



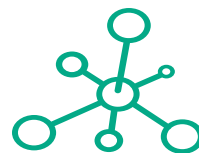
Project and Portfolio Management Center

Software Version: 9.40

Demand Management Configuration Guide

Document Release Date: September 2016

Software Release Date: September 2016



**Hewlett Packard
Enterprise**

Legal Notices

Warranty

The only warranties for Hewlett Packard Enterprise 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. Hewlett Packard Enterprise 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 Hewlett Packard Enterprise 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

© 2016 Hewlett Packard Enterprise Development LP

Trademark Notices

Adobe® is a trademark of Adobe Systems Incorporated.

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

UNIX® is a registered trademark of The Open Group.

Documentation Updates

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

This site requires that you register for an HP Passport and to sign in. To register for an HP Passport ID, click **Register** on the HPE Software Support site or click **Create an Account** 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 HPE sales representative for details.

The following table indicates changes made to this document since the last released edition.

Support

Visit the HPE Software Support site at: <https://softwaresupport.hpe.com/>.

Most of the support areas require that you register as an HP Passport user and to sign in. Many also require a support contract. To register for an HP Passport ID, click **Register** on the HPE Support site or click **Create an Account** on the HP Passport login page.

To find more information about access levels, go to: <https://softwaresupport.hpe.com/web/softwaresupport/access-levels>.

HPE Software Solutions Now accesses the HPSW Solution and Integration Portal website. This site enables you to explore HPE Product Solutions to meet your business needs, includes a full list of Integrations between HPE Products, as well as a listing of ITIL Processes. The URL for this website is <https://softwaresupport.hpe.com/km/KM01702731>.

Contents

Chapter 1: Getting Started with Demand Management Configuration	11
Introduction to Demand Management	12
Demand Management Concepts	12
Demand Management Entities	13
Overview of a Simplified Demand Management Process	14
Overview of Configuring Demand Management	15
When you're ready to deploy Demand Management: Educate your users	16
Related Information	16
Chapter 2: Gathering Process Requirements	18
Overview of Gathering Process Requirements	19
Defining Workflows	19
Gathering Information for Workflow Steps	20
Gathering Information for Decision Steps	21
Gathering Information for Execution Steps	22
Gathering Information for Condition Steps	23
Gathering Information for Subworkflow Steps	24
General Workflow Design Guidelines	25
Workflow and Request Interaction	27
Defining Request Types	28
Request and Workflow Interaction	30
Request Type Checklist	31
Defining Contacts	33
Contacts Checklist	33
Defining Notification Templates	34
Notification Template Checklist	34
Defining User Data Fields	35
User Data Checklist	35
Defining Security and Access	35
Security and User Access Checklist	36
Chapter 3: Configuring Workflows	38
Overview of Workflows	39

Opening the Workflow Workbench	43
Creating Workflows	44
Configuring General Information for a Workflow	44
Choosing Workflow Steps	45
Decisions Workflow Steps	46
Condition Workflow Steps	46
Execution Workflow Steps	47
Subworkflow Workflow Steps	48
Adding Steps to a Workflow	48
Adding the Close Step	49
Configuring Reopen Workflow Steps	50
Adjusting Workflow Step Sequences	50
Verifying and Enabling Workflows	51
Configuring Workflow Steps	51
Configuring Properties of a Workflow Step	52
Configuring Security for Workflow Steps	54
Configuring Notifications for Workflow Steps	55
Configuring the Setup Tab	57
Configuring Message Content	65
Configuring Timeouts for Workflow Steps	68
Configuring Transitions for Workflow Steps	69
Adding Transitions Based on Specific Results	70
Adding Transitions not Based on Specific Results	71
Adding Transitions Back to the Same Step	77
Adding Transitions Based on Previous Workflow Step Results	79
Adding Transitions To and Removing them From Subworkflows	81
Configuring Validations for Workflow Steps	82
Validations and Execution Type Relationships	83
Adding Color to Workflow Steps	84
Specifying Font Size for Workflow Layout Images	85
Configuring Segregation of Duties for Workflow Steps	85
Integrating Request Types and Workflows	86
Integrating Workflows and Request Types through Workflow Step Properties	86
Integrating Request Type Commands and Workflows	87

Integrating Request and Package Workflows	88
Step 1. Setting Up WF - Jump/Receive Step Label Validations	89
Step 2. Generating Jump Step Sources	90
Step 3. Generating Receive Step Sources	91
Step 4. Including Jump and Receive Workflow Steps in Workflows ...	92
Chapter 4: PPM Center Mobility Access	94
Overview of PPM Center Mobility Access	95
Mobility Access Best Practices	97
Mobility Access Limitations	97
Mobility Access in a Multilingual User Interface Context	98
Mobility Access Deployment	98
Installing the Mobility Access	98
Mobility Access Configuration	99
Enabling and Scheduling the Mobility Access Service	99
Mobility Access Server-Side Configuration Settings	100
Configuring the Email Server	101
Configuring User-Defined Markers	104
Adding Mobility Access to Decision Step Notifications	105
Configuring Mobility Access Service Logging	106
Chapter 5: Configuring Request Types and Request Header Types .	108
Overview of Request Types	108
Request Type Components and Configuration Options	111
Controlling Request Field Behavior	113
Status Dependencies	113
Request Type Rules	114
Opening the Request Type Workbench	114
Setting Request Type Defaults	115
Configuring General Information for Request Types	116
Creating and Configuring Request Type Fields	118
Overview of Request Type Fields	118
Criteria for Visible Fields	118
Criteria for Editable Fields	119
Criteria for Default Fields	120
Creating Fields for Request Types	121

Copying Fields for Request Types	126
Removing Fields from Request Types	127
Configuring Layouts for Request Types	127
Modifying Field Widths on Request Types	127
Moving Fields On Request Types	128
Adding Sections to Request Types	129
Changing Section Names on Request Types	130
Deleting Sections on Request Types	131
Configuring Display Columns for Request Types	131
Configuring Request Statuses for Request Types	133
Overview of Request Statuses	133
Creating Request Statuses for Request Types	135
Configuring Request Field Status Dependencies	136
Status Dependencies Interactions	138
Request Type Rules	139
Types of Request Type Rules	140
Predefined JavaScript Functions (Advanced Rules Only)	140
Rule Event Behavior on the Request Details Page	142
Rule Event Precedence	144
Creating Simple Default Rules for Request Types	144
Advanced Rules for Request Types	146
Important Considerations for Configuring Advanced UI Rules	146
UI Rules: Examples	150
Creating Advanced Request Type Rules	157
SQL Rules: Using Functions of KNTA_USER_UTIL Package	160
Configuring Commands for Request Types	165
Adding Commands to Request Types	166
Editing Commands of Request Types	167
Copying Commands in Request Types	168
Deleting Commands in Request Types	168
Command Conditions	168
Configuring Sub-Types for Request Types	169
Adding Sub-Types to Request Types	169
Editing Sub-Types for Request Types	170
Deleting Subtypes from Request Types	171

Configuring Request Types to Work with Workflows	171
Adding Workflows to Request Types	171
Deleting Workflows from Request Types	172
Configuring Participants for Requests	172
Adding Request Participants to a Request Type	172
Editing Participants on Request Types	174
Deleting Participants from Request Types	175
Configuring View and Edit Access Grants for Request Creators	175
Configuring Import and Export of Requests	176
Configuring an XML Importable Request Template	176
Developing an XSL File	176
Configuring an XSLT Template	178
Exporting Requests to XML Files	179
Enabling Report Type (REFERENCE) Export Request Report	179
Configuring an XML Exportable Request Template	180
Exporting Requests by Running the Special Command	181
Configuring Quick Edit and Mass Update	182
Configuring Resource Tracking	182
Tracking Resources Assigned to Requests	182
Configuring Request Types for Use with Time Management	185
Configuring Notifications for Request Types	187
Adding Notifications	187
Configuring the Setup Tab	187
Configuring Message Tab	190
Editing Notifications	192
Copying Notifications	193
Deleting Notifications	193
Configuring Ownerships of Request Types	194
Adding Ownerships to Request Types	194
Deleting Ownerships from Request Types	195
Configuring Help Contents for Request Types	195
Configuring Request Header Types	196
Overview of Request Header Types	197
Request Header Type Field Groups	198
Opening the Request Header Type Workbench	201

Configuring General Information for Request Header Types	201
Configuring Filters for Request Header Types	202
Chapter 6: Enabling Service for Requests	205
Enabling Service for Requests	205
Enable Service Field Group for Request Header Type	206
Enable Service Field Group for Request Type	207
Chapter 7: Configuring Workflow Components	208
Overview of Workflow Step Sources	209
Restrictions on Configuring and Using Workflow Step Source	210
Opening the Workflow Workbench	210
Creating Workflow Step Sources	211
Configuring Ownership of Workflow Step Sources	212
Creating Decision Workflow Step Sources	212
Creating Execution Workflow Step Sources	215
Setting Up Execution Steps	219
Defining Executions Types	220
Executing Request Type Commands	220
Closing Requests as Success	221
Closing Requests as Failed	222
Executing PL/SQL Functions and Creating Transitions Based on the Results	223
Executing SQL Statements and Creating Transitions Based on the Results	223
Evaluating Tokens and Creating Transitions Based on the Results ..	224
Executing Multiple System-Level Commands	226
Creating Subworkflow Workflow Step Sources	226
Subworkflows Returning to Demand Management Workflows	227
Using Workflow Parameters	228
Creating Workflow Parameters	228
Example: Using Workflow Parameters to Build a Loop Counter	229
Modifying Workflows Already In Use	230
Performance Considerations	231
Copying and Testing Trial Versions of Workflows	231
Modifying Production Workflows	232
Disabling Workflow Steps	232

Redirecting Workflows	232
Moving Requests or Packages Out of Steps	233
Chapter 8: Configuring Contacts	234
Overview of Contacts	234
Opening the Contact Workbench	235
Creating Contacts	235
Chapter 9: Configuring Notification Templates	237
Overview of Notification Templates	238
Opening the Notification Template Workbench	238
Deleting Notification Templates	239
Creating Notification Templates	239
Configuring Ownership of Notification Templates	241
Deleting Ownerships from Notification Templates	242
Configuring Notification Intervals	242
Checking the Usage of Notification Templates	243
Chapter 10: Configuring User Data	245
Overview of User Data	246
Referencing User Data	247
Migrating User Data	247
User Data Configuration Tasks	248
Opening the User Data Workbench	248
Viewing General Information for User Data Types	249
Creating a User Data Context	250
Configuring User Data Fields	251
Copying a Field Definition	255
Editing User Data Fields	256
Configuring User Data Field Dependencies	256
Removing Fields	258
Configuring User Data Layouts	258
Changing Column Widths	259
Moving Fields	259
Swapping Positions of Two Fields	260
Previewing the Layout	260
Configuring User Data for Resources	261

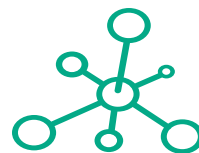
Appendix A: Worksheets	264
Configuration Workflow Worksheet	265
Execution Workflow Step Worksheets	266
Decision Workflow Step Worksheets	267
Subworkflow Workflow Step Worksheets	269
Request Type Configuration Sheets	270
Appendix B: Examples	275
A Simple PL/SQL Function for Execution Steps and Field Population	275
Executing PL/SQL Functions With A Parameter and Creating Transitions Based on the Results	277
Examples of Using Advanced Rule with SQL-default Logic	279
PL/SQL Function Example	280
Example A	281
Example B	282
Appendix C: Configuring A Single Email Notification for Multiple Recipients with Different Locales	284
Appendix B: Appendix D: Switching Off Pagination on Builder Portlets of Requests Category	286
Best Practices on Builder Portlets Pagination	286
Send documentation feedback	288



Chapter 1: Getting Started with Demand

Management Configuration

- "Introduction to Demand Management" on the next



**Hewlett Packard
Enterprise**

[page](#)

- ["Demand Management Concepts " below](#)
- ["Overview of Configuring Demand Management" on page 15](#)
- ["Related Information" on page 16](#)

Introduction to Demand Management

Demand Management is the Project and Portfolio Management Center (PPM Center) product that automates your business processes. At the core of this functionality are a flexible form builder and an integrated workflow engine that let you digitize both simple and complex processes. Demand Management works by capturing requests and processing them based on the processes and business rules created for each type of request.

