

You can use the data migration tool to migrate data from legacy process records to new process records within the same Service Manager system. It is not used to migrate data across different Service Manager systems. You can configure the migration tool from the GUI, and it requires administrator privileges.

The migration tool allows an operator to configure the field/value mapping between two files. It also covers customized fields. The out-of-box Process Designer provides several sample migration scripts for the following scenarios.

Caution: During data migration, there are no dependencies between the modules, however the scripts for each specific module must be executed in the listed order. Additionally, scripts that are marked with an asterisk **(*)** in the following table cannot be run in Batch Update mode. For more information about this mode, see "Batch Update mode" on page 34.

Module	Script	Notes
Request	1. *Legacy Quote to new Request* ***(*)*** 2. *Legacy Quote Line Item to new RequestTask* ***(*)*** 3. *Legacy Order to new Request* ***(*)*** 4. *Legacy Order Line Item to new RequestTask* ***(*)*** 5. Legacy attachment to new attachment for new Request Module 6. Legacy Approval to new Approval for new Request 7. Legacy ApprovalLog to new ApprovalLog for new Request Module	
PD Framework	1. *Legacy change model taskplanner data to new taskplanner* ***(*)*** 2. Legacy change instance taskplanner data to new taskplanner	Both scripts apply to customers who have upgraded from Process Designer Content Pack 9.30.3 to version 9.40.
Incident	1. Legacy incident to new incident	

Module	Script	Notes
Interaction	1. Legacy interaction to new interaction	
Problem	1. Legacy problem to new problem 2. Legacy problem task to new problem task 3. *Legacy known error to new known error **(*)*** 4. Legacy PD known error to new PD known error 5. Legacy known error related record migration 6. Legacy known error attachment to new known error attachment	The **Legacy PD known error to new PD known error** script applies to customers who have upgraded from Process Designer Content Pack 9.30.3 to version 9.40.
Service Level Agreement	1. *Legacy SLA to new Agreement **(*)***	This script applies to all customers who have upgraded to Service Manager 9.40 (Codeless or Classic).

If further migrations are needed, an administrator can use these samples to define their own. Considering that Incident and Service Desk typically have a large data volume, the Data Migration tool supports an SQL batch update mode to ensure good performance.

Note: We recommend that you back up the production data before migration.

How to access the tool

To access the Data Migration tool, follow these steps:

1. Type `db` in the command line to open Database Manager.
2. Type **migrationSetting** in the **Table** field, and then click **Search**.

   The Data Migration Tool form opens. The following table describes the fields and options on this form.

Field	Description
Name	Migration script name (must be unique)
Description	No longer than 600 characters.
Source table	The source table from which you will migrate data.
Target table	The target table to which you migrate data. If the Target table = Source table, it is updated on same table.
Query	Defines an initial query condition on the source table.
Filter	Defines a data filter that further filters the data returned by the query for migration.
Fields Mapping Definition	Defines field mappings between the source and target tables.
Value Mapping Definition	Defines value mappings between the source and target tables.
Post Script	JavaScript expressions that should be executed during the migration.
Batch Update	Enable this option when the target file equals the source file. For more information, see "Batch Update mode" on page 34.

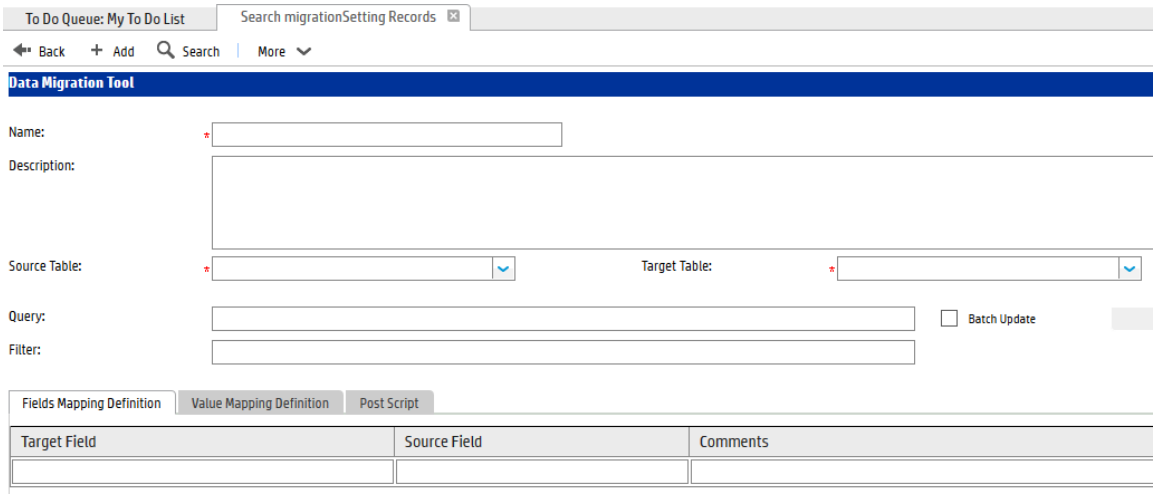
3. To create your own migration script, click **Add**.
4. To find existing migration scripts, click **Search**.

How to run a migration script

Note: If you are upgrading an existing Service Manager system to Process Designer, you need to customize the legacy out-of-box migration script samples for your specific environment.

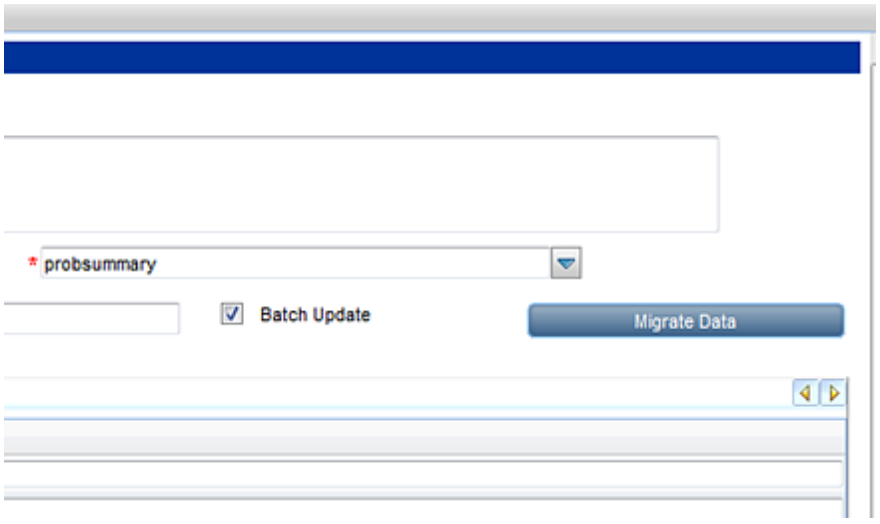
To run a migration script, follow these steps:

1. Enter the name of the migration script that you wish to run, and then click **Search**.



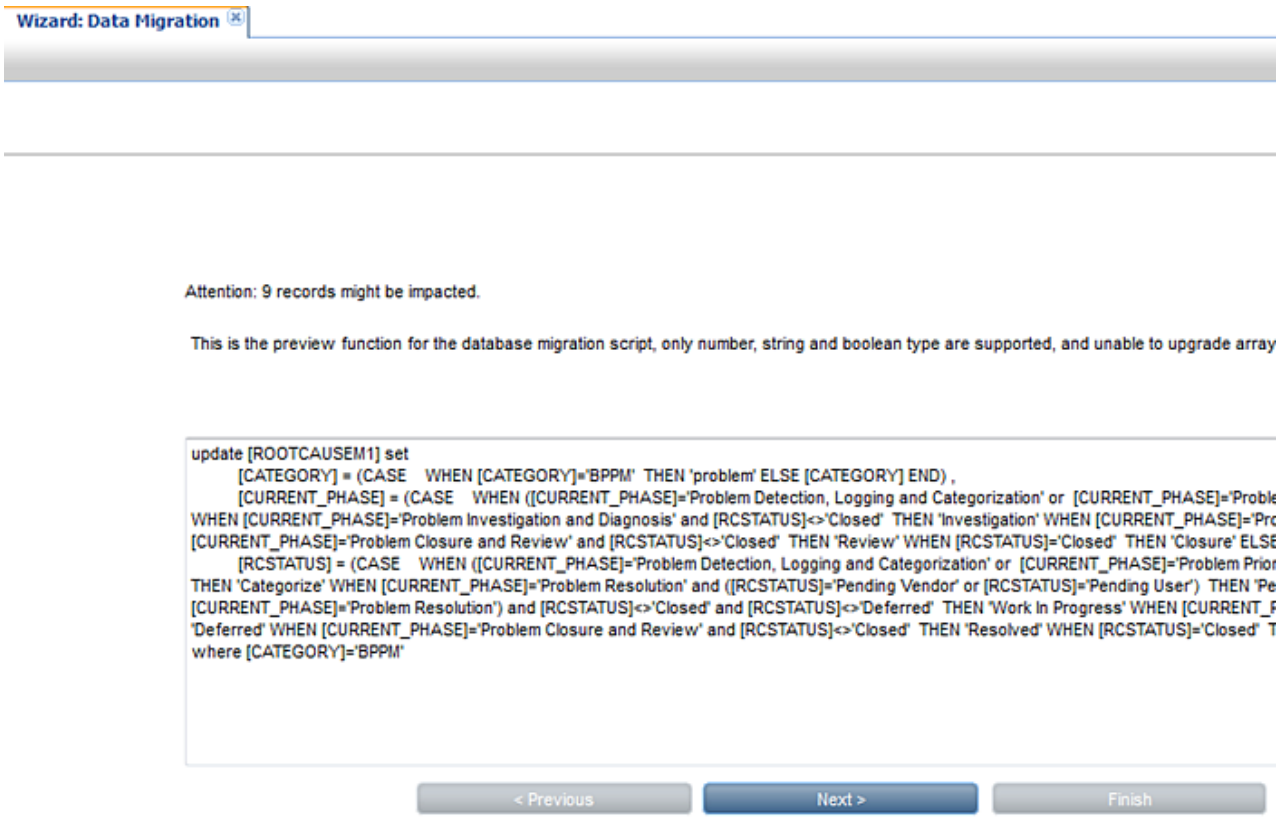
2. In the Migration Script setting detail form, click **Migrate Data** to run the migration script.

> **Note:** You can run the migration in Batch Update mode to get better performance when the source and target files are the same. The Batch Update mode cannot be used when you migrate data between two different files, such as from Legacy known error to Problem. For a list of scripts that you cannot run in Batch Update mode, see "How to use the data migration tool" on page 25.



3. The Data Migration pop-up window shows the RDBMS SQL Statements that will be run on the

database. Change these SQL statements if needed:

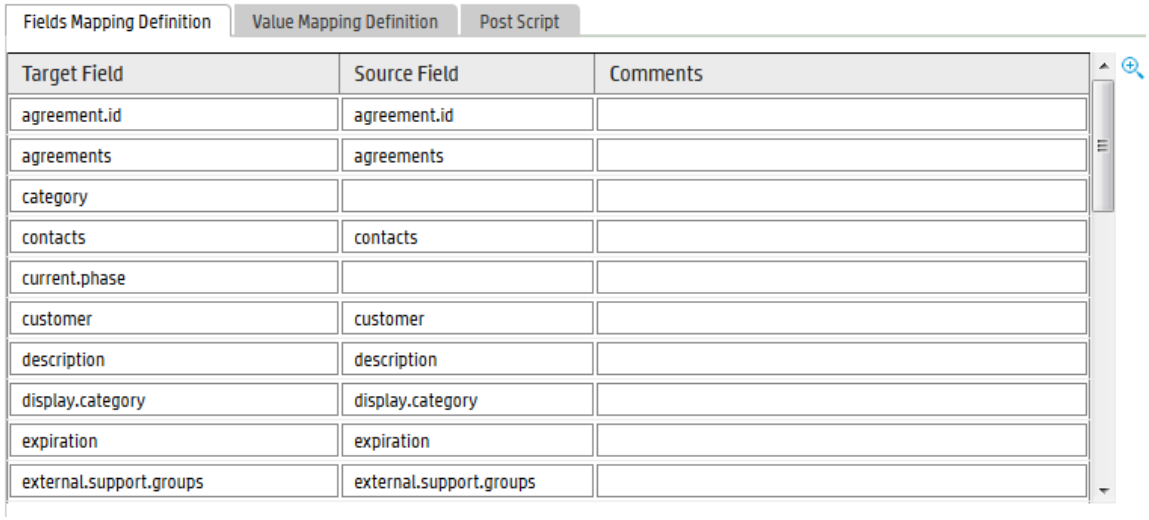


4. Check that the SQL statements are correct, click **Next**, and then click **Finish** to start the migration in Batch Update mode.