The process behind each request is modeled, automated, enforced, and measured on your best-practice business processes. In addition, a detailed audit trail helps you pinpoint problems quickly and supports regulatory compliance requirements, such as segregation of duties (SOD), at both the role level and the process step level.

Users complete a request form using a standard Web browser. Each type of request has its own configurable form and an associated workflow that determines what data must be captured and what process applied for reviewing, evaluating, prioritizing, scheduling, and resolving the request. Based on the workflow, the reviewer can assign the request to a person or team for scheduling and delivery.

Notifications defined as part of the process can be activated at any step in the process to indicate work is to be done, has not been done, is being escalated, and so on. Demand Management includes the Web-based PPM Dashboard, which delivers the right information to anyone with a browser.

This document provides the details on how to configure an Demand Management system using the PPM Workbench, and includes the information you need to ensure that your requests follow your digitized business processes. This section presents an overview of how you can configure Demand Management to support your business processes.

Demand Management Concepts

To understand how Demand Management works, it is important that you be familiar with the entities described in this section.

Demand Management Entities

The following four high-level PPM Center entities are associated with Demand Management:

- **Request Header Types.** Request header types are configuration entities that determine the fields displayed in the header section of the request details page for requests of a given type. To see all fields in the header section, open the workbench and go to the **Layout** tab of the Request Type window.
- **Request Types.** Request types are configuration entities that define the structure, logic, and access control of request Web forms. Demand Management includes such predefined system request types as the Bug request type and Enhancement request type to serve as example configurations.
- **Workflows.** Workflows are another kind of Demand Management configuration entity. A workflow is a digitized process composed of a logical series of steps that define a business process. Workflow steps can range in usage from reviews and approvals to performing migrations and executing system commands.
- **Requests.** Requests are transactional entities that represent the fundamental work unit of the request resolution piece of Demand Management. Users create requests and submit them to a resolution process (workflow). The request contains all information typically required to complete a specific business process. The process that the request follows is determined by the workflow assigned to it.

In addition to these configuration and transactional entities, Demand Management involves the following entities:

- **Contacts.** Contacts contain business contact information (such as a business card) about people who serve as points of reference for other Demand Management entities, such as requests. A contact can refer to a PPM Center user, but more likely represents a person outside of the organization who may have some interest in the entity on which he is referenced.

Note: Because contact information does not represent a user account in PPM Center, a contact cannot access Demand Management unless that contact has a valid PPM Center user account.

- **Notification Templates.** Notification templates are preconfigured, parameterized email notification messages that you can use with the various Demand Management entities, such as workflows and requests, to automatically send email notifications of various events. You can also create your own

notification templates.

- **Request Resolutions.** Request resolution refers to the creation, processing, and closing of requests. A request can be anything from a simple question to a detailed report of a software defect.

Overview of a Simplified Demand Management

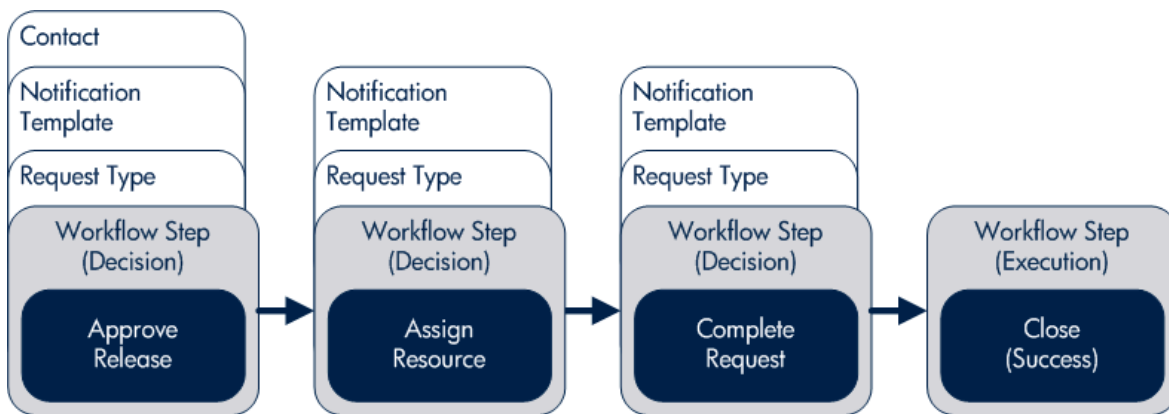
Process

"Figure 1-1. PPM Center components" below shows a simple four-step Demand Management process to approve a release. The first step, Approve Release, is a decision workflow step in which a user receives a release request. After the user manually approves the release, the request process moves to the second step.

Note: Decision steps represent manual activities performed outside of PPM Center, whereas execution steps represent actions that are automated through PPM Center.

In the second step, Assign Resource (a decision workflow step), a manager manually assigns a resource to the release. Once a resource is assigned, the step is completed and the process moves to the third step.

Figure 1-1. PPM Center components



On the third step, Complete Request (a decision workflow step), the assigned resource fulfills the request. The request then moves to the fourth and, in this example, final step, Close (Success). This is an execution step at which the release process automatically closes and notifies users the release was successfully closed.

Overview of Configuring Demand Management

Demand Management system configuration involves the following tasks:

Tip: ["Appendix A: Worksheets" on page 264](#) contains a series of worksheets to help you gather the information required to build an Demand Management system.

Step 1: Gather process requirements

Before configuring an Demand Management system, you should collect specific information concerning your process, the types of requests required, and your contacts. For detailed information, see ["Gathering Process Requirements " on page 18](#).

Step 2: Configure workflows

Configuring the workflows that you assign to requests involves setting up the required workflow steps (decision and execution steps), adding transitions between the steps, and configuring notifications, security groups, segregation of duties, and so on for each step. For information about how to configure workflows, see ["Configuring Workflows" on page 38](#) and ["Configuring Workflow Components" on page 208](#).

Step 3: Configure request types

Request types gather and track the information required to perform workflow steps. For information about how to configure request types, see ["Configuring Request Types and Request Header Types" on page 108](#).

Step 4: Configure contacts

Contacts are Demand Management users used as points of reference or information by other Demand Management entities, such as requests. For information about how to configure contacts, see ["Configuring Contacts" on page 234](#).

Step 5: Configure notification templates

Notification templates are preconfigured notification forms used with Demand Management workflows and request types. ["Configuring Notification Templates" on page 237](#) presents detailed information on how to create and configure notification templates.

Step 6: Configure user data fields

In addition to the fields defined for each type of request in request types and request header types, you may want to define some additional, more global fields for all request types. Creating *user data* is a convenient way to define such global fields for Demand Management workflows and request types.

"[Configuring User Data](#)" on page 245 provides more information about user data fields, including instructions on how to configure them.

Step 7: Configure your security and access requirements

Part of any process are the permissions required to perform various decision steps. PPM Center controls access to perform these decisions through licenses and access grants. For information about licenses and access grants, see the *Security Model Guide and Reference*.

When you're ready to deploy Demand Management: Educate your users

After your Demand Management system is configured and tested, train your users on the new business process. The following offers some guidance on how to prepare your Demand Management users:

- **Basic Demand Management training.** Make sure that each user understands how to create, process, and report on requests.
- **Process-specific training.** Make sure that each user understands the new process. Consider holding a formal meeting or publishing documents on the workflow steps and requests.
- **User Responsibilities.** Make sure that every user understands their respective role in the process. For example, the QA team may be restricted to only approving the testing phase of a release. You can use email notifications that are part of Demand Management to communicate information about user roles. Your notifications can be very detailed.

Related Information

The following documents include additional information on how to configure or use Demand Management:

- *Release Notes*
- *Demand Management User's Guide*
- *Tracking and Managing IT Demand Configuration Guide*
- *Tracking and Managing IT Demand User's Guide*
- *Commands, Tokens, and Validations Guide and Reference*

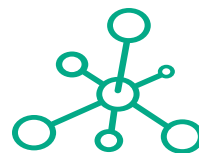
- *Time Management User's Guide*
- *Open Interface Guide and Reference*
- *Reports Guide and Reference*
- *Security Model Guide and Reference*
- *Creating Portlets and Modules*
- *Customizing the Standard Interface*
- *Getting Started*
- *What's New and What's Changed*
- *Multilingual User Interface Guide*
- *HPE-Supplied Entities Guide* (includes descriptions of all Demand Management portlets, request types, and workflows)



Chapter 2: Gathering Process Requirements

- "Overview of Gathering Process Requirements" on the next page
- "Defining Workflows" on the next page
- "Defining Request Types" on page 28
- "Defining Contacts" on page 33

- "Defining Notification Templates" on page 34



**Hewlett Packard
Enterprise**

- ["Defining User Data Fields" on page 35](#)
- ["Defining Security and Access" on page 35](#)

Overview of Gathering Process Requirements

This section presents an overview of the information to collect before you configure an Demand Management process and guidance on how to collect it. This information includes the steps to add to your workflows, the types of requests your organization requires, and the contacts you might need. After you collect this information, you can begin to configure your Demand Management process.

This section covers the following topics:

- **Defining workflows.** What are the steps of your demand management process (workflow)? Which steps require manual decisions (reviews and approvals)? Which steps require automatic executions? (See ["Defining Workflows" below](#).)
- **Defining request types.** What are you requesting? For detailed information, see ["Defining Request Types" on page 28](#).
- **Defining contacts.** What contacts are required? For detailed information, see ["Defining Contacts" on page 33](#).
- **Defining notification templates.** Is the correct notification template in place? Does your process require a new notification template? For detailed information, see ["Defining Notification Templates" on page 34](#).
- **Defining user data fields.** Does your process require additional user information to process correctly? For detailed information, see ["Defining User Data Fields" on page 35](#).
- **Defining security and access.** Who can submit requests? Who can receive notifications? Who can approve the request at each step? For instructions on how to configure security, see ["Defining Security and Access" on page 35](#).

Defining Workflows

A workflow is a digitized process in which a logical series of steps define the path that the request follows. Workflow steps can range from reviews and approvals to automatically updating a status or closing a workflow process.

Before you define a request workflow, you must first determine the objective of the business process that you want the workflow to achieve. For example:

- Do you want to design a simple approval process with little oversight or supervision?
- Do you want to design a business-wide bug-tracking system that has intensive oversight and supervision?

Once you determine the objective of the business process, you can begin to define the workflow itself. The basic workflow components are:

- **Workflow steps.** Workflow steps are the events that link together to form the process.
- **Transitions between workflow steps.** Transitions between workflow steps represent the outcome of one workflow step that leads to next workflow step. Workflow steps can have more than one transition.
- **Security determines who can access a workflow step.** Each workflow step includes a list of who can access workflow step. Who can approve a workflow step? Can only one user approve the workflow step? Can one of several users approve the workflow step? Must multiple users approve the workflow step?
- **Notification determines who hears about the workflow step and when they hear about it.** Each workflow step includes a list of users to be notified about the workflow step.

Gathering Information for Workflow Steps

Workflow steps are the events of the process. Demand Management workflows can include the following types of steps:

- **Decision steps.** These are steps that require an external action (such as review, approval, or coding) to determine outcome.
- **Execution steps.** Execution steps perform work or actions, such as automatic time-stamping or automatic request status changes.
- **Condition steps.** Condition steps, such as AND and OR, are logic steps used for complex workflow processing.
- **Subworkflows steps.** Subworkflow steps, such as code rework or unit testing, contain multiple workflow steps that follow a consistent pattern.

To determine what steps to include in a workflow, consider the following:

- What event starts the business process?
- At what points in the process must decisions be made?
- At what points in the process must actions be taken?

Gathering Information for Decision Steps

"[Table 2-1. Decision workflow checklist](#)" below provides a checklist of issues to consider as you define decision type workflow steps. For a complete list, see "[Decision Workflow Step Worksheets](#)" on [page 267](#).

Table 2-1. Decision workflow checklist

Done	Decision Step Check Item	Example
	What is the name of this workflow step?	<ul style="list-style-type: none"> • Review request • On hold • In rework
	What is the status of the request at this workflow step?	<ul style="list-style-type: none"> • On hold • New • In review
	What are the transitions from this workflow step?	<ul style="list-style-type: none"> • Assign • Review • Approve • On hold
	Who or what groups can act on this step (approve, cancel, reassign)?	<ul style="list-style-type: none"> • Security groups • Users • Tokens
	How many decisions are required to exit this workflow step?	<ul style="list-style-type: none"> • Only one • At least one • All
	What event triggers the notification?	<ul style="list-style-type: none"> • Process reaches the workflow step

Table 2-1. Decision workflow checklist, continued

Done	Decision Step Check Item	Example
		<ul style="list-style-type: none"> • Specific result is achieved
	Who receives the notification?	<ul style="list-style-type: none"> • Email address (group alias) • Security group
	What is the notification message?	<ul style="list-style-type: none"> • Test complete • Approval required
	Use this workflow step as a timeout? If yes, then for how long?	<ul style="list-style-type: none"> • 1 day • 2 days
	Are you using segregation of duties?	<ul style="list-style-type: none"> • Based on workflow owner? • Based on the workflow step?

Gathering Information for Execution Steps

Execution steps involve work or actions, such as time-stamping or request status changes, that PPM Center performs automatically. Use the checklist in "[Table 2-2. Execution workflow checklist](#)" below to help you define execution steps. For a complete list of execution step issues to consider, see "[Execution Workflow Step Worksheets](#)" on page 266.

Table 2-2. Execution workflow checklist

Done	Execution Step Check Item	Example
	What is the name of this workflow step?	<ul style="list-style-type: none"> • Create request • Close • Set temp date
	Will this workflow step execute this command?	<ul style="list-style-type: none"> • Cancel request • Update request
	What is the execution type?	<ul style="list-style-type: none"> • Close • Jump • Return from subworkflow
	What is the processing type?	<ul style="list-style-type: none"> • Immediate

Table 2-2. Execution workflow checklist, continued

Done	Execution Step Check Item	Example
		<ul style="list-style-type: none"> • Manual
	What is the source environment (group)?	PPM Server
	What is the destination environment (group)?	PPM Server
	What are the transitions from this workflow step?	<ul style="list-style-type: none"> • Succeeded • Failed
	Who owns this execution step?	<ul style="list-style-type: none"> • Security group • User
	What event triggers the notification?	<ul style="list-style-type: none"> • The process reaches the workflow step • A specific result is achieved
	Who receives the notification?	<ul style="list-style-type: none"> • Email address (group alias) • Security group
	What is the notification message?	<ul style="list-style-type: none"> • Test complete. • Approval required.
	Use this workflow step as a timeout? If yes, then for how long?	<ul style="list-style-type: none"> • 1 day • 2 days
	Are you using segregation of duties?	<ul style="list-style-type: none"> • Based on the workflow owner? • Based on workflow step?

Gathering Information for Condition Steps

Condition steps are logic steps, such as AND and OR, that are used for complex workflow processing. "Table 2-3. Condition workflow checklist" below provides a checklist of items to consider as you define the condition steps for a workflow.

Table 2-3. Condition workflow checklist

Done	Condition Step Check Item	Example
	What is the name of this workflow step?	<ul style="list-style-type: none"> • AND • OR

Table 2-3. Condition workflow checklist, continued

Done	Condition Step Check Item	Example
	What is the status of the request at this workflow step?	<ul style="list-style-type: none"> • On hold • New • In review
	What are the transitions from this workflow step?	<ul style="list-style-type: none"> • Succeeded • Failed
	Who (or what group or token) owns this workflow step?	<ul style="list-style-type: none"> • Security group • User • Standard token • User-defined token
	What event triggers the notification?	<ul style="list-style-type: none"> • The process reaches the workflow step • A specific result is achieved
	Who or how many receive the notification?	<ul style="list-style-type: none"> • Email address (group alias) • Security group
	What is the notification message?	<ul style="list-style-type: none"> • Test complete • Approval required
	Use this workflow step as a timeout? If yes, then for how long?	<ul style="list-style-type: none"> • 1 day • 2 days
	Are you using segregation of duties?	<ul style="list-style-type: none"> • Based on the workflow owner? • Based on the workflow step?

Gathering Information for Subworkflow Steps

A subworkflow step, such as code rework or unit testing, includes multiple workflow steps that follow a consistent pattern. You can use the checklist in "[Table 2-4. Subworkflow Workflow Checklist](#)" on the [next page](#) to help you define subworkflow steps. For a complete list of subworkflow step considerations, see "[Subworkflow Workflow Step Worksheets](#)" on [page 269](#).

Table 2-4. Subworkflow Workflow Checklist

Done	Subworkflow Step Check Item	Example
	Is an existing workflow available as a subworkflow?	<ul style="list-style-type: none"> • Yes • No
	What is the name of this subworkflow?	<ul style="list-style-type: none"> • QA test cycle • QA review cycle
	What are the transitions from this workflow step?	<ul style="list-style-type: none"> • Succeeded • Failed
	Who owns this workflow step?	<ul style="list-style-type: none"> • Security group • User
	What event triggers the notification?	<ul style="list-style-type: none"> • The process reaches the workflow step • A specific result is achieved
	Who receives the notification?	<ul style="list-style-type: none"> • Email address (group alias) • Security group
	What is the notification message?	<ul style="list-style-type: none"> • QA test cycle succeeded. • QA test cycle failed.
	Use this workflow step as a timeout? If yes, then for how long?	<ul style="list-style-type: none"> • 1 day • 2 days
	Are you using segregation of duties?	<ul style="list-style-type: none"> • Based on owner of the workflow? • Based on workflow step?

General Workflow Design Guidelines

Use the checklist in "[Table 2-5. Logical workflow guidelines](#)" below to help you configure your workflow.

Table 2-5. Logical workflow guidelines

Done	Guideline	Reason
Workflows		
	Make one or more workflows available to	Each workflow is assigned one of the

Table 2-5. Logical workflow guidelines, continued

Done	Guideline	Reason
	process the request.	following workflow scopes: <ul style="list-style-type: none"> • Request (Demand Management) • Packages (Deployment Management) • Release distributions (Deployment Management)
Beginning and Closing Steps		
	Workflow must have a beginning step.	No processing can occur if the workflow has no starting point.
	Workflow must have at least one step.	No processing can occur if the workflow has no steps.
	Workflow must have at least one Close step.	Request cannot be closed without a Close step in the workflow.
	First workflow step cannot be a condition step.	Workflow processing may not be correct if the first step is a condition.
	Close steps must not have a transition on 'Success' or 'Failure.' Return steps must have no outgoing transitions.	Request cannot close if a transition exists on 'Success.'
	Close step in subworkflow closes entire request.	Do not include a Close step in a subworkflow unless you want to close the workflow in the subworkflow.
All Steps		
	All steps must be enabled.	Because the workflow cannot use disabled steps, the process stops.
	Each step (except the first step) must have at least one incoming transition.	It is not possible to flow to a workflow step without an incoming transition.
	Transition value is not a validation value (error).	The validation value has changed since the transition was made.
	'Other Values' and 'All Values' transitions must not occur at the same step.	If both transitions occur at a step, the 'Other Values' transition is ignored.
	Each workflow step must have at least one outbound transition.	Without an outbound transition, the workflow branch stops indefinitely without closing the request.
	Each value from a list-validated validation must have an outbound transition.	Some validation values do not have defined transitions.

Table 2-5. Logical workflow guidelines, continued

Done	Guideline	Reason
	Steps with either a text or numeric validation must have an 'Other Values' or 'All Values' transition.	Because text and numeric validations are not limited, you must specify an 'Other Values' or 'All Values' transition.
	Notifications with reminders must not be set on results that have transitions.	Transition into the Return Step does not match the validation.
Decision Steps		
	Each decision step must have at least one security group, user, or token specified on the Security tab.	No one can act on the step if security is not configured.
Execution Steps		
	Each manual execution step must have at least one security group, user, or token specified on the Security tab.	No one can act on the step if security is not configured.
	An immediate execution step must not have a transition to itself on 'Success' or 'Failure.'	The workflow could loop indefinitely.
Condition Steps		
	A condition step must not have a transition to itself.	A condition with a transition to itself could cause the workflow to run indefinitely.
	An AND or OR step must have at least two incoming transitions.	An AND or OR condition with only one incoming transition will always be true and have no effect.
Subworkflows		
	Subworkflows must have at least one Return step.	Must include a Return step.
	A top-level workflow must not have a Return step.	Only subworkflows can have a Return step.