> **Note:** If there is no data to migrate, or if the data is already migrated, you receive the following notification:
>
> "No records match the filter condition. At least one matching record is required."

Field mapping and value mapping

Field mapping defines the mapping between the fields in the source file and target file, as shown in the following figure.

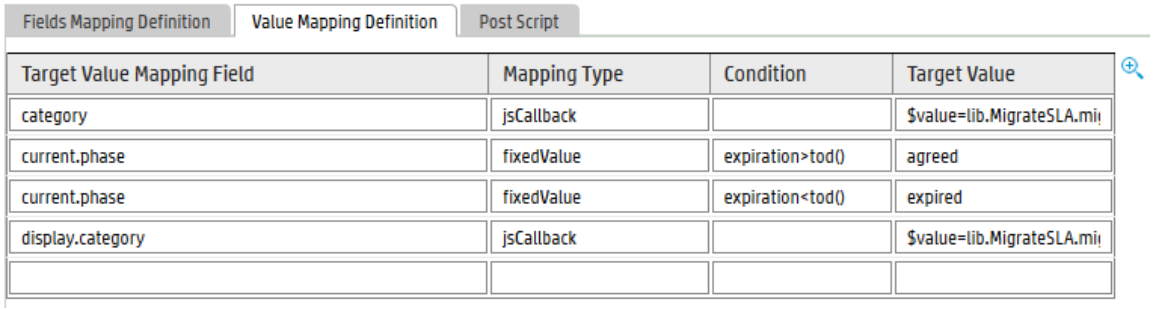


Note:

- The Source and Target fields cannot be empty.
- The Source and Target fields must already exist in the table.
- If there is a structure field mapping and you are not in Batch Update mode, both fields must have the exact same structure definition.

- Array of structure and structure fields are not supported when in Batch Update mode.
- The M2 and alias table are not supported in Batch Update mode.
- You cannot use jscallback in Batch Update mode.

Value mapping defines the value mapping for each mapped field.



Note:

- For the fields that do not have a value mapping defined, the default is direct value copy.
- You can define multiple value mappings for the same field.

The following table describes each column in the Value Mapping Definition tab.

Column	Description
Target value mapping field	The field from target field that needs to define value mapping.

Column	Description

Mapping Type	The following three types are supported:
	1. **fixedValue:** target field value = the fixed value defined in “Target Value” column. The supported fixed value are number, string, datetime, and Boolean. You cannot specify an expression in a fixed table, nor a fixed value for structure and array field type.
	2. **sourceField:** target field value = source field value defined in “Target Value” column.
	3. **jsCallback:** target field value = a JavaScript script to dynamically set the value. Three keywords ($sourceTable, $targetable, $value) can be used if the direct script code is written in the Target Value column:
	▪ $sourceTable, $targetTable reference the field value in the source table field and the target table field, respectively as shown in the following example:
	`var status = $sourceTable[“rcStatus”],$targeTable[“rcStatus”] = “open”`
	▪ $value is used to specify the current target field value as shown in the following example:
	`$value=vars.$G_user_role retrieves the value from another variable.`
	jsCallback can also support directly calling a JavaScript function defined in Script library, such as the following:
	`lib.UpgradeScript.migrateData($sourceTable,.$targetTable);`

Column	Description
Condition	If the condition is true, then the value mapping is applied during the migration. The condition expression supports the following: 1. Arithmetic operators: >, <, >=, <=, =, ~= 2. Logical operators: or, and 3. The “null” keyword can be used as a null value in the formula.
Target Value	Depending on the Mapping Type setting, Target Value can be a fixed value string, or a source field name or a JavaScript script.

Batch Update mode

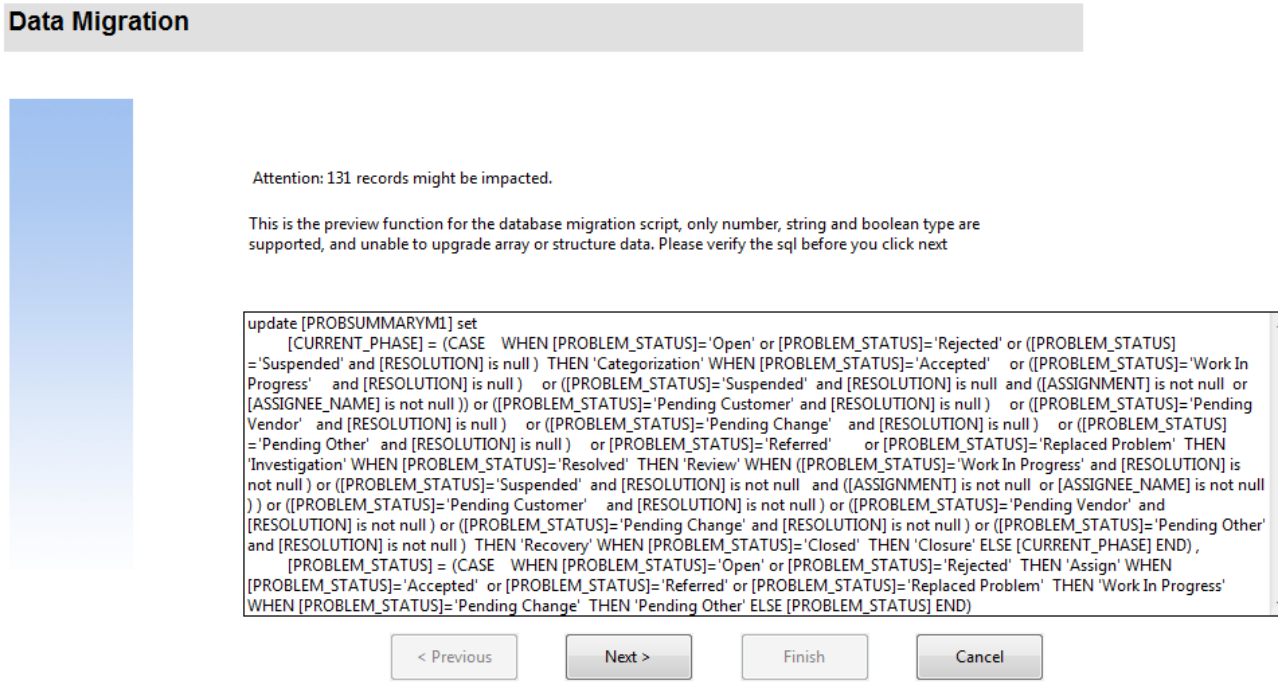
When the target file equals the source file, the migration tool executes the update operation on the same table. In addition, an additional SQL batch update mode is automatically enabled to allow the migration tool to directly update the underlying database.

Note: Some of the out-of-box migration scripts cannot be run in Batch Update mode. For more information, see "How to use the data migration tool" on page 25.

Batch update mode can only be enabled when the target file equals the source file. If it is enabled, a batch SQL update is generated in a popup wizard instead of using the Service Manager API to update the data records one by one. After confirmation, the batch SQL update is sent to the underlying database.

If Batch Update mode is enabled, you should note the following:

- You should update the data records directly via the database without triggering Service Manager. Otherwise, updating the data records will trigger additional background process.
- The Filter Condition field should follow the RDBMS SQL grammar for your database. Otherwise, Filter Condition should follow query statement grammar of Service Manager.
- The “Condition” field in the Value Mapping Definition should follow the RDBMS SQL grammar for your database. Otherwise, Filter Condition should follow query statement grammar of Service Manager.



Out-of-box data migration settings

The Data Migration Tool provides the following out-of-box data migration settings.

Interaction data migration

Considering that Interaction does not need to copy data from two different tables, and that it typically has a large volume of data, we recommend that you run the migration by using the batch update mode (especially for migrating a large volume of closed records).

The out-of-box example data migration is based on the following three key fields:

- status
- category
- current.phase

Note: You can add additional fields if needed, but do not remove these three fields.

Default settings:

Source Table	incidents
Target Table	incidents
Filter Condition	current.phase=null

Field Mapping:

Target Field	Source Field
active	
current.phase	current.phase
open	open

Value Mapping:

Target Value Mapping Field	Mapping Type	Condition	Target Value
active	fixedValue	open~="Closed"	true
current.phase	fixedValue	(open="Open - Callback" and category~="service catalog") or (open="Open - Linked" and resolution~=null)	Review
current.phase	fixedValue	open="Open - Linked" and resolution=null	Work In Progress
current.phase	fixedValue	(open="Open - Idle" and category~="service catalog") or (category="service catalog" and (approval.status="pending" or approval.status="denied"))	Categorization
current.phase	fixedValue	open="Closed"	Closure
current.phase	fixedValue		Work In Progress
open		open="Open - Callback" and category~="service catalog"	Callback
open	fixedValue	open="Open - Linked" and resolution=null	Dispatched
open	fixedValue	open="Open - Linked" and resolution~=null	Resolved

Target Value Mapping Field	Mapping Type	Condition	Target Value
open	fixedValue	open="Open - Idle" and category~="service catalog"	Categorize
open	fixedValue	open="Closed"	Closed
open	fixedValue	category="service catalog" and (approval.status="pending" or approval.status="denied")	Assign
open	fixedValue		In Progress

Incident data migration

Considering that Incident does not need to copy data from two different tables, and that it typically has a large volume of data, we recommend that you run the migration by using the batch update mode (especially for migrating a large volume of closed records).

The out-of-box example data migration is based on the following three key fields:

- status
- category
- current.phase

Note: You can add additional fields if needed, but do not remove these three fields.

Default settings:

Source Table	probsummary

Target Table	probsummary
Filter Condition	current.phase=null

Field Mapping:

Target Field	Source Field
current.phase	
problem.status	problem.status

Value Mapping:

Target Value Mapping Field	Mapping Type	Condition	Target Value
current.phase	fixedValue	problem.status="Open" or problem.status="Rejected" or (problem.status="Suspended" and assignment=null and assignee.name=null)	Categorization
current.phase	fixedValue	problem.status="Accepted" or (problem.status="Work In Progress"and resolution=null)or (problem.status="Suspended"and resolution=nulland (assignment~=null or assignee.name~=null)) or (problem.status="Pending Customer"and resolution=null)or (problem.status="Pending Vendor"and resolution=null)or (problem.status="Pending Change"and resolution=null)or (problem.status="Pending Other"and resolution=null)or problem.status="Referred"or problem.status="Replaced Problem"	Investigation
current.phase	fixedValue	problem.status="Resolved"	Review

Target Value Mapping Field	Mapping Type	Condition	Target Value
current.phase	fixedValue	(problem.status="Work In Progress" and resolution~=null) or (problem.status="Suspended"and resolution~=nulland (assignment~=null or assignee.name~=null)) or (problem.status="Pending Customer"and resolution~=null) or (problem.status="Pending Vendor"and resolution~=null) or (problem.status="Pending Change" and resolution~=null) or (problem.status="Pending Other"and resolution~=null)	Recovery
current.phase	fixedValue	problem.status="Closed"	Closure
current.phase	sourceField		current.phase
problem.status	fixedValue	problem.status="Open" or problem.status="Rejected"	Assign
problem.status	fixedValue	problem.status="Accepted" or problem.status="Referred" or problem.status="Replaced Problem"	Work In Progress
problem.status	fixedValue	problem.status="Pending Change"	Pending Other
problem.status	sourceField		problem.status

Problem data migration

Considering that Problem does not need to copy data from two different tables, and that it typically has a large volume of data, we recommend that you run the migration by using the batch update mode (especially for migrating a large volume of closed tickets).

Note: After you migrate the Problem data, remove the “Problem_Library_disabled_by_PDHD” knowledge base as this legacy library is no longer used.

The out-of-box example data migration is based on the following three key fields:

- status
- category
- current.phase

Note: You can add additional fields if needed, but do not remove these three fields.