Workflow and Request Interaction

Request status can change as a request moves through its resolution process. Each request status can control request field attributes, such as whether or not a field is visible, editable, required, optional, and so on.

Request status can be tied to a workflow step, so that when a request reaches a certain workflow step, it acquires a status that changes the attributes of a field. The request status at a given workflow step can also drive field logic during the life of the request.

Typically, a given request type is associated with a single workflow. Information contained in the request (defined in the request type) works together with the workflow process to ensure that the request is correctly processed. Although you can use one workflow with many different request types, the level of possible integrations between request type and workflow is easier with a one-to-one mapping.

Defining Request Types

Requests are instances of *request types*. A request type defines the Web form that users see when they create or view requests of that type. Each request type defines the set of fields specific to that type of request.

Each request type definition also specifies which *request header type* to use. The request header type defines sets of standard fields that are common to multiple request types. The request header type includes options for enabling integration with other HPE products, both within the PPM Center product suite (Program Management, Project Management, and Portfolio Management) and outside of the suite (such as HPE Universal Configuration Management Database, HPE Quality Center, and HPE Service Center).

Different information is required to process each request. For example, to resolve a software bug, you might need to know the software unit, product version, problem, priority, and so on. The fields on the request type and request header type capture this information.

Before you create a request type, determine what standard fields are available for the request (request header types and field groups). The fields displayed in the **Summary** section of a request detail page (see "[Figure 2-1. Sample request](#)" on the next page) are derived from the request header type associated with the request type. The fields in the details section are defined in the request type itself. To see all fields in the details section, open the workbench and go to the **Layout** tab of the Request Type window.

Figure 2-1. Sample request

Create New Enhancement

Expand All | Collapse All

Submit Cancel

Summary

Created By:
Admin User

*Department: Manufacturing Sub-Type:

*Workflow:
Bug Request Type Workflow

Priority: High Application: HR Application

Assigned To: PPM Demand Management Administrator

Request Group:

*Description:
Create a new module for onboarding in Singapore office.

Request Status:
Not Submitted

*Contact Name:

Contact Phone:

Contact Email:

Enhancement

Module: Module C Difficulty: Medium

Modification Type: New Estimated Time to Complete:

Report Name:

Program Name:

*Justification:
Newly opened Singapore office staffing

Resolution:

Duplicate ID:

Resolution Summary:

+ Notes

+ References

For each request type, provide the following information:

- Name of the request and request type
- Request header type attached to this request
- Fields to display on the request
- Request status values, such as Pending, On hold, Approved, and Canceled
- Notifications to send when the value of a selected field changes
- Request-level access information to specify who is allowed to create, view, and edit requests of

this type

- Workflows that can be used by requests of this type

For each new field required on the request type (or the request header type), gather the following information:

- Field label. Specify the field label to display next to the field in the Web form, to ensure that the correct information is captured.
- Information type. What type of information must be collected? Is this a text field, a drop-down list, or an auto-complete field? The validation specified for a field determines this.
- Field behavior. You can control many aspects of field behavior, including:
 - Whether (and at what point in the workflow) the field is editable, read-only, required, hidden, and so on. Both the workflow (process) and the behavior of other fields in the form can control field behavior. For example, you can configure a field to be required only when the request reaches the "Assign" status.
 - Whether the field is populated automatically based on values in other fields.
 - Who can view and edit the field, and who must be restricted from viewing the information in the field.

For more information about request types and request type fields, see ["Appendix A: Worksheets" on page 264](#).

Request and Workflow Interaction

Request status can change as the request moves through a workflow toward resolution. Each request status can control its request field attributes, such as field visibility or editability. A request status can be tied to a workflow step so that when the request reaches that step, it acquires the status specified by that step. The request status at a particular workflow step can then drive field logic during the life of the request.

In addition to setting the status of the parent request, you can also configure a workflow to specify who is assigned to the request at each step. The workflow step can drive both the **Assigned To** field and the **Assigned Group** field. You can set these fields based on dynamic properties of the parent request through the use of tokens, facilitating automatic routing of the request as it moves through its workflow process.

Typically, a single request type is associated with a single workflow. Information contained in the request (defined in the request type) works together with the workflow process to ensure that the request is processed correctly. Although you can apply one workflow to many different request types, the level of possible integration between request type and workflow is more practical with a one-to-one mapping.

Request Type Checklist

"[Table 2-6. Request type configuration checklist](#)" below provides a configuration consideration checklist to help define your Demand Management system. For a complete list of request type considerations, see "[Request Type Configuration Sheets](#)" on page 270.

Table 2-6. Request type configuration checklist

Done	Request Type Check Item	Configuration Consideration
	Request type considerations.	A request type must be defined for each type of request to be resolved. This includes creating fields that describe the request and decisions and field logic required to process it during resolution.
	Is a request header type associated with the request type?	A request header type must be associated with the request type. If no appropriate request header type exists, create one.
	Are fields defined?	<ul style="list-style-type: none"> • Fields are required to define the request. • Ensure the correct parameters are used to describe the request to be processed. <p>For more information, see "Creating and Configuring Request Type Fields" on page 118 and the <i>Commands, Tokens, and Validations Guide and Reference</i>.</p>
	Are request rules defined?	You can set rules to automatically populate fields in the request, or define more dynamic behavior on the request form. For details, see " Request Type Rules " on page 139.
	Are request status values defined?	Define the status values that the request can have and associate them with the request type. You can add new status values, if necessary. For details, see " Configuring Request Statuses for Request Types " on page 133.
	Are status dependencies set?	You can configure request fields to be hidden,

Table 2-6. Request type configuration checklist, continued

Done	Request Type Check Item	Configuration Consideration
		required, read-only, cleared, or reconfirmed, based on the request status. For details, see "Configuring Request Field Status Dependencies" on page 136.
	Is request security set?	You can control who participates in request resolution. For information about how to set request security, see "Creating Fields for Request Types" on page 121 and the <i>Security Model Guide and Reference</i> .
	Is request field security set?	You can configure request fields to be invisible to specific users and security groups. For more information, see "Creating and Configuring Request Type Fields" on page 118 and the <i>Security Model Guide and Reference</i> .
	Are request notifications set?	You can configure notifications to be sent automatically at specific points in your process. For details, see "Configuring Notifications for Request Types" on page 187.
	Are user data fields defined?	Use user data to define global fields for requests, if necessary. For more information, see "Configuring User Data" on page 245.
	Are fields defined for the request type?	Fields are required to define the request. Make sure that the correct parameters describe the request. For more information, see "Creating Fields for Request Types" on page 121.
	Is the request type enabled?	Disabled request types cannot be submitted by users. (You can find the Enabled option in the Request Type window.)
	Cover all request type and workflow considerations.	<ul style="list-style-type: none"> • Decide which request type status values correspond to each workflow step. • Decide which workflow steps will change the request's Assigned To or Assigned Group fields. • Decide which workflow steps are to execute any request type commands. • Verify that workflow step source validations and request type field validations agree. This is required if a transition is based on a field value (using tokens, SQL or PL/SQL

Table 2-6. Request type configuration checklist, continued

Done	Request Type Check Item	Configuration Consideration
		<p>execution types).</p> <ul style="list-style-type: none"> • Allow the request type use for the workflow (set on the workflow window Request Types tab). • Allow the workflow to be used by the request type (set on the Request Type window Workflows tab).

Defining Contacts

Contacts are resources that Demand Management entities (such as requests) use as reference points or information sources. For information about how to configure contacts, see ["Configuring Contacts" on page 234](#).

For each contact you plan to configure, collect the following information:

- First name
- Last name
- Username
- Phone number
- Email address
- Company

Contacts Checklist

You can use the configuration checklist in ["Table 2-7. Contacts checklist" below](#) to define your contacts.

Table 2-7. Contacts checklist

Done	Contacts Check Item	Configuration Consideration
	Is the contact enabled?	Disabled contacts are unusable.

Table 2-7. Contacts checklist, continued

Done	Contacts Check Item	Configuration Consideration
	Is the contact a PPM Center user?	The Username field is an auto-complete list for selecting PPM Center users. If the contact is not a PPM Center user, leave the field empty.

Defining Notification Templates

Notification templates are preconfigured email forms that you can use to quickly construct the body of an email message. You can use these notification templates with Demand Management entities such as workflows and requests.

As you configure a workflow, you can select a notification template to use for each workflow step. Demand Management comes with a set of standard notification templates. You can use these templates as they are, customize them, or create new notification templates tailored to your business process. For detailed information on how to configure notification templates, see ["Configuring Notification Templates" on page 237](#).

Notification Template Checklist

Use the checklist in ["Table 2-8. Notification template checklist" below](#) to help define your notification templates.

Table 2-8. Notification template checklist

Done	Notification Template Issue	Configuration Consideration
	Is the notification template enabled?	Disabled notification templates are unusable.
	Notification template and security group considerations.	Set ownership groups for these entities. Only ownership group members (determined by associating security groups) can edit the entities.

Defining User Data Fields

In addition to the process-specific fields defined in request types and request header types, you may want to capture specific information on every request submitted in Demand Management, regardless of request type. To capture such information, you can define global user data fields. For instructions on how to create user data fields, see ["Configuring User Data" on page 245](#).

User Data Checklist

Use the checklist in ["Table 2-9. User data checklist" below](#) as you define user data fields.

Table 2-9. User data checklist

Done	User Data Issue	Configuration Consideration
	Are the user data fields enabled?	Disabled user data fields are unusable.
	User data field-level security considerations	For each user data field, specify who can view and who can edit the field contents on a request (if necessary).

Defining Security and Access

Part of an Demand Management process is the security configured for workflow steps. PPM Center controls permission to perform decision and execution steps using the following mechanisms:

- **Licenses.** Licenses give users access to PPM Center products, but do not determine the specific actions a user is authorized to perform within the products.
- **Access Grants.** Access grants (used with licenses) determine the actions a user can perform within a given PPM Center product.

For example, you can restrict what an Demand Management user can do using the following license and access grant combination:

- License
 - Demand Management
- Access Grants

- View Requests
- Edit Requests

For more information about licenses and access grants, see the *Security Model Guide and Reference*.

HPE recommends that you specify security groups or tokens (dynamic access) to set workflow security. Avoid using a list of individual users to control an action. If the user list changes (as a result of department reorganization, for example), you would have to update your workflow configuration in several places to keep the process running correctly. If you use a security group, you update the security group once, and the changes are propagated throughout the workflow. Tokens are resolved dynamically at runtime and thus adapt to the current system context as necessary.

"[Table 2-10. Example of workflow security groups](#)" below lists example workflow steps and the security groups that have access to the workflow and each workflow step.

Table 2-10. Example of workflow security groups

Workflow Step	Security Groups
Validate Request	Financial Apps - Validate and Approve Requests Financial Apps - Manage Resolution System
Pending More Information	Financial Apps - Create and View Requests Financial Apps - Manage Resolution System
Approve Request	Financial Apps - Validate and Approve Requests Financial Apps - Manage Resolution System
Schedule Work	Financial Apps - Schedule Requests Financial Apps - Manage Resolution System
Develop Enhancement	Financial Apps - Develop Requests Financial Apps - Manage Resolution System

For more information about setting security for workflows and requests, see the *Security Model Guide and Reference*.

Security and User Access Checklist

Use the checklist in "[Table 2-11. Security and user access checklist](#)" on the next page to help determine your security and user access requirements.

Table 2-11. Security and user access checklist

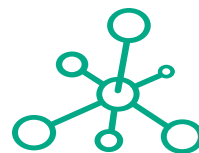
Done	Security and User Access Issue	Configuration Consideration
	Created the security groups to be granted access to screens and functions.	Required security groups have been created.
	Created security groups to associate with workflow steps.	Security groups to allow users to act on a specific workflow step have been created.
	Set security on request creation.	All available options that restrict who can create and submit requests are set.
	Set security on request processing.	All available options that restrict who can process requests are set.
	Set security on request system configuration.	Users who can modify the request process have been granted required permissions. This includes editing the workflow, object type, environment, security group assignment, and so on.
	Cover all security group and workflow considerations.	<ul style="list-style-type: none"> • Associate security groups with workflow steps. Group members can act on the step. • Set workflow and workflow step ownership.
	Cover all security group and object type considerations	Set ownership groups for object types. Only members of the ownership group (determined by associating security groups) can edit the object type.
	Cover all security group and environments considerations.	Set ownership groups for environments. Only members of the ownership group (determined by associating security groups) can edit the environments.
	Cover all security group and notification template considerations.	Set ownership groups for notification templates. Only members of the ownership group (determined by associating security groups) can edit the notification templates.
	Cover all security group and user data considerations.	Set ownership groups for user data. Only members of the ownership group (determined by associating security groups) can edit user data.



Chapter 3: Configuring Workflows

- "Overview of Workflows " on the next page
- "Opening the Workflow Workbench" on page 43
- "Creating Workflows" on page 44
- "Configuring Workflow Steps " on page 51

- "Integrating Request Types and Workflows" on



Hewlett Packard
Enterprise

Overview of Workflows

A workflow represents a business process and is used to map business rules and processes to your organization. This section covers information about Demand Management workflows.

The basic components of a workflow are as follows:

- **Begin.** For each workflow, you must explicitly define the first eligible workflow step.
- **Workflow step.** Workflow steps are events that are linked together to form a complete workflow. The basic types of workflow step are:
 - **Decision.** Decision steps represent manual activities performed outside of PPM Center. For example, a user or group of users approves a request.
 - **Execution.** Execution steps represent actions that are automated through PPM Center. For example, a Web page is updated with the results of a test.
 - **Condition.** Condition steps are logic steps used in complex workflow processing. For example, you can set up a condition step that allows the workflow to proceed only after each workflow step is completed.
 - **Subworkflows.** A subworkflow step represents multiple workflows steps (the subworkflow) in a workflow. For example, a test workflow step in the main workflow represents a series of tests and approvals.
- **Transition.** The results of workflow step that must be communicated to another workflow step. A transition occurs after a workflow step is completed.

Examples

- The result of a decision step is Approved or Not Approved.
- The transition for a step labeled Analysis and Design (for a software application) could be Completed or Needs More Work.

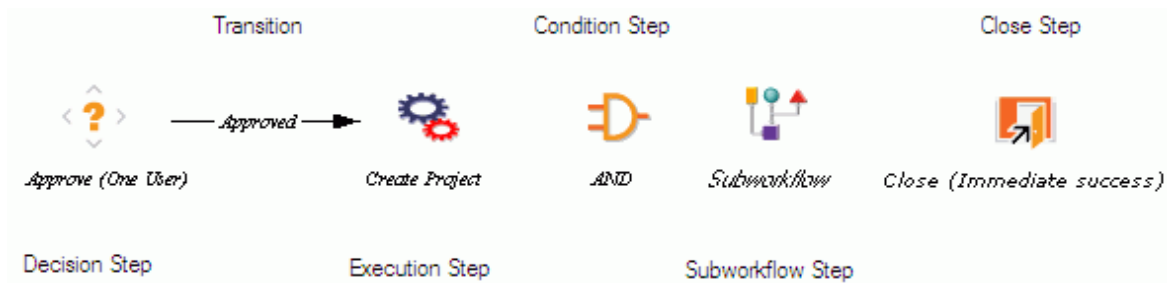
Because a single step can have several possible results, you can define multiple outgoing transitions for each workflow step.

- **Workflow step security.** Workflow step security determines who has permission to execute or choose a result for a workflow step. For example, you can specify that only the IT project manager can approve or deny an Approve Request decision step.

- **Notification.** Notifications are email alerts sent out at specific workflow steps. For example, when a request reaches an Approve Request decision step in the workflow, an email alert is sent to the product manager.
- **Close step.** A close step ends the workflow. It is an execution step that marks the request as completed.

"Figure 3-1. Workflow components" below shows examples of common workflow components.

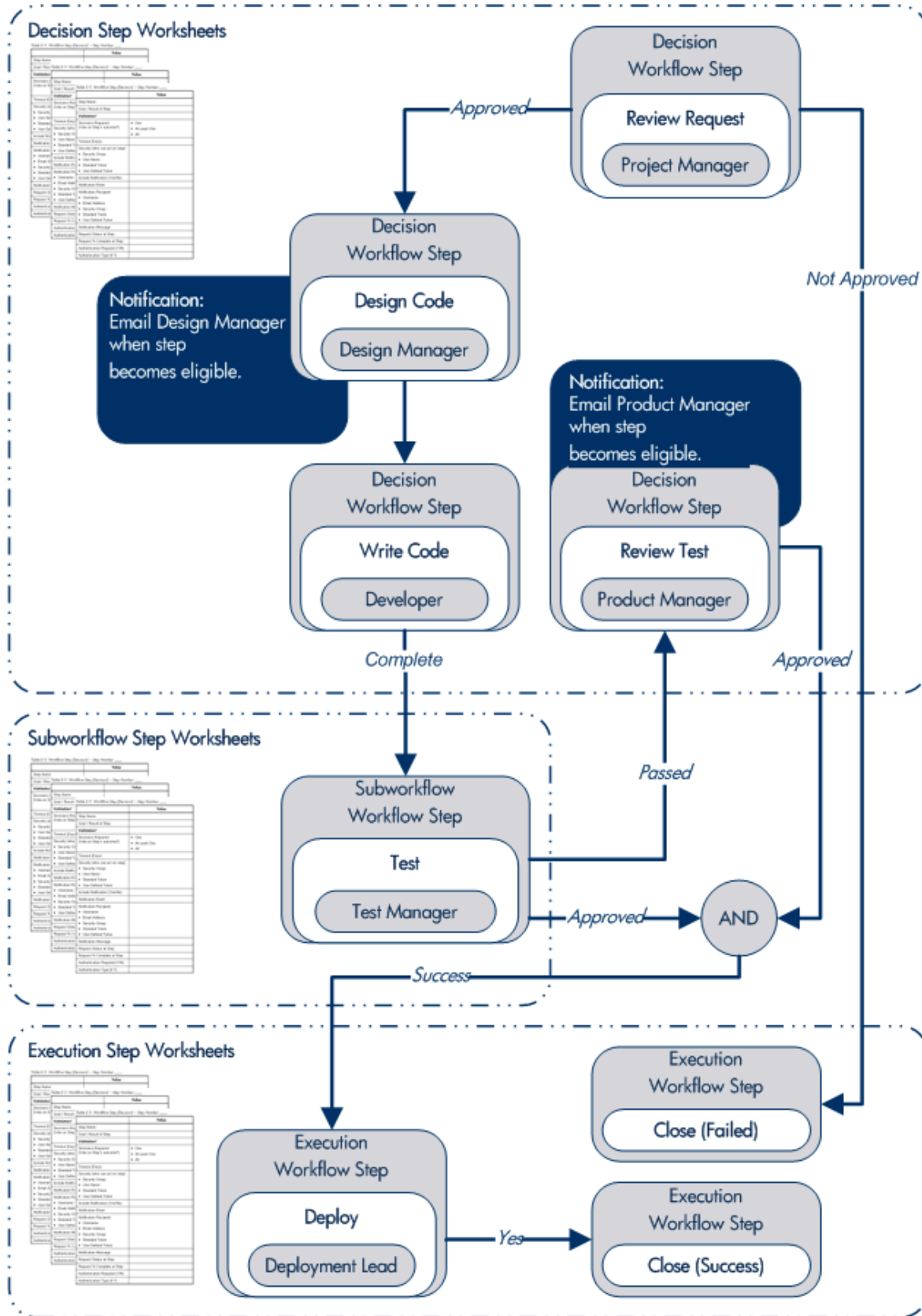
Figure 3-1. Workflow components



Mapping all of the individual workflow steps into a single workflow is a two-stage process.