Problem

Default settings:

Source Table	rootcause
Target Table	rootcause
Filter Condition	category="BPPM"

Field Mapping:

Target Field	Source Field
category	category
current.phase	current.phase
rcStatus	rcStatus

Value Mapping:

Target Value Mapping Field	Mapping Type	Condition	Target Value
category	fixedValue	category="BPPM"	problem
current.phase	fixedValue	(current.phase="Problem Detection, Logging and Categorization" or current.phase="Problem Prioritization and Planning") and rcStatus~="Closed"	Categorization
current.phase	fixedValue	current.phase="Problem Investigation and Diagnosis" and rcStatus~="Closed"	Investigation
current.phase	fixedValue	current.phase="Problem Resolution" and rcStatus~="Closed"	Resolution
current.phase	fixedValue	current.phase="Problem Closure and Review" and rcStatus~="Closed"	Review
current.phase	fixedValue	rcStatus="Closed"	Closure
rcStatus	fixedValue	(current.phase="Problem Detection, Logging and Categorization" or current.phase="Problem Prioritization and Planning") and rcStatus~="Closed" and rcStatus~="Deferred"	Categorize
rcStatus	fixedValue	current.phase="Problem Resolution" and (rcStatus="Pending Vendor" or rcStatus="Pending User")	Pending
rcStatus	fixedValue	(current.phase="Problem Investigation and Diagnosis" or current.phase="Problem Resolution") and rcStatus~="Closed" and rcStatus~="Deferred"	Work In Progress
rcStatus	fixedValue	current.phase~="Problem Closure and Review" and rcStatus="Deferred"	Deferred
rcStatus	fixedValue	current.phase="Problem Closure and Review" and rcStatus~="Closed"	Resolved
rcStatus	fixedValue	rcStatus="Closed"	Closed

Problem Task

Default Settings:

Source Table	rootcausetask
Target Table	rootcausetask
Filter Condition	task.category="Default"

Field Mapping:

Target Field	Source Field
task.category	task.category
current.phase	current.phase
rcStatus	rcStatus

Value Mapping:

Target Value Mapping Field	Mapping Type	Condition	Target Value
task.category	fixedValue		Investigation
current.phase	fixedValue	rcStatus~="Closed"	Active
current.phase	fixedValue	rcStatus="Closed"	Closure
rcStatus	fixedValue	rcStatus="Work In Progress"	Work In Progress
rcStatus	fixedValue	rcStatus="Pending Vendor" or rcStatus="Pending User"	Pending
rcStatus	fixedValue	rcStatus="Closed"	Closed
rcStatus	fixedValue	rcStatus~="Work In Progress" and rcStatus~="Pending Vendor" and rcStatus~="Pending User" and rcStatus~="Closed"	Assigned

Task Planner data migration

In versions of Service Manager earlier than 9.40, Task Planner data is stored in the changeModel and changePlan tables. However, in SM 9.40, Task Planner data is stored in the changePlan table only. Therefore, two data migration settings are provided to migrate Task Planner data to Service Manager 9.40 from earlier versions.

Migrate legacy change model Task Planner data to new Task Planner

Default settings:

Source table	changeModel
Target Table	changePlan
Query	not null(tasks)
Batch Update	false

Field mapping:

Target field	Source field
fileName	
number	id

Value mapping:

Target value mapping field	Mapping type	Condition	Target value
fileName	fixedValue		changeModel

Post script:

```
for(var i=0;i<$sourceTable.tasks.length();i++){
	$targetTable.tasks[i].taskId=$sourceTable.tasks[i].taskId;
	$targetTable.tasks[i].taskCoords=$sourceTable.tasks[i].taskCoords;
	$targetTable.tasks[i].dependentIds=$sourceTable.tasks[i].dependentIds;
	$targetTable.tasks[i].dependentCoords=$sourceTable.tasks[i].dependentCoords;
	$targetTable.tasks[i].taskCategory=$sourceTable.tasks[i].taskCategory;
	$targetTable.tasks[i].taskTemplate=$sourceTable.tasks[i].taskTemplate;
	$targetTable.tasks[i].taskDescription=$sourceTable.tasks[i].taskDescription;
	$targetTable.tasks[i].openInPhase=$sourceTable.tasks[i].openInPhase;
	$targetTable.tasks[i].closeByPhase=$sourceTable.tasks[i].closeByPhase;
	$targetTable.tasks[i].activeCond=$sourceTable.tasks[i].activeCond;
	$targetTable.tasks[i].activeCondXML=$sourceTable.tasks[i].activeCondXML;
	$targetTable.tasks[i].activeCondDesc=$sourceTable.tasks[i].activeCondDesc;
	$targetTable.tasks[i].mandatory=$sourceTable.tasks[i].mandatory;
}
$targetTable.doUpdate();
```

Migrate legacy change instance Task Planner data to new Task Planner

Default settings:

Source table	changePlan
Target Table	changePlan
Query	null(fileName)
Batch Update	true

Field mapping:

Target field	Source field
fileName	fileName

Value mapping:

Target value mapping field	Mapping type	Condition	Target value
fileName	fixedValue		cm3r

Known Error data migration

Known Error data migration differs from Interaction, Incident, and Problem because Known Error data must be copied from the knownerror file to the rootcause file, and the field mapping is also more extensive than in other modules.

Because it is a data copy, you cannot use the batch update mode. However, knownerror data volume is usually much less than that of Incident or Interaction. Therefore, the normal copy mode should be sufficient for most environments.

Note: After you migrate the knownerror data, remove the “KnownError_Library_disabled_by_PDHD” knowledge base as this legacy library is no longer used.

The Data Migration Tool provides four migration settings, which you should run in their corresponding scenario as described in the following table.

Scenario	Migration setting(s)
Migrating from legacy Known Error	Legacy known error to new known error Legacy known error related record migration Legacy known error attachment to new known error attachment
Migrating from legacy PD Known Error	Legacy PD known error to new PD known error For more information about how to use this migration setting, see "Migrating from legacy PD known error records" on page 51.

Additionally, because knownerror data is copied and the records are re-created in the rootcause file, you must also update the screlation file to correct the relations of known errors. To do this, the out-of-box Process Designer provides another migration script called “Legacy known error related record migration”.

Note: Attachments for legacy knownerror records are migrated as attachments for the new known error records.

Default Settings:

Source Table	knownerror
Target Table	rootcause
Filter Condition	true

Field Mappings:

Target Field(Problem)	Source Field(Known Error)
id	id
category	category
assignment	assignment
status	status
logical.name	logical.name
brief.description	brief.description
description	description
root.cause	root.cause

Target Field(Problem)	Source Field(Known Error)
update	update
open	open
open.time	open.time
opened.by	opened.by
update.time	update.time
updated.by	updated.by
close.time	close.time
closed.by	closed.by
reopen.time	reopen.time
reopened.by	reopened.by
priority.code	priority.code
ticket.owner	ticket.owner
severity	severity
sysmodtime	sysmodtime
sysmodcount	sysmodcount
sysmoduser	sysmoduser
assignee.name	assignee.name
subcategory	subcategory

Target Field(Problem)	Source Field(Known Error)
product.type	product.type
problem.type	problem.type
company	company
dump	dump
resolution	resolution
workaround	workaround
incident.category	incident.category
current.phase	current.phase
incident.count	incident.count
initial.impact	initial.impact
future.impact	future.impact
impact	impact
users.affected	users.affected
problem.start.time	problem.start.time
expected.resolution.time	expected.resolution.time
location.type	location.type
frequency	frequency
proposed.solution	proposed.solution

Target Field(Problem)	Source Field(Known Error)
review.notes	review.notes
cause.code	cause.code
affected.ci	matching.ci
ci.device.name	matching.device.name
ci.device.type	matching.device.type
ci.assign.group	matching.assign.group
ci.location	matching.location
affected.companies	affected.companies
affected.ci.count	matching.ci.count
last.task.no	last.task.no
kpf.id	kpf.id
kpf.file	kpf.file
folder	folder
closure.code	closure.code
estimatedCost	estimatedCost
estimatedMandays	estimatedMandays
rootcauseDate	knownerrorDate
solutionDate	solutionDate

Target Field(Problem)	Source Field(Known Error)
affected.item	affected.item
rcStatus	rcStatus
interaction.count	interaction.count
publishWorkaround	publishWorkaround

Value Mappings:

Target Value Mapping Field	Mapping Type	Condition	Target Value
category	fixedValue	true	known error
current.phase	fixedValue	rcStatus~="Closed"	Logging
current.phase	fixedValue	rcStatus="Closed"	Closure
isKnownError	fixedValue	true	true
rcStatus	fixedValue	rcStatus~="Closed"	Open
rcStatus	fixedValue	rcStatus="Closed"	Closed

Migrating from legacy PD known error records

If you upgraded your system from an older Process Designer version, you need to migrate your old PD knownerror records to new PD knownerror records by using the **Legacy PD known error to new PD known error** migration setting. The migration process includes the following procedures:

1. Change the old Known Error record to Problem record, and then create a new Known Error record with a category of "known error."
2. Link the new Known Error record to the Problem record.

3. Copy or do not copy the original relations to the new Known Error record, based on the option you select in the **Post Script** tab of this migration setting, as described in the following table.

Option for relation processing	Description
Skip Copying Relations to New Known Error Records	This option does not copy original relations to new known error records.
Copy Relations to New Known Error Records	This option copies original relations to new known error records.

Service Level Management data migration

As of Service Manager 9.40, the Service Level Management (SLM) module is reimplemented on Process Designer (PD). If you upgraded from an earlier version of Service Manager to version 9.40 or later, you need to migrate your legacy SLM data so that your SLM module can work correctly on PD-based workflows.

The purpose of SLM data migration is to set a category and set a phase for agreement records based on the following rules:

- The Category of an agreement record is populated with the category of one of the targets if all the targets of the agreement have the same category (that is, the **Service Level Category** field in the target record).
- The Category of an agreement record is populated with a value of **Service Level Agreement** if the targets of the agreement have different categories or have an empty category.
- The Phase of an agreement record is populated with “agreed” if the expiration date is later than the current date.
- The Phase of an agreement record is populated with “expired” if the expiration date is earlier than the current date.

For SLM data migration, the **Legacy SLA to new Agreement** migration setting is provided. See the following tables for its details.

Default settings

Source Table	
Source Table	sla
Target Table	sla
Filter Condition	true

Field Mappings

Source Field	Target Field
agreement.id	agreement.id
agreements	agreements
category	
contacts	contacts
current.phase	
customer	customer
description	description
display.category	display.category
expiration	expiration
external.support.groups	external.support.groups

Value Mappings

Target Value Mapping Field	Mapping Type	Condition	Target Value
category	jsCallback		$value=lib.MigrateSLA.migrateSLA(sourceTable['agreement.id'])
current.phase	fixedValue	expiration>tod()	agreed
current.phase	fixedValue	expiration<tod()	expired
display.category	jsCallback		$value=lib.MigrateSLA.migrateSLA(sourceTable['agreement.id'])

Request data migration

Request data migration involves several tables. Transactional data in the Request module are copied from the ocmq/ocmo/ocml file to the request/requestTask file. Because it is a data copy, you cannot use the batch update mode in the data migration tool.

Data Migration for the Request module includes the following items:

- Request
- Request Task
- Support Data for Request and Request Task
- Request Fulfillment Catalog
- Service Catalog Connector

Request

Legacy quote and order are migrated to PD request.

Migrate Quote to PD Request

Default Settings:

Source Table	ocmq
Target Table	request
Filter Condition	true

Field Mappings:

Target Field	Source Field
agreement.ids	
alert	alert
alert.names	alert.names
alert.status	alert.status
approval.status	approval.status
approved.groups	approved.groups
assigned.to	assigned.to
bill.to.code	bill.to.code
bill.to.dept	bill.to.dept
brief.description	brief.description

Target Field	Source Field
cancelled.date	cancelled.date
category	category
close.date	close.date
closure.code	completion.code
closure.comments	
company	company
coordinator.name	coordinator
current.pending.groups	current.pending.groups
current.phase	current.phase
delivery.date	requested.date
description	description
folder	folder
future.groups	future.groups
impact	impact
number	number
open	open
pending.groups	pending.groups
priority	priority

Target Field	Source Field
project.id	project.id
requested.for	requested.for
requestor.name	requestor.name
ship.to.code	ship.to.code
status	status
subcategory	subcategory
submit.date	submit.date
total.cost	total.cost
update.date	update.date

Value Mappings:

Target Value Mapping Field	Mapping Type	Condition	Target Value
agreement.ids	jsCallback	true	$value=[sourceTable['agreement.id']]
category	fixedValue	true	Generic Request
closure.comments	jsCallback	true	$value=sourceTable['comments'].toArray().toString()

Target Value Mapping Field	Mapping Type	Condition	Target Value
current.phase	fixedValue	(current.phase="Customer follow-up" or current.phase="HR Customer follow-up") and status~="closed"	Review
current.phase	fixedValue	current.phase="Initial Quote" or current.phase="MM Initial Quote"	Logging
current.phase	fixedValue	current.phase="MM Approval" or current.phase="Manager Approval" or current.phase="Quote Approval" or current.phase="tech/bus approval"	Authorization
current.phase	fixedValue	current.phase="MM Order 1" or current.phase="MM Order 2" or current.phase="Ordering" or current.phase="Working"	Fulfillment
current.phase	fixedValue	(current.phase="Customer follow-up" or current.phase="HR Customer follow-up") and status="closed"	Closure
current.phase	fixedValue	current.phase~="Customer follow-up" and current.phase~="HR Customer follow-up" and current.phase~="Initial Quote" and current.phase~="MM Initial Quote" and current.phase~="MM Approval" and current.phase~="Manager Approval" and current.phase~="Quote Approval" and current.phase~="tech/bus approval"	Fulfillment
status	fixedValue	(current.phase="Customer follow-up" or current.phase="HR Customer follow-up") and status~="closed"	Fulfilled
status	fixedValue	(current.phase="Customer follow-up" or current.phase="HR Customer follow-up") and status="closed"	Closed
status	fixedValue	current.phase="Initial Quote" or current.phase="MM Initial Quote" or current.phase="MM Approval" or current.phase="Manager Approval" or current.phase="Quote Approval" or current.phase="tech/bus approval"	Open
status	fixedValue	current.phase="MM Order 1" or current.phase="MM Order 2" or current.phase="Ordering" or current.phase="Working"	In Progress

Target Value Mapping Field	Mapping Type	Condition	Target Value
status	fixedValue	current.phase~="Customer follow-up" and current.phase~="HR Customer follow-up" and current.phase~="Initial Quote" and current.phase~="MM Initial Quote" and current.phase~="MM Approval" and current.phase~="Manager Approval" and current.phase~="Quote Approval" and current.phase~="tech/bus approval"	In Progress

Migrate Order to PD Request

Default Settings:

Source Table	ocmo
Target Table	request
Filter Condition	1 in description like "*reordered*"

Field Mappings:

Target Field	Source Field
alert	alert
alert.names	alert.names
approval.status	approval.status
bill.to.code	bill.to.code
bill.to.dept	bill.to.dept
category	category

Target Field	Source Field
close.date	close.date
closure.code	completion.code
closure.comments	
company	company
coordinator.name	coordinator
current.phase	current.phase
description	description
folder	folder
impact	impact
number	number
open	open
pending.groups	pending.groups
priority	priority
project.id	project.id
requestor.name	
ship.to.code	ship.to.code
status	status
subcategory	

Target Field	Source Field
submit.date	submit.date
total.cost	total
update.date	update.date

Value Mappings:

Target Value Mapping Field	Mapping Type	Condition	Target Value
category	fixedValue	true	Order
closure.comments	jsCallback	true	$value=sourceTable['comments'].toArray().toString()
current.phase	fixedValue	status~="closed"	Order
current.phase	fixedValue	status="closed"	Closure
requestor.name	fixedValue	true	scheduler
status	fixedValue	status~="closed"	Ordering
status	fixedValue	status="closed"	Closed
subcategory	fixedValue	true	Order

Request Task

Legacy quote line item and order line item are migrated to PD request task.

Migrate Quote Line Item to PD Request Task

Default Settings:

Source Table	ocml
Target Table	requestTask
Filter Condition	not null(parent.quote)

Field Mappings:

Target Field	Source Field
actual.end	delivered.date
actual.lead.time	target.lead.time
alert.names	alert.names
balance	quantity.balance
category	category
close.date	close.date
closure.code	completion.code
closure.comments	
company	company
current.phase	

Target Field	Source Field
description	description
folder	folder
number	number
open	open
ordered.quantity	quantity
parent.request	parent.quote
part.no	part.no
planned.end	target.completion
planned.lead.time	normal.lead.time
planned.start	target.order
received.quantity	quantity.received
ship.to.code	ship.to.code
status	status
submit.date	submit.date
total.cost	total
update.date	update.date
vendor	vendor

Value Mappings:

Target Value Mapping Field	Mapping Type	Condition	Target Value
category	fixedValue	trans.type="purchase" or trans.type="lease" or trans.type="rental"	Purchase
category	fixedValue	trans.type=null	Labor
category	fixedValue	trans.type~="purchase" and trans.type~="lease" and trans.type~="rental"	Labor
closure.comments	jsCallback	true	$value=sourceTable['comments'].toArray().toString()
current.phase	fixedValue	status="closed"	Closure
current.phase	fixedValue	status~="closed" and status~="error"	Active
current.phase	fixedValue	status="error"	Cancelled
status	fixedValue	status="closed"	Closed
status	fixedValue	status~="closed" and status~="error"	In Progress
status	fixedValue	status="error"	Cancelled

Migrate Order Line Item to PD Request Task

Default Settings:

Source Table	ocml
Target Table	requestTask
Filter Condition	not null(parent.order) and parent order is "reorder"

Field Mappings:

Target Field	Source Field
actual.end	delivered.date
actual.lead.time	actual.lead.time
actual.start	ordered.date
alert.names	alert.names
balance	quantity.balance
category	category
close.date	close.date
closure.code	completion.code
closure.comments	
company	company
current.phase	
description	description
folder	folder
number	number
open	open
ordered.quantity	quantity
parent.request	parent.order

Target Field	Source Field
part.no	part.no
received.quantity	quantity.received
ship.to.code	ship.to.code
status	status
submit.date	submit.date
total.cost	total
update.date	update.date
vendor	vendor

Value Mappings:

Target Value Mapping Field	Mapping Type	Condition	Target Value
category	fixedValue	trans.type="purchase" or trans.type="lease" or trans.type="rental"	Purchase
category	fixedValue	trans.type=null	Labor
category	fixedValue	trans.type~="purchase" and trans.type~="lease" and trans.type~="rental"	Labor
closure.comments	jsCallback	true	$value=sourceTable['comments'].toArray().toString()
current.phase	fixedValue	status="closed"	Closure

Target Value Mapping Field	Mapping Type	Condition	Target Value
current.phase	fixedValue	status="error"	Cancelled
current.phase	fixedValue	status~="closed" and status~="error"	Active
status	fixedValue	status="closed"	Closed
status	fixedValue	status="error"	Cancelled
status	fixedValue	status~="closed" and status~="error"	In Progress

Support Data for Request and Request Task

Support data migration include migrating the related approval and attachment data for request and request task.

Request and Request Task Related Approval Migration

"Approval" records migration

Default Settings:

Source Table	Approval
Target Table	Approval
Filter Condition	file.name="ocmq" or file.name="ocmo" or file.name="ocml"

Field Mappings:

Target Field	Source Field
file.name	file.name

Value Mappings:

Target Value Mapping Field	Mapping Type	Condition	Target Value
file.name	fixedValue	file.name="ocmq" or file.name="ocmo"	request
file.name	fixedValue	file.name="ocml"	requestTask

"ApprovalLog" records migration

Default Settings:

Source Table	ApprovalLog
Target Table	ApprovalLog
Filter Condition	file.name="ocmq" or file.name="ocmo" or file.name="ocml"

Field Mappings:

Target Field	Source Field
file.name	file.name

Value Mappings:

Target Value Mapping Field	Mapping Type	Condition	Target Value
file.name	fixedValue	file.name="ocmq" or file.name="ocmo"	request
file.name	fixedValue	file.name="ocml"	requestTask

Request and Request Task Related Attachment Migration

Default Settings:

Source Table	SYSATTACHMENTS
Target Table	SYSATTACHMENTS
Filter Condition	application="ocmq" or application="ocmo" or application="ocml"

Field Mappings:

Target Field	Source Field
application	application

Value Mappings:

Target Value Mapping Field	Mapping Type	Condition	Target Value
application	fixedValue	application="ocmq" or application="ocmo"	request
application	fixedValue	application="ocml"	requestTask

Request Fulfillment Catalog

Two out-of-box scripts are provided to help you migrate your legacy request fulfillment catalog data to the new PD request catalog structure. SM out-of-box provides a “PD Data Migration” menu record for the administrator to run these two out-of-box scripts.

Migrate Independent Fulfillment Catalog Items

An out-of-box script is provided to help migrate data for independent fulfillment catalog items. The data will be migrated from the “model” table to the “productCatalog” table.

In the “model” table, all catalog items without component will be migrated to the “productCatalog” table. Meanwhile, the master category of the “LI Category” of these catalog items will be migrated to the “prodCatalogCategory” table.

Migrate Packages

An out-of-box script is provided to help migrate data for packages (that is, bundle). The package data will be migrated from the “model” table to the “requestModel” table.

The required, optional, or default components in a legacy request package will be migrated to PD request tasks. In run time, the condition of PD request tasks can decide whether they are required, optional, or default. However, in static migration script, the condition of all PD request tasks is “true” by default.

After migration, open phase and close phase of PD request tasks in Request Model is set as a fixed value, “Fulfillment.”

In general, “Brief Description” of components in legacy request package is migrated as task title in PD task planner. Meanwhile, “Brief Description” and part number of legacy request package are migrated as the name of PD Request Model.

After migration, the category of PD Request Model is set to a fixed value, “Generic Request.” The subcategory of PD Request Model is set according to the following migration policy:

Legacy LI Category	PD Request Model Subcategory
Computer Accessories	Hardware
Computer Components	Hardware
Computers and Related	Hardware
Desktop	Hardware
Desktop Packages	Hardware
Desktops	Hardware
Handheld and PDAs	Hardware
Hardware	Hardware
Hardware Upgrade	Hardware
Monitor	Hardware
Non PC Computers	Hardware
Notebook	Hardware
Notebooks	Hardware
PC's and Related	Hardware
Printer	Hardware
Toner Products	Hardware
pc	Hardware
Employee Termination	Employee Off-boarding

Legacy LI Category	PD Request Model Subcategory
Termination	Employee Off-boarding
New Accounts	Request for Administration
User Accounts	Request for Administration
PC Apps	Software
PC Software	Software
Programming	Software
Software	Software
Software Applications	Software
Software Categories	Software
Software Group	Software
Software Installation	Software
Software License	Software
Software Purchase	Software
Software Upgrade	Software
UNIX Applications	Software
New Employee Setup	Employee On-boarding
Security	Request for Administration
Security Access	Request for Administration

Legacy LI Category	PD Request Model Subcategory
Other LI categories	Others

The following table describes the data migration policy for PD request task category:

Legacy LI Category	PD Request Task Category
Changes	Labor
Changes to Lines	
Common Office Environment	
Contractor Conversion	
Corporate	
Corporate Security	
Employee Change	
Employee Office Move	
Employee Promotion	
Employee Termination	
Employee Transfer	
HR Tasks	
Human Resources	
Installation	

Legacy LI Category	PD Request Task Category
Installation Services	
Move Equipment	
Network	
New Accounts	
New Employee Setup	
New Lines	
Office Move	
PC Software	
Payroll	
Payroll Tasks	
Programming	
Promotion	
Repair	
Repair Services	
Security	
Security Access	
Software Installation	
Software Upgrade	