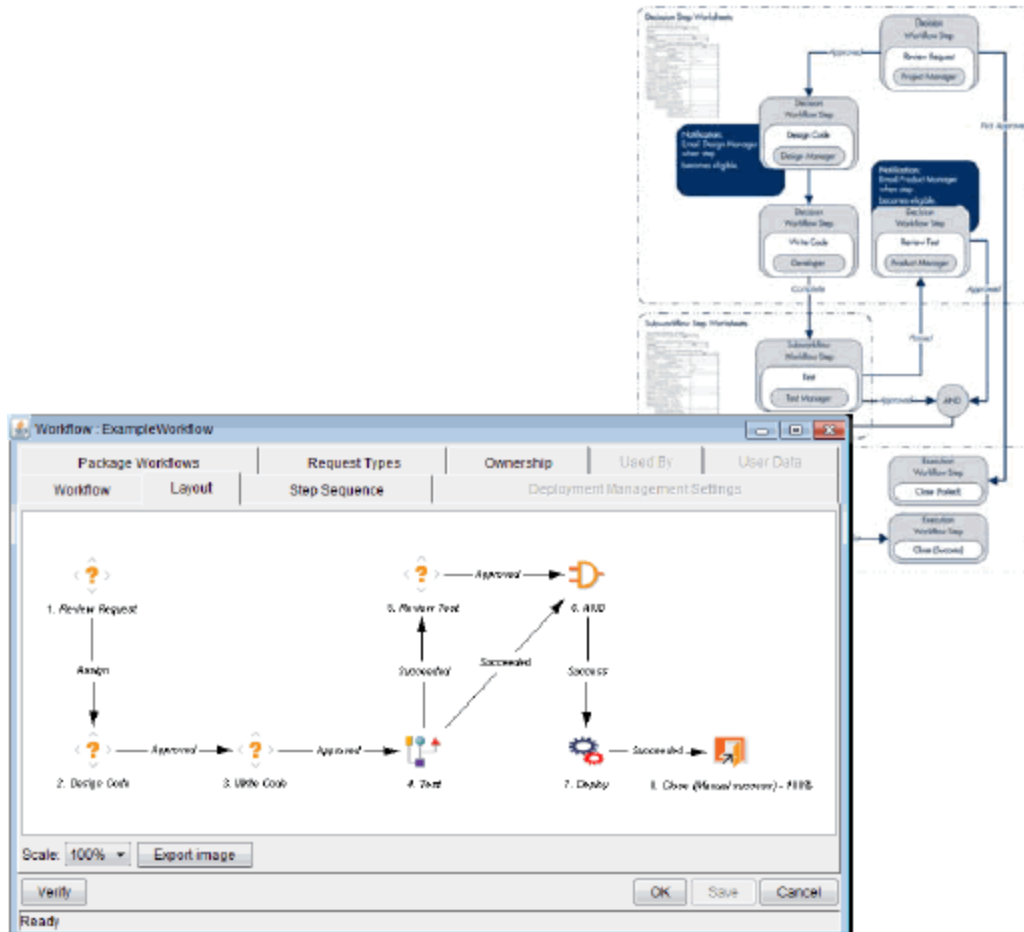
Stage 1. Create a block diagram (see "Figure 3-2. Stage 1. Create a block diagram" on the next page). Map each workflow step worksheet as one block in the diagram. Include transitions, workflow step security, and notifications. Use the worksheets provided in "Appendix A: Worksheets" on page 264 to help you construct the diagram.

Figure 3-2. Stage 1. Create a block diagram



Stage 2. Map the block diagram to the workflow. Open the Workflow Workbench and create a workflow. Map each component from the block diagram to the new workflow (see "Figure 3-3. Stage 2. Create the workflow" below).

Figure 3-3. Stage 2. Create the workflow



Opening the Workflow Workbench

To open the Workflow Workbench:

1. Log on to PPM Center.
2. On the **Open** menu, click **Administration > Open Workbench**.

The PPM Workbench opens.

3. On the shortcut bar, click **Configuration > Workflows**.

The Workflow Workbench and Workflow Step Sources windows open.

For information about how to search for and select an existing workflow, copy a workflow, or delete a workflow, see the *Getting Started* guide.

Creating Workflows

This section provides instructions on how to use the Workflow Workbench to create a workflow.

Configuring General Information for a Workflow

To create and provide basic information for a workflow:

1. On the PPM Workbench shortcut bar, click **Configuration > Workflows**.

The Workflow Workbench and Workflow Step Sources windows open.

2. In the Workflow Workbench window, click **New Workflow**.

The Workflow window opens.

3. Provide values for the fields listed in the following table.

Field or Option	Description
*Required	
*Name	Type a name for the workflow.
*Reference Code	After you type the workflow name, a reference code is automatically generated. You can either leave this default value, or type a different value.
Workflow Scope	Leave Requests selected.
Description	Type a short description of the workflow and its purpose.
Enabled	To make this workflow available in PPM Center, click Yes .
*First Step	This box displays the value NONE until you add steps to the workflow from the Layout tab. (See "Adding Steps to a Workflow" on page 48.)
Subworkflow	A workflow can contain other workflows. If you want to nest another workflow within the new workflow, click Yes .
Validation	Use this auto-complete to specify the validation that sets the possible subworkflow results. A value is required if Subworkflow is set to Yes .
Icon Name	Type the name of an image file to represent the subworkflow on the Layout tab. This graphic file must be in .gif format and must reside in the <PPM_Home>/icons directory.

4. Click **Save**.

Choosing Workflow Steps

PPM Center comes with predefined templates for commonly used workflow steps. These are available through the Workflow Step Sources window in the Workflow Workbench.

A workflow step source defines the behavior of a step (conditions for exiting the step, commands to execute for the step, timeout duration, which icon to display, and so on) as well as the list of possible result values or outcomes for the step.

Note: For detailed information about workflow step sources, see ["Configuring Workflow Components" on page 208.](#)

You can use the **Filter by** fields in the Workflow Step Sources window to filter the workflow steps listed. The following folders, which contain workflow steps classified by type, are available in the Workflow Step Source window:

- **Decisions**
- **Conditions**
- **Executions**
- **Subworkflows**

To add a step to your workflow, determine which of the folders it corresponds to. Expand the folder, and then drag the workflow step that best suits your needs to the **Layout** tab.

If you do not find an available workflow step source that meets the requirements of the workflow you are configuring, you can define a new workflow step source. For instructions on how to define a workflow step source, see ["Creating Decision Workflow Step Sources" on page 212](#).

Decisions Workflow Steps

Decision workflow steps represent manual activities performed outside of PPM Center. Decision workflow steps include such activities as:

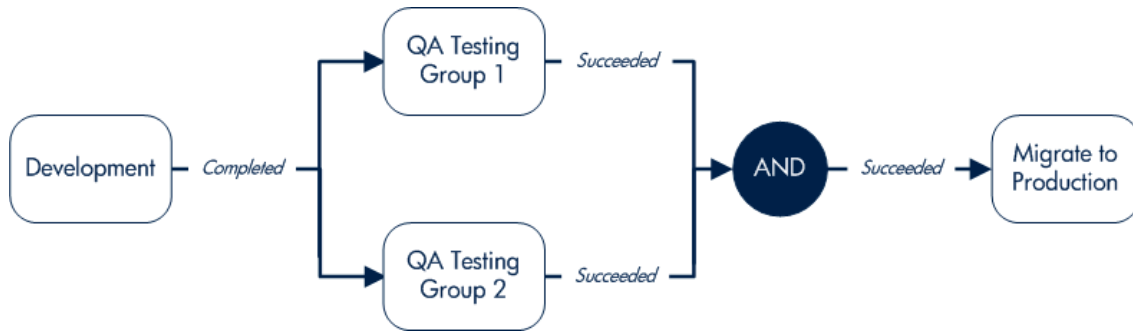
- Decisions made by committees
- Code designs and reviews

Condition Workflow Steps

Condition workflow steps are logic steps used for complex workflow processing, such as allowing the workflow to proceed only after each workflow step is completed. The condition workflow steps are as follows:

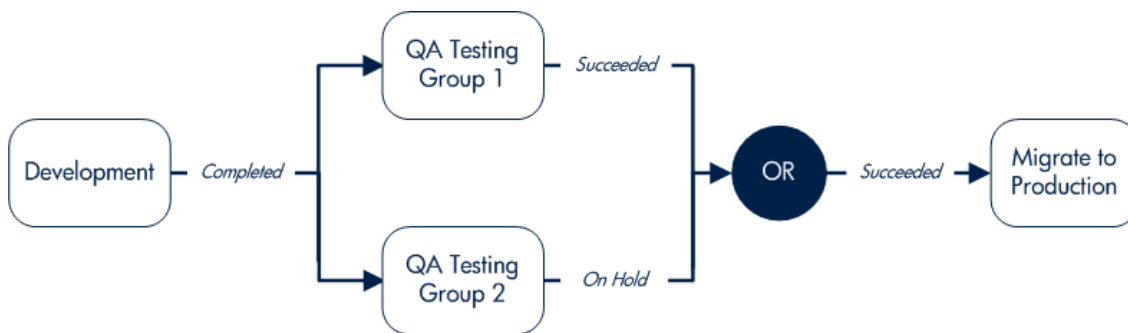
- **AND**. The AND condition is met only after all workflow steps leading to it reach the specified required status. ["Figure 3-5. AND example" on the next page](#) shows an AND condition workflow step.

Figure 3-5. AND example



- **OR.** The OR condition is met if at least one of the workflow steps leading to it reaches the required status specified for it "Figure 3-6. OR example" below shows an OR condition workflow step.

Figure 3-6. OR example



Execution Workflow Steps

Execution workflow steps represent actions that are automated through PPM Center. Execution workflow steps include such activities as:

- Create a package
- Run object type commands
- Package priority
- Create a request
- Execute request commands
- Run workflow step commands
- Close the workflow (Close workflow step)

Subworkflow Workflow Steps

A subworkflow is a process unit that contains a series of steps that perform a functional subcomponent of a workflow. Subworkflows allow you to model complex business processes in logical, manageable, and reusable subprocesses. Within its parent workflow, each subworkflow is represented as a single workflow step.

After the workflow process reaches the subworkflow step, it follows the path defined in that subworkflow. Subworkflows can either end the workflow or return to the parent workflow.

The following restrictions apply to subworkflows:

- You cannot use a subworkflow to process a request or a package as a stand-alone business process.
- A subworkflow can reference other subworkflows, but not itself.
- A subworkflow can be referenced only by workflows or subworkflows of the same workflow scope.
- Permissions specified on the **Security** tab of the calling subworkflow step determine who can bypass the steps with the subworkflow.

Adding Steps to a Workflow

You assemble workflow steps into workflows on the **Layout** tab of the Workflow window.

To add a step to a new workflow:

1. In the Workflow window for your new workflow, click the **Layout** tab.

To the right of the Workflow window, the Workflow Step Sources window contains a library of steps, classified by type, that you can use to build your workflows. The window also includes **Filter by** lists, which you can use to selectively display a subset of available steps.
2. From the first **Filter by** list, select **Requests**.
3. You can use the second **Filter by** list to select an additional filter condition to further refine the steps available for this workflow.
4. To view the available steps, expand the folders in the Workflow Step Sources window.

Note: For more information about how to select the steps for your workflows, see "[Choosing Workflow Steps](#)" on page 45

5. Determine which step to add as the first step, and then drag and drop it on the **Layout** tab.

After you add a step to the **Layout** tab, the Workflow Step window opens. Use this window to configure the following:

- General workflow step properties

For instructions on how to configure the properties of a step, see "[Configuring Properties of a Workflow Step](#)" on page 52.

- Workflow step security

For instructions on how to configure step security, see "[Configuring Security for Workflow Steps](#)" on page 54.

- Notifications for the workflow step

For instructions on how to configure workflow step notifications, see "[Configuring Notifications for Workflow Steps](#)" on page 55.

- Timeouts for the workflow step

For instructions on how to configure workflow timeouts, see "[Configuring Timeouts for Workflow Steps](#)" on page 68.

- Step fill color for graphic workflow display

For instructions on how to select a fill color for a step, see "[Adding Color to Workflow Steps](#)" on page 84.

- Segregation of duties

For instructions on how to configure segregation of duties, see "[Configuring Segregation of Duties for Workflow Steps](#)" on page 85

6. After you finish configuring all of the steps in the workflow, click **OK**.

Adding the Close Step

Every workflow must include a close step. A close step is a type of execution workflow step. You can find it in the **Executions** folder in the Workflow Step Sources window.