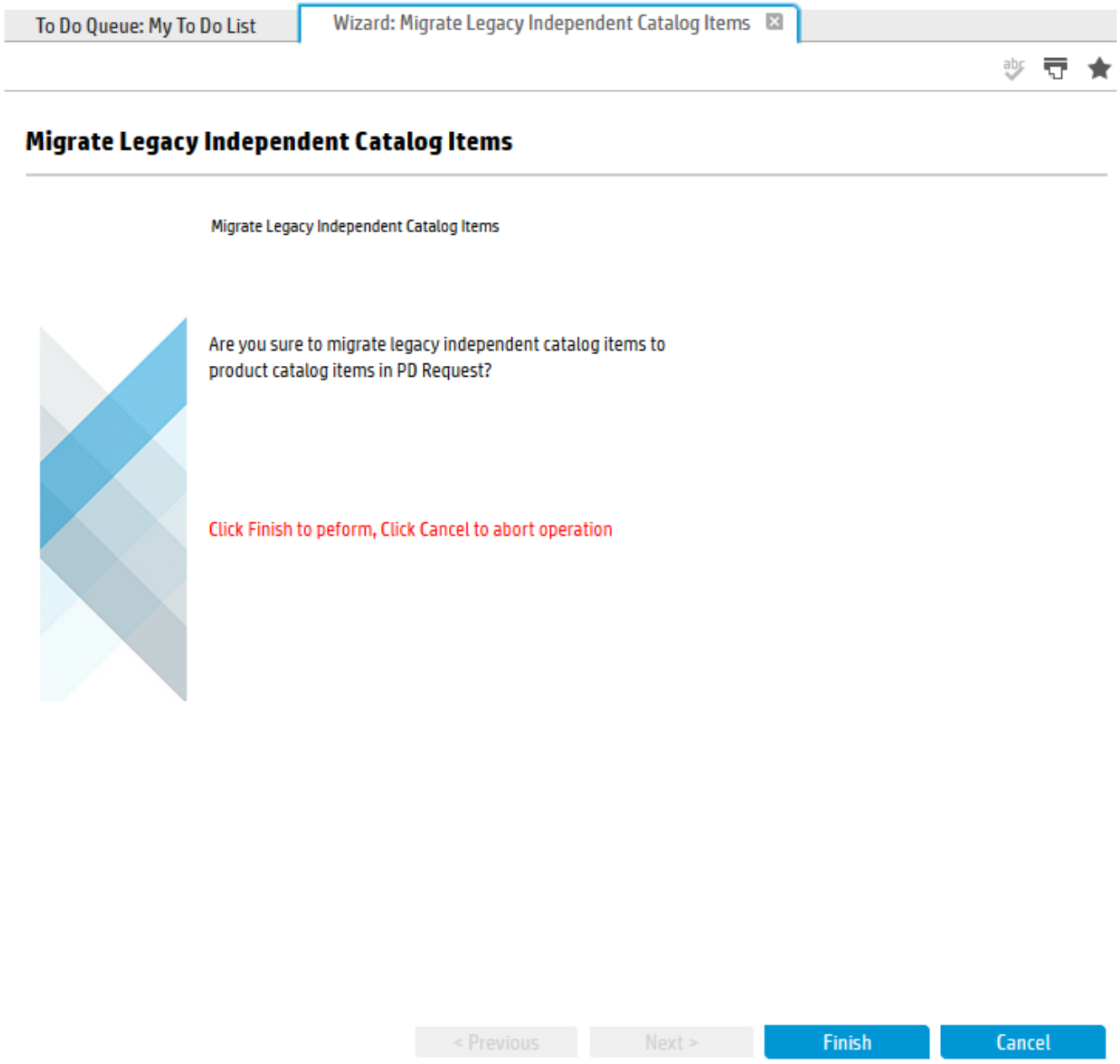
Legacy LI Category	PD Request Task Category
Telecom	
Termination	
Transfer	
UNIX Applications	
User Accounts	
Voice Requests	
Workorder	
labor	
phone	
training	
Other LI categories	Purchase

Menu of PD Data Migration

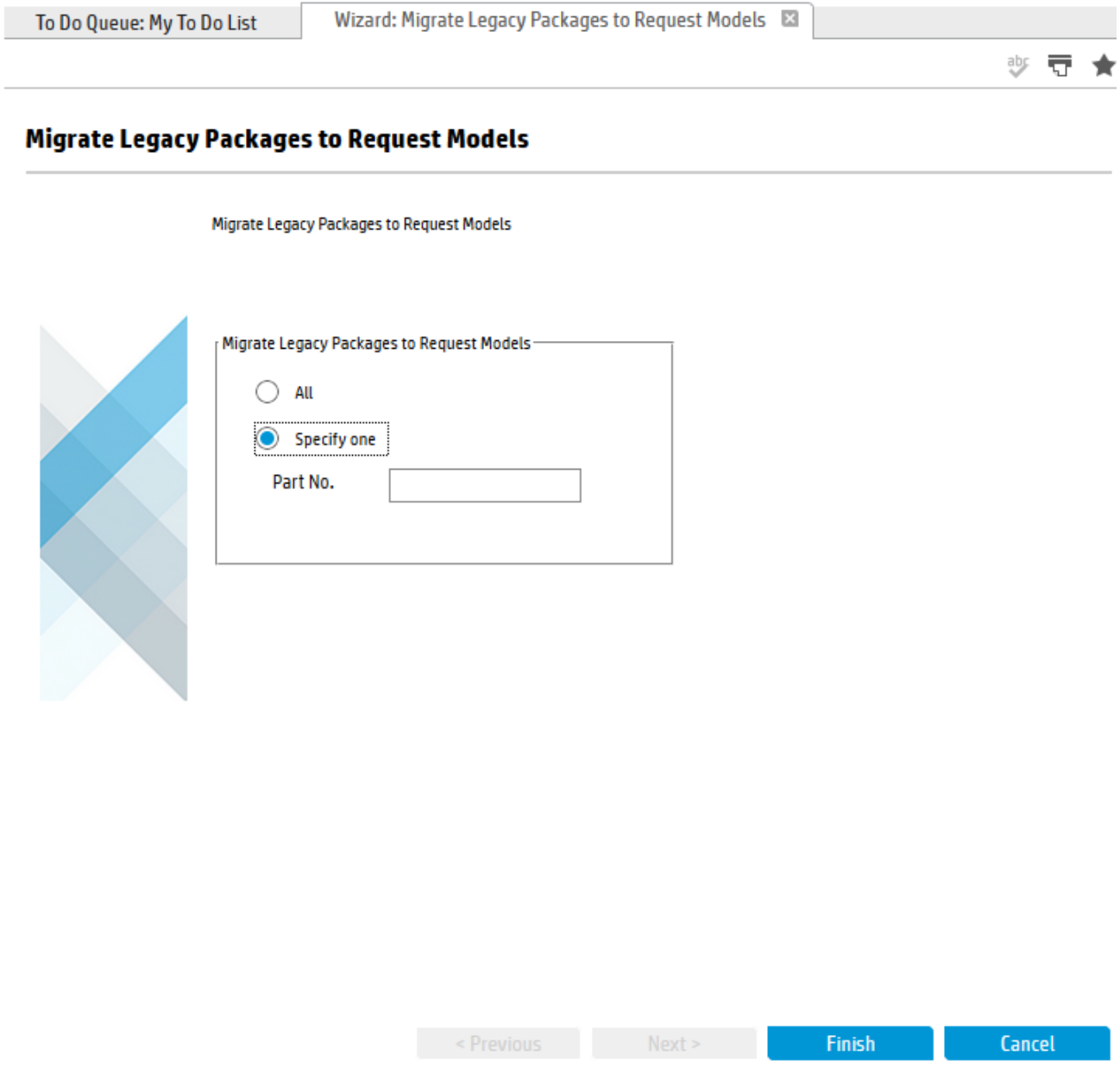
A separate “PD Data Migration” menu record is provided in the out-of-box system. If you want to migrate your legacy fulfillment catalog data with the out-of-box migration scripts, you can add the “PD Data Migration” menu record to the administrator menu. As a result, the administrator is able to run the data migration scripts from this menu item.

The following is the out-of-box “PD Data Migration” menu record:

When “Migrate Legacy Independent Catalog Items” is clicked, the following wizard form is displayed. Once you click **Finish**, the out-of-box migration script to migrate legacy independent catalog items will be executed.


To Do Queue: My To Do List
Wizard: Migrate Legacy Independent Catalog Items
Migrate Legacy Independent Catalog Items
Migrate Legacy Independent Catalog Items
Are you sure to migrate legacy independent catalog items to product catalog items in PD Request?
Click Finish to peform, Click Cancel to abort operation
< Previous
Next >
Finish
Cancel

When you click “Migrate Legacy Packages” from the PD Data Migration menu, the following wizard is displayed. The out-of-box migration script to migrate the legacy packages will be executed once you click **Finish**. You can either migrate all legacy packages at one time or just one specified package.


To Do Queue: My To Do List
Wizard: Migrate Legacy Packages to Request Models
Migrate Legacy Packages to Request Models
Migrate Legacy Packages to Request Models
Migrate Legacy Packages to Request Models
All
Specify one
Part No.
< Previous
Next >
Finish
Cancel

Service Catalog connector

If your system is upgraded to PD, the “Open New Request” and “Open a Change” connectors defined in the existing Service Catalog items need to be manually reconfigured, so that the fulfillment records can be created successfully in the PD environment.

Reconfigure the “Open New Request” connector

To reconfigure the “Open New Request” connector in the existing Service Catalog items, follow these steps:

1. Navigate to **Service Catalog** > **Administration** > **Manage Catalog**.
2. Search for the Service Catalog items that use the “Open New Request” connector:
    a. Select **Item** in the **Type** drop-down list.
    b. Select **Open New Request** in the **Interface Type** drop-down list.
    c. Click **Search**. All the Service Catalog items using the “Open New Request” connector are displayed.
3. In each Service Catalog item definition, go to the **Connector Details** tab, and then click **Add/Edit Information**.
4. Enter information for the following fields, and the click **Next**.
    - Request Category (mandatory)
    - Request SubCategory (optional)

- Department (optional)
- Request Model (optional)

5. Enter information for the following fields, and the click **Finish**.
   - Urgency (mandatory)
   - Impact (mandatory)
   - Assignment (optional)
6. Save the Service Catalog item.
7. Repeat step 3 to step 6 to reconfigure the “Open New Request” connector information for all the Service Catalog items.

Reconfigure the “Open a Change” connector

To reconfigure the “Open a Change” connector in the existing Service Catalog items, follow these steps:

1. Navigate to **Service Catalog** > **Administration** > **Manage Catalog**.
2. Search for the Service Catalog items that use the “Open a Change” connector:
   a. Select **Item** from the **Type** drop-down list.
   b. Select **Open a Change** from the **Interface Type** drop-down list.
   c. Click **Search**. All the Service Catalog items using the “Open a Change” connector are displayed.
3. In each Service Catalog item definition, go to the **Connector Details** tab, and then click **Add/Edit Information**.

4. Enter information for the following fields, and the click **Next**.
   - Change Category (mandatory)
   - Change SubCategory (optional)
   - Change Model (optional)
5. Enter information for the following mandatory fields, and the click **Finish**.
   - Impact Assessment
   - Urgency
   - Assignment
   - Service
   - Requested End Date
   - Reason for Change
6. Save the Service Catalog item.
7. Repeat step 3 to step 6 to reconfigure the “Open a Change” connector information for all the Service Catalog items.

Re-index the knowledge bases

After you remove the Problem_Library_disabled_by_PDHD and KnownError_Library_disabled_by_PDHD knowledgebases, you must run a full re-index of the knowledgebases if you use Knowledge Management in a production environment.

Appendix A

Changes made by the Process Designer Enablement script

This appendix describes the security changes and other changes that will be made to the Help Desk, Change, and Request modules by the Process Designer Enablement script.

Security changes

The legacy Service Manager security profile settings include module profiles, capability words, security folders, Mandanten, query group, and user role. The following table lists what is migrated or unchanged after you enable Process Designer.

SM Security Setting	What is changed in Process Designer based modules?
Module profile	The following module profiles are migrated to Process Designer security roles and security rights. ▪ Change ▪ Service Desk ▪ Incident ▪ Problem/Known Error ▪ Request
Capability word	No changes
Security folder	No changes.
User Role	No changes to user roles. However, the previous associations that existed between User Role and profile now exist between User Role and Process Designer Security Role.

SM Security Setting	What is changed in Process Designer based modules?
Operator	No changes to the Operator. However, the previous association between Operator and profile now exists between Operator and Process Designer Security Role.
Query Group	No changes.

The Process Designer security framework is based on the security role, security area and security rights concepts. A user's specific security rights (such as view, update, delete, allowed category etc.) are defined by applying a set of security roles to the set of security areas. This concept is important to understand the mapping rules used in the security migration script. For more information on the details of Process Designer security concept, refer to the Service Manager 9.40 online help.

During the migration, each original Service Manager security profile is mapped to a combination of security roles and security rights according to the following rules:

- For modules other than Request, a security role with the same name of the profile name is created (in the secRole file). For Request, a suffix is added to each profile name as a security role name and the Process Designer enablement script assigns the security role to operators using that profile.
- The settings on the original Service Manager security profile are moved to the security rights (in secRights file) that are defined for the security area that corresponds to the profile. The exact migration path is also based on a mapping table described in the Security section of each module in the Service Manager Online Help. Some very rarely used setting are not mapped.

For example, after the “Incident Analyst” security profile in Incident Management is migrated, a security role also named “Incident Analyst” is created. Then, the profile setting is moved to the security rights record that is defined for the “Incident” security area. The Incident security area maps the Incident Module to which “Incident Analyst” belongs.

The following table specifies the mapping rules from an original Service Manager security profile to Process Designer security roles and security rights.

Module	Source Profile Files	Map to Target Process Designer secRole and secRights

Change	cm3profile	A security role of the same name is created for each profile record (including customized profiles). The profile setting is migrated to security rights as follows: • If Profile Area = Tasks, the security rights are migrated to the Change Tasks area. • If Profile Area = changes, the security rights are migrated to the Change area. • If Profile Area = all, the security rights are migrated to the Change and Change Tasks areas.
Service Desk	smenv	A security role of the same name is created for each profile record (including customized profiles). The profile setting is migrated to the security rights that are defined for the Service Desk security area.
Incident	pmenv	A security role of the same name is created for each profile record (including customized profiles). The profile setting is migrated to the security rights that are defined for the Incident security area.
Problem	rcenv	A security role of the same name is created for each profile record (including customized profiles). The profile setting is migrated to the security rights that are defined for the Problem security area.
Request	ocmprofile	A security role of the same name plus a suffix is created for each profile record (including customized profiles). The profile setting is migrated to security rights as follows: • If Profile Area = Quotes, the security rights are migrated to the Request area. • If Profile Area = Orders, the security rights are migrated to the Request area. • If Profile Area = Line Items, the security rights are migrated to the Request Task area. • If Profile Area = All, the security rights are migrated to the Request and Request Tasks areas.

After Process Designer is enabled, the profile settings are simplified into Process Designer security rights. However, the legacy profile variables are still populated during user login to maintain compatibility and to avoid changes to the runtime application code that still uses the legacy variables.

The following legacy profile variables of the following five modules are populated from the security rights (not from the profile) when a user logs in:

- Change Management: $G.cm3r.environment, and $G.cm3t.environment
- Incident Management: $G.pm.environment
- Problem Management: $G.rc.environment
- Service Desk: $G.sm.environment
- Knowledge Management: $G.km.environment
- Request Management: $G.ocmq.environment, and $G.ocml.environment

Unmapped Legacy Profile Settings

When you migrate the profile settings to Process Designer security rights, not all the legacy profile settings are migrated to security rights. For more information about the unmapped legacy profile settings, see the Service Manager 9.40 online help.

Changes for PD Help Desk

This section describes the changes that will be made to the Help Desk modules by the Process Designer Enablement script.

Object changes for PD Help Desk

The Process Designer Enablement script changes the following Help Desk Objects by appending "_ legacy_disabled_by_PD_HelpDesk" to the original file name and renaming the temporary new Objects from <file name>_for_pd4_tobe_used to just <file name>.

Object File Name Before Enablement	Object File Name After Enablement	Description
probsummary	probsummary_legacy_disabled_by_ PD_HelpDesk	The Legacy probsummary Object.

probsummary_for_pd4_tobe_used	probsummary	The Process Designer probsummary Object.
incidents	incidents_legacy_disabled_by_PD_HelpDesk	The Legacy incidents Object.
incidents_for_pd4_tobe_used	incidents	The Process Designerincidents Object.
rootcause	rootcause_legacy_disabled_by_PD_HelpDesk	The Legacy rootcause Object.
rootcause_for_pd4_tobe_used	rootcause	The Process Designerrootcause Object.
rootcausetask	rootcausetask_legacy_disabled_by_PD_HelpDesk	The Legacy rootcause task Object.
rootcausetask_for_pd4_tobe_used	rootcausetask	Process DesignerPD rootcause task Object.

Datadict changes for PD Help Desk

The Process Designer Enablement script changes the following Datadict entries to update the field captions and usage types:

Note:

- Captions are the same as the field labels in forms, and the same as the mandatory and validation check messages.
- The usage type is used by the template feature and copy record feature and also used in the Process Designer workflow/ruleset condition editor. System-type fields will not be in the field list by default.

probsummary:

Field Name	New Caption	New Usage Type
category	Category	System
subcategory	Subcategory	System
product.type	Area	System

current.phase	Phase	System
problem.status	Status	System

rootcause:

Field Name	New Caption	New Usage Type
category	Category	System
subcategory	Subcategory	System
product.type	Area	System
current.phase	Phase	System
rcStatus	Status	System

rootcausetask:

Field Name	New Caption	New Usage Type
id	Task ID	System
incident.category	Parent Category	System
category	Category	System
task.category	Task Category	System
subcategory	Subcategory	System
product.type	Area	System
rcStatus	Status	System
current.phase	Phase	System
parent.problem	Parent Problem	System

incidents:

Field Name	New Caption	New Usage Type
category	Category	System
subcategory	Subcategory	System
product.type	Area	System

current.phase	Phase	System
open	Status	System

cm3r & cm3t

The Process Designer Enablement script modifies datadict of "cm3rcatphase" and "cm3tcatphase" by changing the Default Format from "cm3rcatphase.main" and "cm3tcphs.main" to "chm.cm3rcatphase.main.g" and "chm.cm3tcphs.main.g", which will make the "Additional Phase Information" function work correctly after enabling PD Change.

Menu changes for PD Help Desk

Process Designer Help Desk uses new menu items to access the Help Desk modules. The Enablement script renames the legacy Help Desk menu item descriptions and adds related new Process Designer Help Desk menus.

Top Level Menu Name
ADMIN
HELPDESK GEN
HOME
ICM GEN
IM GEN
PM GEN

In the out-of-box Service Manager system, there are nine top level menus. Three other top level menus (APPROVER GEN, CM GEN and CM Main) do not need to be updated because they do not include Help Desk menu items.

The following table shows the changes to the legacy Help Desk menu items under the top level menus:

Legacy Menu Item Name	Legacy Menu Item Description before Enablement	Renamed Legacy Menu Item Description after Enablement
IM	Incident Management	Legacy Incident Management
PM	Problem Management	Legacy Problem Management
CC	Service Desk	Legacy Service Desk

Note: The renamed Legacy Help Desk menu items are no longer used. You can remove them manually after you finish migrating to Process Designer Help Desk. In addition to the six top level menus, if you have customized Service Manager to have more top level menus that use the legacy Help Desk menu items, you must also manually add the new Process Designer Help Desk menu items into those top level menus.

The following table shows the Process Designer Help Desk menu items that are added to the top level menus:

Menu Item Name	Menu Item Description
IM Process Designer	Incident Management
PM Process Designer	Problem Management
CC Process Designer	Service Desk

The following menu items are removed from the "ENV RECORDS" menu record (located under **System Administration** > **Ongoing Maintenance** > **Environment Records**):

ENV menu item
Incident Management Environment
Service Desk Environment
Problem Management Environment

SearchConfig

Process Designer Help Desk uses a new search form, so the Enablement script changes the SearchConfig value for the search forms as shown in the following table:

File Name	Legacy Search Form Before Enablement	New Search Form After Enablement
probsummary	advFind.incident.search	im.advFind.incident.search
rootcause	advFind.search.problem	pbm.advFind.search.problem
rootcausetask	advFind.search.problem.task	pbm.advFind.search.problem.task
incidents	advFind.SD.search	sd.advFind.search

In addition, the following search configurations have "and false" appended to the new "allowAdvAccess" field. This is because knownerror and knownerrortask are removed from Service Manager:

File Name	Legacy Allow Advanced Find	New Allow Advanced Find
knownerror	kne.browse in $G.rc.environment	kne.browse in $G.rc.environment and false
knownerrortask	ket.browse in $G.rc.environment	ket.browse in $G.rc.environment and false

Queue displays

The Mass Close button of the Incident queue is no longer available for Process Designer Help Desk. Therefore, the Enablement script replaces the scm.advanced_massclose display option with the “false” condition.

Display option id Before Enablement	Display option id After Enablement	Description
scm.advanced_ massclose	scm.advanced_massclose_ legacy_disabled_by_PD_ HelpDesk	The Legacy display option to “mass close” incidents in queue
scm.advanced_ massclose_for_pd4_ tobe_used	scm.advanced_massclose	The Process Designer display option to “mass close” incidents in queue

Related record links

Links

The Enablement script changes the Link lines of the screlate.get.association link by renaming the legacy link line to _legacy_disabled_by_PD_HelpDesk and the temporary new link lines from _for_pd4_tobe_ used as shown in the following table:

Link line name	Description
rootcause	The new search format is changed from screlate.search.rc.g to screlate.search.rc.pd.g.
problem	The new search format is changed from screlate.search.problem.g to screlate.search.problem.pd.g.
incidents	The new search format is changed from screlate.search.incident.g to screlate.search.incident.pd.g.

Global List

To support the new link type, the Enablement script appends"Related Records Type Change List" to global list as follows:

List value	Display value	Message Value
Caused Problems	Caused Problems	Caused Problems
Solved Problems	Solved Problems	Solved Problems

In addition, the Enablement script removes the entry from the "Related Records Type Change List" global list:

List value	Display value	Message Value
Related Known Errors	Related Known Errors	Related Known Errors

Limitation

The Enablement script cannot migrate legacy relationship types to the types that are introduced in Process Designer Content Pack (PDCP) 9.30.3 and in out-of-box deployments of Service Manager 9.40 Codeless. Therefore, if your environment contains any records that were related before PDCP 9.30.3 was applied or before Service Manager 9.40 was installed, some feature (such as the Run Actions rule) that depend on these relationship types will not work correctly for these older related records.

KM related records

Format changes

Process Designer Help Desk uses the following new formats for advanced Knowledge Management search:

- kmknowledgebase.advsearch.g
- kmknowledgebase.advsearch.pd.g
- PM.error.km
- PM.error.km.pd

- PM.problem.km
- PM.problem.km.pd

Therefore, the following records are changed using the format names as shown in the following table:

Type	name	Update Field	Substr	Replacement
Process	kmquery.advancedoptions	pre.expressions	"kmknowledgebase.advsearch.g"	"kmknowledgebase.advsearch.pd.g"
displayoption	kmknowledgebase.search.view_fill	condition	"kmknowledgebase.advsearch.g"	"kmknowledgebase.advsearch.pd.g"
displayoption	kmknowledgebase.search.view_find	condition	"kmknowledgebase.advsearch.g"	"kmknowledgebase.advsearch.pd.g"
displayoption	kmquery.default_newsearch_1	condition	"kmknowledgebase.advsearch.g"	"kmknowledgebase.advsearch.pd.g"
Process	kmquery.linkrequest	javascript.pre	"PM.error.km" "PM.problem.km"	"PM.error.km.pd" "PM.problem.km.pd"

Links

The Enablement script changes the Link lines of the kmquery link by renaming the legacy link to <link line name>_legacy_disabled_by_PD_HelpDesk and the temporary new link line from <link line name>_for_pd4_tobe_used to the link name shown in the following table.

Link line name	Description
incidentlib.category	The target table is changed from category to imCategory.
incidentlib.subcategory	The target table is changed from subcategory to imSubcategory.
incidentlib.producttype	The target table is changed from producttype to imArea
interactionlib.category	The target table is changed from category to sdCategory.
interactionlib.subcategory	The target table is changed from subcategory to sdSubcategory.
interactionlib.producttype	The target table is changed from producttype to sdArea.

Link line name	Description
knownerrorlib.category	The target table is changed from category to pbmCategory.
knownerrorlib.subcategory	The target table is changed from subcategory to pbmSubcategory.
knownerrorlib.producttype	The target table is changed from producttype to pbmArea
problemlib.category	The target table is changed from category to pbmCategory.
problemlib.subcategory	The target table is changed from subcategory to pbmSubcategory.
problemlib.producttype	The target table is changed from producttype to pbmArea

Knowledge Management knowledge base

kmknowledgebase **Before Enablement**	**kmknowledgebase** **After Enablement**	**Description**
KnownError_Library	KnownError_Library_disabled_by_PDHD	The Legacy kmknowledgebase.
KnownError_Library_for_pd4_tobe_used	KnownError_Library	The Process Designer kmknowledgebase differs from the legacy kmknowledgebase as follows: • The table name is changed from knownerror to rootcause; • The "isKnownError=true" scquery is removed.
Problem_Library	Problem_Library_disabled_by_PDHD	The Legacy kmknowledgebase.
Problem_Library_for_pd4_tobe_used	Problem_Library	The Process Designer kmknowledgebase differs from the legacy kmknowledgebase as follows: • The "isKnownError=NULL or isKnownError=false"; scquery is removed • The Category field name is changed from incident.category to category. • The sysmodtime field is removed.