You can use one of the following three close steps in a workflow:

- **Close (Immediate success).** This close step immediately completes a request or package with a status of Success.
- **Close (Manual success).** This close step requires manual intervention to complete a request or package and set the request or package status to Success.
- **Close (Immediate failure).** This close step immediately completes a request or package with a status of Failure.

You add a close workflow step to a workflow as you would any other type of step.

Configuring Reopen Workflow Steps

If necessary, users who have the required access grants can reopen closed requests. A reopened request begins at the reopen workflow step specified for the workflow.

To specify a reopen step for a workflow:

1. Open a workflow in the Workflow Workbench.
2. Click the **Workflow** tab.
3. In the **Reopen Step** list, select the reopen workflow step.
4. Click **Save**.

Adjusting Workflow Step Sequences

After you assemble all of the workflow steps on **Layout** tab, you can adjust their sequence.

To adjust the sequence of steps in an open workflow:

1. In the Workflow window, click the **Step Sequence** tab.
The **Step Sequence** tab lists all of the workflow steps.
2. Select a workflow step, and then click the up and down arrows at the bottom of the tab to move the selected workflow in the display sequence.
3. Click **Save**.

On the **Workflow** tab, the **First Step** field displays the first workflow step.

Verifying and Enabling Workflows

To make a workflow available for use you must verify it, and then enable it. Workflow verification ensures correct workflow logic. Enablement makes the workflow available to users.

To verify a workflow:

1. On the PPM Workbench shortcut bar, click **Configuration > Workflows**.
The Workflow Workbench opens.
2. Open the workflow to verify.
3. In the lower left corner of the Workflow window, click **Verify**.

If the verification process uncovers no problems in the logic of the workflow, a message is displayed to indicate that no errors were detected. If the verification process uncovers problems with the workflow, its steps, or its transitions, the Verify window opens and lists the errors.

To enable a workflow that is not enabled:

1. Open the Workflow Workbench.
2. Open the workflow that you want to enable.
The Workflow window opens to the **Workflow** tab.
3. For **Enabled**, click **Yes**.
4. Click **Save**.

Configuring Workflow Steps

After you drag a workflow step from the Workflow Step Source window to the **Layout** tab in the Workflow window, the Workflow Step window opens. You can provide some or none of the step information after the window first opens, or you can provide it later in the workflow design process.

Tip: "[Appendix A: Worksheets](#)" on page 264 contains worksheets that you can use to capture detailed information about your workflows, workflow steps, and transitions.

The following table lists the tabs available in the Workflow Step window.

Tab	Description
Properties	This tab displays general information about the workflow step.
Security	This tab displays permission settings for specific individuals or groups authorized to act on a workflow step.
Notifications	Use this to define email notifications to send when a workflow step becomes eligible or after a workflow step is completed. Notifications can inform a user of a task (workflow step) to perform (such as review and approve a new request). Notifications can also inform a group of users of the results of a task.
Timeout	Use this tab to specify how long a workflow step can remain inactive before an error is generated.
User Data	Product entities such as packages, workflows, requests, and projects include a set of standard fields that provide information about those entities. While these fields are normally sufficient for day-to-day processing, user data fields provide the ability to capture additional information specific to each organization. User data is defined under the User Data tab. If there are no user data fields, the User Data tab is disabled.
Results	This tab lists the validation included in each workflow step, the component type, and the results.
Segregation of Duties	Use this tab to configure workflow steps to take into account segregation of duties, excluding the participants for a workflow step from participating in a different workflow step.
Display Settings	Use this tab to select a fill color for the graphical display of the selected step.

Configuring Properties of a Workflow Step

You can use the **Properties** tab in the Workflow Step window to complete or edit general information about a workflow step.

To configure workflow step properties:

1. On the PPM Workbench shortcut bar, click **Configuration > Workflows**.

The Workflow Workbench opens.

2. Open a workflow.

3. On the **Layout** tab, double-click a workflow step.

The Workflow Step window opens to the **Properties** tab.

4. Provide (or modify) information for the fields listed in the following table.

Field	Description
Step Name	Name of the workflow step. The name is displayed on both the Layout and Step Sequence tabs.
Action Summary	Summary of what the step accomplishes.
Description	Short description of the step.
Enabled	Determines whether the step is available to the system for now.
Display	To display the step only when the step is available for action, select Only When Active from this list. To display the step at all times, leave Always selected.
Avg Lead Time	Informational field that you can use for reporting.
Request Status	Use this auto-complete to specify which status to set on the parent request when the request reaches this step. (This setting is important for integrating workflows and Demand Management request types.)
Current % Complete	Value to display for the parent request's percent complete when the request reaches this step. (This setting is important for integrating workflows and Demand Management request types.)
Parent Assigned To User	Specify the user to which the parent request is to be assigned when the request reaches this step. (This setting is important for integrating workflows and Demand Management request types.)
Parent Assigned To Group	Specify the security group to which the parent request is to be assigned when the request reaches this step. (This setting is important for integrating workflows and Demand Management request types.)
Workflow Step Information	Type the address of a Web page with information associated with this step.
Authentication Required	To require users to submit a username and password or just a password before they can act on this step, select Username & Password or Password . Otherwise, leave None selected.

5. Click **OK**.

Configuring Security for Workflow Steps

To determine which users or groups are authorized to act on a workflow step, you must set the permissions for the step.

To add security to a workflow step:

1. From the Workflow Workbench, open a workflow.
2. In the Workflow window, click the **Layout** tab.
3. Double-click a workflow step for which you want to configure security.

The Workflow Step window opens.

4. Click the **Security** tab.
5. Click **New**.

The Workflow Step Security window opens.

6. In the list at the top of the window, do one of the following:
 - To authorize security groups to act on the workflow step:
 - i. Leave **Enter a Security Group Name** selected.
 - ii. Use the **Security Group** auto-complete to select one or more security groups to act on the workflow step. (You can use `Shift` or `Ctrl` to select multiple groups.)
 - To authorize users to act on the workflow step:
 - i. Select **Enter a Username**.
 - ii. Use the **Username** auto-complete to select one or more users to act on the workflow step. (You can use `Shift` or `Ctrl` to select multiple usernames.)
 - To authorize users and security groups to act on the workflow step using standard tokens (that resolve to users and security groups):
 - i. Select **Enter a Standard Token**.
 - ii. Use the **Standard Token** auto-complete to select a standard token to act on the workflow step.
 - iii. In the Workflow Step Security window, click **Add**.

The token you select determines the value displayed in the **Security Type** field.

- iv. To add another token, repeat [step ii](#) and [step iii](#).
- o To authorize users and security groups to act on the workflow step using user-defined tokens (that resolve to users and security groups):
 - i. Select **Enter a User Defined Token**.
 - ii. If the token has already been defined, then in the **User Defined Token** field, type the token name. Otherwise, to open the Token Builder and define a new token that returns the resource(s) you want to act on the workflow step, click **Tokens**. (For information about how to use the Token Builder to define tokens, see the *Commands, Tokens, and Validations Guide and Reference*.)
 - iii. In the **Security Type** list, select the security type to which the token resolves.
 - iv. Click **Add**.
 - v. To add another user-defined token, repeat [step ii](#) through [step iii](#).
 - vi. Click **OK**.
 - **Username**. The token resolves to a username.
 - **User ID**. The token resolves to a user ID.
 - **Security Group Name**. The token resolves to a security group name.
 - **Security Group ID**. The token resolves to a security group ID.
7. Click **OK**.
8. To add items of a different security type, repeat [step 6](#).
9. In the Workflow Step window, click **OK**.
10. On the **Security** tab, click **OK**.
11. In the Workflow window, click **OK**.

Configuring Notifications for Workflow Steps

You can configure notifications to be sent when a workflow step becomes eligible or after a workflow step is completed. Notifications can inform a user of a task (workflow step) to perform, such as review and approve a new request. Notifications can also inform a group of users of the results of a task (workflow step). You configure notifications on the **Notifications** tab in the Workflow Step window.

Note: If you have installed and enabled the Mobility Access add-on, you can configure notifications for decision steps to be acted on by PPM Center users from their email inboxes. For information, see "[PPM Center Mobility Access](#)" on page 94

Review your Workflow Step Worksheet for notification information.

To add a notification to a workflow step:

1. On the PPM Workbench shortcut bar, click **Configuration > Workflows**.
2. Open a workflow.
3. On the **Layout** tab in the Workflow window, double-click a workflow step.
4. In the Workflow Step window, click the **Notifications** tab.
5. Click **New**.

The Add Notification for Step: <Step_Name> window opens to the **Setup** tab.

6. From the **Event** list, select an event to trigger the new notification, and then do one of the following:
 - If you selected **ALL** or **Eligible**, proceed to [step 7](#).
 - If you selected **Specific Result**, then from the **Value** list, select a step result to trigger the notification.
 - If you selected **Specific Error**, then from the **Error** list, select an error to trigger the notification.
7. From the **Interval** list, select the time interval at which to send a triggered notification.
8. In the **Recipients** section, do one of the following:
 - Click **New**, and then use the Add New Recipient window to select the notification recipients (users, security groups, or tokens).
 - To specify the users or groups listed on the **Security** tab for the step as notification recipients, click **Copy Security**.

Note: If you have installed and enabled the Mobility Access add-on, the Enable Decision by Email checkbox is available. In this case, you can configure notifications for decision steps to be acted on by PPM Center users from their email inboxes. For details, see "[PPM Center Mobility Access](#)" on page 94.

9. Click the **Message** tab.
10. From the **Notification Format** list, select the format for the message content.

11. From the **Notification Template** list, select an email template to use.
12. Configure the body of the notification, and then click **OK**.

The **Notifications** tab lists the new notification details. To send a different notification to other recipients for a different event, click **New**, and then repeat this process.

You might want to send different notifications for a single workflow step if, for example:

- A step has several possible results, which require different responses.
- The notification content depends on the type of error encountered.
- Depending on the type of step error that occurs, you want to notify recipients at different time intervals.

13. Click **OK**.

Configuring the Setup Tab

You can configure a workflow step to send notifications at different times, different intervals, following different events, and to different recipients.

Sending Notifications When Workflow Steps Become Eligible

To send a notification when a workflow step becomes eligible:

1. In the Workflow Step window, click the **Notifications** tab.

See "[Configuring Notifications for Workflow Steps](#)" on page 55.

2. Click **New**.

The Add Notification for step window opens to the **Setup** tab.

3. From the **Event** list, select **Eligible**.
4. To determine the frequency with which the notification is sent, from the **Interval** list, select a value.

Note: If you select **8:00 AM Daily M-F**, the notification will go out every morning at 8:00 AM from Monday through Friday after the step becomes eligible.