Inbox

Enablement script makes the following changes to the view inbox records:

View	File	Query in Legacy System	How to modify
My Group's To Do List	Todo	((itemType="probsummary" or itemType="incidents" or itemType="rootcause" or itemType="rootcausetask" or itemType="knownerror" or itemType="cm3r" or itemType="cm3t") and group isin $lo.pm.assignments) or ((itemType="ocmq" or itemType="ocml") and group=$lo.dept.corp.structure)	Change the Query definition as follows: ((itemType="probsummary" or itemType="incidents" or itemType="rootcause" or itemType="rootcausetask" or itemType="imTask" or itemType="cm3r" or itemType="cm3t") and group isin $lo.pm.assignments) or ((itemType="ocmq" or itemType="ocml") and group=$lo.dept.corp.structure)
Rejected Incidents	probsummary	(flag#true and problem.status#"Rejected")	Change the Query as follows: flag#true and resolution.code="Rejected" and problem.status="Closed"
All Open - Callback Interactions	incidents	open="Open - Callback"	Change the Query as follows: open="Callback"
All Open Interactions	incidents	open#"Open"	Change the Query as follows: open~="Closed"

Self-Service Interactions	incidents	ess.entry=true and open="Open - Idle"	Change the Query as follows: ess.entry=true and (open="Open" or open="Categorize" or open = "Assign" or open = "In Progress")
All Open Problems	rootcause	open#true and (isKnownError=NULL or isKnownError=false)	Change the Query as follows: open#true and (isKnownError=NULL or isKnownError=false)
High Priority Problems	rootcause	open#true and priority.code#"1" and (isKnownError=NULL or isKnownError=false)	Change the Query as follows: open#true and priority.code#"1" and (isKnownError=NULL or isKnownError=false)
Open Problems Assigned to Me	rootcause	open#true and assignee.name=operator() and (isKnownError=NULL or isKnownError=false)	Change the Query as follows: open#true and assignee.name=operator() and (isKnownError=NULL or isKnownError=false)
Open Problems Assigned to My Group	rootcause	open#true and assignment isin $lo.pm.assignments and (isKnownError=NULL or isKnownError=false)	Change the Query as follows: open#true and assignment isin $lo.pm.assignments and (isKnownError=NULL or isKnownError=false)
Problem Tickets Owned by Me	rootcause	open#true and ticket.owner=operator() and (isKnownError=NULL or isKnownError=false)	Change the Query as follows: open#true and ticket.owner=operator() and (isKnownError=NULL or isKnownError=false)

Problems in 'Problem Prioritization and Planning' Phase	rootcause	open#true and current.phase="Problem Prioritization and Planning" and (isKnownError=NULL or isKnownError=false)	Change the view to "Problem in Categorization Phase" and change the query as follows: open#true and current.phase="Categorization" and (isKnownError=NULL or isKnownError=false)
Problems Not Assigned in 'Problem Investigation and Diagnosis' Phase	rootcause	open#true and current.phase="Problem Investigation and Diagnosis" and null(assignee.name) and (isKnownError=NULL or isKnownError=false)	Change the view to "Problem in Investigation Phase" and change the query as follows: open#true and current.phase="Investigation" and (isKnownError=NULL or isKnownError=false)
All Open Known Errors	knownerror	open#true	Change the file to rootcause, and change the query as follows: open#true and isKnownError=true
Known Errors Not Assigned in 'Known Error Investigation' Phase	knownerror	open#true and current.phase="Known Error Investigation" and null(assignee.name)	Change the file to rootcause and the view to "Known Error not assigned in investigation phase." Then, Change the query as follows: open#true and current.phase="Investigation" and null(assignee.name) and isKnownError=true
Known Errors Where Solution Time Is Not Met	knownerror	open#true and solutionDate	Change the file to rootcause and change the query as follows: open#true and solutionDate

Open Known Errors Assigned to Me	knownerror	open#true and assignee.name=operator()	Change the file to rootcause and change the query as follows: open#true and assignee.name=operator() and isKnownError=true
Open Known Errors Assigned to My Group	knownerror	open#true and assignment isin $lo.pm.assignments	Change the file to rootcause and change the query as follows: open#true and assignment isin $lo.pm.assignments and isKnownError=true
All Open Incident Tasks	imTask	open#true	Add
All Open Known Error Tasks	knownerrortask	open#true	Remove
All Open Incident Tasks Assigned to Me	imTask	open#true and assignee.name=operator()	Add
All Open Known Error Tasks Assigned to Me	knownerrortask	open#true and assignee.name=operator()	Remove

Service Level Agreements

To support Incident tasks in the Service Level Agreement (SLA) module, the Enablement script appends the following list and display values to the "SLO Tables" global list:

List value	Display value
imTask	Incident Tasks

Service Desk

SD links

svcCat.wizard.support.triplet and joinsvcDisplay

The Enablement script changes the link lines of the svcCat.wizard.support.triplet and joinsvcDisplay links by renaming the legacy link lines to _ legacy_disabled_by_ProcessDesigner_HelpDesk and the temporary new _for_pd4_tobe_used link lines as shown in the following table:

Link line name	Description
category	The target file name is changed from category to sdCategory.
subcategory	The target file name is changed from subcategory to sdSubcategory.
product.type	The target file name is changed from producttype to sdArea.

svcCatInterface

The Enablement script updates the "Open an Incident" svcCatInterface to change the process.name to "im.first.PD".

screlconfig

The RAD to open a new Incident is changed to "document.new." Therefore, the Enablement script updates the "problem" screlconfig to change the open.application.

screlconfig Before Enablement	screlconfig After Enablement
problem	problem_disabled_by_PD_HelpDesk
problem_for_pd4_tobe_used	problem

Validity

The filename / files / field.name before and after Enablement:

Before Enablement	After Enablement
svcCat / joinsvcDisplay / category	svcCat_disabled / joinsvcDisplay_disable / category
svcCat_for_pd4 / joinsvcDisplay_for_pd4 / category	svcCat / joinsvcDisplay / category
svcCat / joinsvcDisplay / subcategory	svcCat_disabled / joinsvcDisplay_disable / subcategory
svcCat_for_pd4 / joinsvcDisplay_for_pd4 / subcategory	svcCat / joinsvcDisplay / subcategory
svcCat / joinsvcDisplay / product.type	svcCat_disabled / joinsvcDisplay_disable / product.type
svcCat_for_pd4 / joinsvcDisplay_for_pd4 / product.type	svcCat / joinsvcDisplay / product.type

activityaction

The Enablement script makes the following changes to the activityactions of the Problem Management module (file incidents):

Before Enablement	After Enablement	Description
Save Activities	Save Activities_disabled_by_PD_HelpDesk	Updates the condition field to: "false"
Save Activities_for_pd4_tobe_used	Save Activities	Updates the condition from "false" to: $L.mode="save" or $L.mode="update"
Update Activities	Update Activities	Updates the condition field to: "false"

Category related link

Process Designer Help Desk uses different tables to store the category/subcategory/area information. Therefore, the Enablement script also updates the following links:

Link Name
probsummary
rootcause
rootcausetask
incidents
svcCat.wizard.support.triplet
joinsvcDisplay

Specifically, the Enablement script changes the following link lines:

Link line name	Description
category	For probsummary, the target table is changed from category to imCategory. For rootcause, the target table is changed from category to pbmCategory. For rootcausetask, the target table is changed from rootcausetaskcat to pbmTaskCat. For incidents, the target table is changed from category to sdCategory.
subcategory	For probsummary, the target table is changed from subcategory to imSubcategory. For rootcause, the target table is changed from subcategory to pbmSubcategory. For incidents, the target table is changed from subcategory to sdSubcategory.
product.type	For probsummary, the target table is changed from producttype to imArea.For rootcause, the target table is changed from producttype to pbmArea. For incidents, the target table is changed from producttype to sdArea.

ModuleStatus

The Enablement script also appends the new status used by Process Designer Help Desk by adding the following ModuleStatus records:

Module	Status	Sort Order
incidents	Categorize	8
incidents	Assign	9
incidents	Dispatched	10
incidents	Callback	11

incidents	Suspended	12
incidents	In Progress	13
incidents	Withdrawal Requested	14
rootcause	Abandoned	12
rootcause	Assign	4
rootcause	Categorize	3
rootcause	Closed	11
rootcause	Pending	6
rootcause	Resolved	10
rootcause	Rejected	13
rootcausetask	Pending	7
rootcausetask	Closed	12
rootcausetask	Pending Review	10
rootcausetask	Planned	2
rootcausetask	Ready	4
probsummary	Categorize	1
probsummary	Assign	2
probsummary	Pending Evidence	4

In addition, the following legacy status is updated:

Module	Status	New Status	Description
probsummary	Work in Progress	Work In Progress	Capitalized the "i" in the word "In."

AlertDef

The Enablement script changes the Alert Definitions that are used by the Problem Management module as shown in the following table:

Alert Name Before Enablement	Alert Name After Enablement	Description
PM Past Due	PM Past Due_disabled_by_PD_ HelpDesk	The Legacy AlertDef

PM Past Due_for_pd4_tobe_ used	PM Past Due	The Process Designer Alert Definitions changes the statements as follows:
		
		From:
		if (rcStatus in $L.file~="Closed") then (rcStatus in $L.file="Past Due")
		To:
		if (rcStatus in $L.file~="Closed") then (description in $L.file=description in $L.file+
		
		Unknown macro: {"**The expected resolution time has been reached**"})

PMT Past Due	PMT Past Due_disabled_by_PD_ HelpDesk	The Legacy Alert Definition

PMT Past Due_for_pd4_tobe_ used	PMT Past Due	The Process Designer Alert Definitions changes the statements as follows:
		
		From:
		if (rcStatus in $L.file~="Closed") then (rcStatus in $L.file="Past Due")
		To:
		if (rcStatus in $L.file~="Closed") then (description in $L.file=description in $L.file+

contextAction

The Enablement script also changes the contextAction that are used by the Problem Management module as follows:

Name Before Enablement	Name After Enablement	Description
Possible workarounds for this Service	Possible workarounds for this Service_disabled_by_PD_ HelpDesk	The Legacy contextAction that is using the table name "knownerror"
Possible workarounds for this Service_for_pd4_tobe_ used	Possible workarounds for this Service	The Process Designer contextAction that changes the table name from "knownerror" to "rootcause", and also changes the SQL string in field code of this record.

extaccess

The Enablement script removes the following actions, which are no longer supported in Service Manager:

Name of extaccess	Actions to be removed
probsummary	mobileclose reopensave mobileresolve mobilesave resolvesave
rootcause	reopen

Changes for PD Change

This section describes the changes that will be made to the Change module by the Process Designer Enablement script.

Object changes for PD Change

The Process Designer Enablement script changes the following Change Management Objects by appending "_disabled_by_PDHD" to the original file name and rename the temporary new Objects from "_for_pd4_tobe_used to" to just <file name>.

Object File Name before Enablement	Object File Name after Enablement	Description
cm3r	cm3r_disabled_by_PDHD	The Legacy cm3r Object.
cm3r_for_pd4_tobe_used	cm3r	The PD cm3r Object.
cm3t	cm3t_disabled_by_PDHD	The Legacy cm3t Object.
cm3t_for_pd4_tobe_used	cm3t	The PD cm3t Object.

Menu changes for PD Change

PD Change uses new menu items to access the Change module, the Enablement script will change the following two top level menus to rename the legacy Change menu item descriptions and add related new PD Change menus.

Top Level Menu Name
HOME

Changes to top level menus

The following table describes the changes of legacy Change menu items under the two top level menus.

Legacy Change Menu Item Name	Legacy Change Menu Item Description before Enablement	Renamed Legacy Change Menu Item Description after Enablement	Note
ChM	Change Management	Legacy Change Management	• The renamed Legacy Change menu items actually will not be used anymore. Therefore, you can remove them manually after you finish your PD Change migration work. • Besides the two top level menus listed above, if you have more top level menus that use the legacy Change menu item listed above, you need to manually add the following new PD Change menu item to those top level menus. **PD Change Menu Item Name** / **PD Change Menu Item Description** ChM PD / Change Management

Additional menu changes

The following table lists additional menu changes for PD Change.

Menu Name	Before Enablement	After Enablement
CM GEN	Adds SM9.40 items to the SM9.33 version.	• Modifies "Change Management" to "Legacy Change Management" • Adds "Change Management"
CM SETUP	Uses the SM9.33 version.	Renames it to "CM SETUP LEGACY"
ChM	• Based on the SM9.33 version, adds SM9.40 menus of Calendar Administration and Time Period Management to 933 ChM. • Renames the 940 version to ChM PD	Uses ChM PD
ChM MAIN MENU	Use the SM9.33 version	Use the SM9.33 version
ChM MAINT MENU	Use the SM9.33 version	Use the SM9.33 version

Removal of the Change Management Profiles menu item

The "Change Management Profiles" menu item is removed from menu record "MODULE PROFILES" (under **System Administration** > **Ongoing Maintenance** > **Profiles**).

Datadict changes for PD Change

The Process Designer Enablement script modifies datadict records of "cm3rcatphase" and "cm3tcatphase" by changing the Default Format from "cm3rcatphase.main" and "cm3tcphs.main" to "chm.cm3rcatphase.main.g" and "chm.cm3tcphs.main.g" respectively, so as to make the "Additional Phase Information" function work correctly after the PD Change Management functionality is enabled.

Process changes for PD Change

The Process Designer (PD) Enablement script will change the following Processes for PD Change by appending " to _disabled_by_PD_HD" to the legacy file names and rename the temporary new Processes from "_for_pd4_tobe_used" to just <file name>. The following table lists the Process changes.

Process File Name before Enablement	Process File Name after Enablement	Description
cm.open	cm.open_disabled_by_PDHD	The Legacy cm.open Process
cm.open.save	cm.open.save_disabled_by_PDHD	The Legacy cm.open.save Process
cm.open.save_for_pd4_tobe_used	cm.open.save	The PD cm.open.save Process
cm.open_for_pd4_tobe_used	cm.open	The PD cm.open Process
cm.update	cm.update_disabled_by_PDHD	The Legacy cm.update Process
cm.update_for_pd4_tobe_used	cm.update	The PD cm.update Process

extaccess

The Process Designer Enablement script modifies the following extaccess definitions to enable PD Change.

Name	Changes made by the Enablement Script
Change	Adds SM9.40 contents for "Allowed Actions" and "Fields Mapping"
ChangeIIA	Adds SM9.40 contents for "Allowed Actions"
ChangeRC	Adds SM9.40 contents for "Allowed Actions," "Fields Mapping," and "Expressions"
ChangeTask	Adds SM9.40 contents for "Allowed Actions"
ChangeTaskRC	Adds SM9.40 contents for "Allowed Actions" and "Expressions"

Template

The Process Designer Enablement script removes the "priority.code" and "subcategory" fields from the following template records to enable PD Change.

Template Name
Activate patch
Add user account for new employee
Change user rights
Create database instance
Install Application on server
Install Database on server
Install Email software on Server

Template Name
Install Operating System
Install Windows Server
Install memory on server/PC
Install router
Memory Upgrade
Monthly printer maintenance
Monthly server maintenance
Move Personal Computer
New laptop request
New printer
New wireless accesspoint
Office installation / Upgrade
Replace Toner/Cartridge
SAP configuration
Update anti-virus software
Upgrade Server

Data cleanup

The Process Designer Enablement script modifies the following records to enable PD Change.

Table	Action	Notes
cm3tcatphase (PD change task phases)	Adds the following phases: • Abandoned • Active • Canceled • Closed • Review • Waiting	If the phases already exist in the target environment, backs them up with a SUFFIX of "_bak_by_PD," and then inserts the PD phases; otherwise, directly inserts the PD phases.

Table	Action	Notes
cm3tcategory (PD change task categories)	Adds the following categories: • Build and Test • Change Backout activity • Change Execution • CMDB Update • Deployment activity • Implementation activity • PIR • Planning and Schedule • Risk and Impact Analysis	If the categories already exist in the target environment, backs them up with a SUFFIX of "_bak_by_PD," sets “Availability” to false, and then inserts the PD categories; otherwise, directly inserts the PD categories.

Table	Action	Notes
cm3rsubcat (PD change subcategories)	Adds the following subcategories: • Emergency Change > Emergency • Normal Change > Major • Normal Change > Minor • Standard Change > Hardware • Standard Change > Maintenance • Standard Change > Network • Standard Change > Routine Jobs • Standard Change >	If the subcategories already exist in the target environment, backs them up with a SUFFIX of "_bak_ by_PD," and then inserts the PD categories; otherwise, directly inserts the PD subcategories.

Table	Action	Notes
	Software • Standard Change > User Admin Requests	

Table	Action	Notes
cm3rcatphase (PD change phases)	Adds the following phases: • Abandoned • Backout • Closure • ECAB Approval • Implementation • Post Implementation Review • Registration and Categorization • Risk and Impact Analysis • CMDB Update • DCAB Approval • Deployment • TCAB Approval	If the phases already exist in the target environment, backs them up with a SUFFIX of "_bak_by_PD," and then inserts the PD phases; otherwise, directly inserts the PD phases. A change category should reference the backup phase if it referenced the original phase that is backed up.

Table	Action	Notes
	• Validation • Execution • Plan and Schedule	
cm3tcatphase (PD change task phases)	Adds the following phases: • Abandoned • Active • Canceled • Closed • Review • Waiting	If the phases already exist in the target environment, backs them up with a SUFFIX of "_bak_by_PD," and then inserts the PD phases; otherwise, directly inserts the PD phases. A change task category should reference the backup phase if it referenced the phase that is backed up.

Changes for PD Request

This section describes the changes that will be made to the Request module by the Process Designer Enablement script.

Object changes for PD Request

The Process Designer Enablement script changes the following Request Objects by appending "_disabled_by_PDRM" to the original file name and renaming the temporary new Objects from <file name>_for_pdrequest_tobe_used to just <file name>:

Object File Name before Enablement	Object File Name after Enablement	Description
ocmq	ocmq_disabled_by_PDRM	The Legacy ocmq Object.
ocml	ocml_disabled_by_PDRM	The Legacy ocml Object.
ocmo	ocmo_disabled_by_PDRM	The Legacy ocmo Object.
request_for_pdrequest_tobe_used	request	The Process Designer request Object.
requestTask_for_pdrequest_tobe_used	requestTask	The Process Designer requestTask Object.

Menus changes for PD Request

Process Designer Request uses new menu items to access the Request modules. The Enablement script renames the legacy Request menu item descriptions and adds related new Process Designer Request menus.

Top Menu Name
ADMIN
HELPDESK GEN
HOME

RM GEN
APPROVER GEN

Changes to top menus

The following table describes the changes of legacy Request menu items under the top menus.

Legacy Menu Item Name	Legacy Menu Item Description before Enablement	New Menu Item Description after Enablement	New Menu Item Name
RM	Request Management	Request Fulfillment	RM PD

The following menu items are removed from the "ENV RECORDS" menu record (located under **System Administration** > **Ongoing Maintenance** > **Environment Records**):

ENV menu item
Request Management Quote Environment
Request Management Order Environment
Request Management Line Item Environment

The following menu item is removed from the "MODULE PROFILES" menu record (located under **System Administration** > **Ongoing Maintenance** > **Profiles**):

Profile menu item
Request Management Profiles

SearchConfig

Process Designer Request uses a new search form, so the Enablement script adds two SearchConfig values as shown in the following table:

File Name	Search Form
request	rm.advFind.search
requestTask	rm.requestTask.advFind.search

In addition, the following search configurations have "and false" appended to the new "allowAdvAccess" field. This is because ocmq, ocml and ocmo are no longer used in Process Designer Request.

File Name	Legacy Allow Advanced Find	New Allow Advanced Find
ocmq	(index("SysAdmin", $lo.ucapex)>0 or index("OCMAdmin", $lo.ucapex)>0 or index("OCMQ", $lo.ucapex)>0)	(index("SysAdmin", $lo.ucapex)>0 or index("OCMAdmin", $lo.ucapex)>0 or index("OCMQ", $lo.ucapex)>0) and false
ocml	(index("SysAdmin", $lo.ucapex)>0 or index("OCMAdmin", $lo.ucapex)>0 or index("OCMQ", $lo.ucapex)>0)	(index("SysAdmin", $lo.ucapex)>0 or index("OCMAdmin", $lo.ucapex)>0 or index("OCMQ", $lo.ucapex)>0) and false
ocmo	(index("SysAdmin", $lo.ucapex)>0 or index("OCMAdmin", $lo.ucapex)>0 or index("OCMQ", $lo.ucapex)>0)	(index("SysAdmin", $lo.ucapex)>0 or index("OCMAdmin", $lo.ucapex)>0 or index("OCMQ", $lo.ucapex)>0) and false

Display Options

The Process Designer Enablement script disables the following Legacy Display Options.

Display Option ID before Enablement	Display Option ID after Enablement	Description
am.display.joinfile_do nothing_18	am.display.joinfile_do nothing_18_disabled_by_ PDRM	Scheduled Maintenance > Generate Recurring > Requests
am.display.joinfile_do nothing_2	am.display.joinfile_do nothing_2_disabled_by_ PDRM	Related > Quotes >Open
am.display.joinfile_do nothing_4	am.display.joinfile_do nothing_4_disabled_by_ PDRM	Related > Quotes > View Existing

The Process Designer Enablement script enables the following Display Options.

Display Option ID before Enablement	Display Option ID after Enablement	Description
am.display.joinfile_do nothing_18_for_pdrequest_ tobe_used	am.display.joinfile_do nothing_ 18	Scheduled Maintenance > Generate Recurring > Requests

svcCatInterface

The Process Designer Enablement script replaces the " Open New Request " svcCatInterface with “Open New Request_for_pdrequest_tobe_used.”

Inbox

The Process Designer Enablement script makes the following changes to the view records.

View	File	Query in Legacy System	How to modify
My Group's To Do List	Todo	((itemType="probsummary" or itemType="incidents" or itemType="rootcause" or itemType="rootcausetask" or itemType="knownerror" or itemType="cm3r" or itemType="cm3t") and group isin $lo.pm.assignments) or ((itemType="ocmq" or itemType="ocml") and group=$lo.dept.corp.structure)	Change the Query definition as follows: (itemType="probsummary" or itemType="incidents" or itemType="rootcause" or itemType="rootcausetask" or itemType="imTask" or itemType="cm3r" or itemType="cm3t"or itemType="timeperiodDefinition" or itemType="SurveyDefinition" or itemType="request" or itemType="requestTask") and group isin $lo.pm.assignments'
My To Do List of Requests	Todo	assignee=$lo.user.name and itemType="request" and status~="Suspended"	Add
My To Do List of RequestTasks	Todo	assignee=$lo.user.name and itemType="requestTask"	Add
My Pending Delegated Approvals	Approval	(file.name="ocmq" and current.pending.groups isin $G.delegated.ocmq.groups)	Replace the legacy query with: (file.name="request" and current.pending.groups isin $G.delegated.pdrm.groups)
All my Approvals	Approval	(file.name="ocmq" and (current.pending.groups isin approval.groups in $G.ocmq.environment or current.pending.groups isin {$lo.user.name}) or current.pending.groups isin $G.delegated.ocmq.groups)	Replace the legacy query with: (file.name="request" and (current.pending.groups isin approval.groups in $G.ocmq.environment or current.pending.groups isin {$lo.user.name}) or current.pending.groups isin $G.delegated.pdrm.groups)

Links

The Enablement script changes the link lines by renaming the legacy link to <link line name>_disabled_by_PDRM and the temporary new link line from <link line name>_for_pdrequest_tobe_used to the link names shown in the following table.

Link	Link line name	Description
myGroups	member.of	The target table is changed from cm3groups to assignment.
myGroups	approver.of	The target table is changed from cm3groups to assignment.
ocmlrec	category	The target table is changed from ocmlcat to prodCatalogCategory.
screlate.get.association	ocmq	Disable this legacy line.

Besides, since we have a new "productCatalog" table to replace the legacy "model" table, the Enablement script searches all link lines with target file as "model", then disables the link lines by adding suffix "_disabled_by_PDRM", and creates a new link line by copying from the old link line and changing the target file to "productCatalog".

extaccess

The Process Designer Enablement script adds the following extaccess to enable PD Request.

Extaccess Name	Service Name	Object Name
request	RequestManagement	RM
requestTask	RequestManagement	RequestTask

Globalist

The Process Designer Enablement Script change the following globalists.

GlobalList Name	Description
SLO Tables	Append 2 values: “request” ,”requestTask”

Scheduler

The Process Designer Enablement script deletes schedulers that are named as “OCM Create Order.”

Service Desk Environment

The Process Designer Enablement script changes the RM Post Back Link of Service Desk Environment to "request.incident.post.back."

Assignment groups

"Assignment" is used as the group file universally instead of "ocmgroups" and "cm3groups" now. Therefore, the legacy groups defined in "ocmgroups" and "cm3groups" are migrated to "assignment".

Send Documentation Feedback

If you have comments about this document, you can contact the documentation team by email. If an email client is configured on this system, click the link above and an email window opens with the following information in the subject line:

Feedback on Process Designer Migration Guide (Service Manager 9.40)

Just add your feedback to the email and click send.

If no email client is available, copy the information above to a new message in a web mail client, and send your feedback to ovdoc-ITSM@hp.com.

We appreciate your feedback!


Java™
COMPATIBLE