If you select **8:00 AM Daily M-F** or **Hourly M-F**, you can send multiple notifications to a single recipient in a batch.

5. To send recipients a reminder if the event is still in effect after a given number of days:
 - a. For **Send Reminder?**, select **Yes**.
 - b. In the **Reminder Days** field, type the number of days after which, if the event is still in effect, a reminder is to be sent.
6. For **Enabled**, leave **Yes** selected.
7. To stop notification transmission once the step is no longer eligible, select the **Don't send if obsolete** checkbox.
8. In the **Recipients** section, do one of the following:
 - Click **New**, and then use the Add New Recipient window to select the notification recipients (users, security groups, or tokens).
 - To specify the users or groups listed on the **Security** tab for the step as notification recipients, click **Copy Security**.
9. Click the **Message** tab.
10. Configure the body of the notification, and then click **OK**.
11. In the Workflow Step window, click **OK**.

Sending Notifications when Workflow Steps have Specific Results

You can configure a notification to be sent when a workflow step has a specific decision or execution result.

Note: If you have installed and enabled the Mobility Access add-on, you can configure notifications for decision steps to be acted on by PPM Center users from their email inboxes. For details, see ["PPM Center Mobility Access" on page 94](#)

To send notification when a workflow step has a specific result:

1. In the Workflow Step window, click the **Notifications** tab.
See ["Configuring Notifications for Workflow Steps" on page 55](#).

2. Click **New**.
3. In the Add Notification for Step window, click the **Setup** tab.
4. From the **Event** list, select **Specific Result**.
5. From the **Value** list, select the workflow step result to trigger the notification.

Note: The available values are determined by the workflow step source validation.

6. To determine the frequency with which the notification is sent, from the **Interval** list, select a value.

Note: If you select **8:00 AM Daily M-F**, the notification will go out every morning at 8:00 AM from Monday through Friday after the step becomes eligible.

If you select **8:00 AM Daily M-F** or **Hourly M-F**, you can send multiple notifications to a single recipient in a batch.

7. To send recipients a reminder if the event is still in effect after a given number of days:
 - a. For **Send Reminder?**, click **Yes**.
 - b. In the **Reminder Days** field, type the number of days after which, if the event is still in effect, a reminder is to be sent.
8. For **Enabled**, leave **Yes** selected.
9. To stop notification transmission once the step is no longer eligible, select the **Don't send if obsolete** checkbox.
10. In the **Recipients** section, do one of the following:
 - o Click **New**, and then use the Add New Recipient window to select the notification recipients (users, security groups, or tokens).
 - o To specify the users or groups listed on the **Security** tab for the step as notification recipients, click **Copy Security**.
11. Click the **Message** tab.
12. Configure the body of the notification, and then click **OK**.
13. In the Workflow Step window, click **OK**.

Sending Notifications When Workflow Steps Result in Specific Errors

You can configure the notification to be sent when a workflow step has a specific error. "Table 3-1. Specific errors for workflow steps" below lists the possible workflow step errors.

Table 3-1. Specific errors for workflow steps

Error	Meaning
No consensus	All users of all security groups, or users linked to the workflow step need to vote, and there is no consensus.
No recipients	None of the security groups linked to the workflow step have users linked to it. No user can act on the workflow step.
Timeout	The workflow step timed out. (Used for execution steps and decision steps.
Invalid token	Invalid token used in the execution.
ORACLE error	Failed PL/SQL execution.
NULL result	No result is returned from the execution.
Invalid integer	Validation includes an invalid value in the Integer field.
Invalid date	Validation includes an invalid value in the Date field.
Command execution error	Execution engine has failed or has a problem.
Invalid Result	Execution or subworkflow has returned a result not included in the validation.
Parent closed	For wf_receive or wf_jump steps, a request is expects a message from a package line that is cancelled or closed.
Child closed	For wf_receive or wf_jump steps, a package line expects a message from a request that is cancelled or closed.
No parent	For wf_receive or wf_jump steps, a request expects a message from a package line that has been deleted.
No child	For wf_receive or wf_jump steps, a package line expects a message from a request that has been deleted.
Multiple jump results	For wf_jump steps in a package Line, different result values were used to transition to the step.
Multiple Return Results	The package-level subworkflow received multiple results from package lines that traversed it.

To send notification when a workflow step has a specific result:

1. In the Workflow Step window, click the **Notifications** tab.
See ["Configuring Notifications for Workflow Steps "](#) on page 55.
2. Click **New**.
3. In the Add Notification for Step window, click the **Setup** tab.
4. From the **Event** list, select **Specific Error**.
5. From the **Error** list, select the error that you want to trigger the notification.
6. To determine the time at which the notification is sent, from the **Interval** list, select a value.

Note: If you select **8:00 AM Daily M-F**, the notification will go out every morning at 8:00 AM from Monday through Friday after the step becomes eligible.

If you select **8:00 AM Daily M-F** or **Hourly M-F**, you can send multiple notifications to a single recipient in a batch.

7. To send recipients a reminder if the event is still in effect after a given number of days:
 - a. For **Send Reminder?**, select **Yes**.
 - b. In the **Reminder Days** field, type the number of days after which, if the event is still in effect, a reminder is to be sent.
8. For **Enabled**, leave **Yes** selected.
9. To stop notification transmission once the step is no longer eligible, select the **Don't send if obsolete** checkbox.
10. In the **Recipients** section, select the notification recipients (users, security groups, or tokens). For detailed instructions, see ["Configuring Notification Recipients"](#) on page 63.
11. Click the **Message** tab, and configure the body of the notification. For details on how to do this, see ["Configuring Message Content"](#) on page 65.
12. Click **OK**.
13. In the Workflow Step window, click **OK**.

Scheduling Notifications

Use the **Interval** field in the workflow step to specify when to send the notification. The interval determines how frequently the notification is sent.

To send the time notification are sent:

1. In the Workflow Step window, click the **Notifications** tab.

See "[Configuring Notifications for Workflow Steps](#) " on page 55.

2. Click **New**.

The Add Notification for Step window opens.

3. Click the **Setup** tab.

4. From the **Interval** list, select one of the following:

- Select **8:00 AM Daily M-F** to have the notification sent every workday at 8:00 a.m. starting on the next workday after the notification event occurs.
- Select **Hourly M-F** to have the notification sent every hour, starting on the next available workday after the notification event occurs.
- Select **Immediate** to have the notification sent immediately.

Caution: If you select an interval other than **Immediate**, consider the following:

- Because PPM Center has an internal counter, a user can receive a notification before than the interval elapses, but not later than set.
- PPM Center does not generate a new notification with the passing of each interval. To generate a new notification, action is required.

5. Click **OK**.

6. In the Workflow Step window, click **OK**.

Sending Follow Up Notifications (Reminders)

A reminder notification can be sent if the notification event is still true after a period of time. For example, a reminder can be sent if a step is still eligible after a number of days. A reminder cannot be sent if the notification event is set to All.

Note: If you have installed and enabled the Mobility Access add-on, you can configure notifications for decision steps to be acted on by PPM Center users from their email inboxes. For details, see "[PPM Center Mobility Access](#)" on page 94.

To send follow-up notifications:

1. In the Workflow Step window, click the **Notifications** tab.

See "[Configuring Notifications for Workflow Steps](#) " on page 55.

2. Click **New**.

The Add Notification for Step window opens to the **Setup** tab.

3. In the **Options** section, configure the fields (all required) described in the following table.

Field Name	Description
Event	List of events. Select any value except for All . Options are: <ul style="list-style-type: none"> ○ Eligible ○ Specific Result ○ Specific Error
Interval	Determines when the notification is to be sent. Options are: <ul style="list-style-type: none"> ○ 8:00 AM Daily M-F ○ Hourly M-F ○ Immediate
Send Reminder?	This option is enabled (and required) after you select an item (other than (All)) from the Event list. To enable the Reminder Days field, select Yes .
Reminder Days	This field is enabled (and required) after you set Send Reminder? to Yes . Type the number of days to wait before sending a reminder notification.

4. Click **OK**.
5. In the Workflow Step window, click **OK**.

Configuring Notification Recipients

You must specify at least one recipient for a notification. The recipient can be a specific user, all members of a security group, or any email address.

To add a recipient to a notification:

1. In the Workflow Step window, click the **Notifications** tab.
2. Click **New**.

The Add Notification for Step window opens.

Note: If you have installed and enabled the Mobility Access add-on, the **Enable Decision by Email** checkbox is available. In this case, you can configure notifications for decision steps to be acted on by PPM Center users from their email inboxes. For details, see "[PPM Center Mobility Access](#)" on page 94.

3. In the **Recipients** section, click **New**.
4. In the Add New Recipient window, do one of the following:
 - To designate the recipient(s) as the primary addressee(s), click **To**.
 - To copy the recipient on the notification, click **Cc**.
 - To blind copy the recipient on the notification, click **Bcc**.
5. From the list at the top right, select one of the following methods to use to specify the notification recipient(s):
 - **Enter a Username**
 - **Enter an Email Address**
 - **Enter a Security Group**
 - **Enter a Standard Token**
 - **Enter a User Defined Token**

Selecting a value updates the value displayed in the **Recipient Type** field. For example, selecting **Enter a Security Group** changes the value to **Security Group**.

6. Provide the specific value that corresponds to the recipient type selected in [step 5](#).
 - To select one or more users to receive the notification, use the **Username** auto-complete. (You can use the **Ctrl** and **Shift** keys to select multiple users.) Each user must have an email address specified.
 - To specify a recipient by his or her email address, in the **Email Address** box, type the email address.
 - To select one or more security groups, use the **Security Group** auto-complete. (You can use the **Ctrl** and **Shift** keys to select multiple groups.) All enabled group members who have an email address in the database will receive the notification.

- To select a standard token from a list of system tokens that correspond to a user, security group, or email address, use the **Standard Token** auto-complete. The value displayed in the **Recipient Type** field indicates whether the token resolves to a user (name or ID), security group (name or ID), or email address.
- To specify a user-defined token, in the **User Defined Token** field, type any field token that corresponds to a user, security group, or email address. Then, from the **Recipient Type** list, select the item that the token resolves to (user name or ID, security group name or ID, or email address).

Tip: Use security groups or dynamic access (distributions) to specify the notification recipients whenever possible. Avoid specifying a list of users or an individual email address. If the list of users changes (as a result of a departmental or company reorganization), you would have to update it manually. If you specify a security group instead, any changes to group membership are automatically propagated throughout the workflow steps.

Use distributions to send a notification to an unnamed resource. For example, to configure the notification to be sent to the assigned user(s), specify the [REQ.ASSIGNED_TO_USERID] token as the recipient.

7. Click **OK**.

8. From the **Setup** tab, click **OK**.

The Workflow Step window opens.

9. Click **OK**.

The changes are added to the workflow.

Configuring Message Content

You can construct the notification's message to ensure that it contains the correct information in the format you want. For example, if a notification is sent to instruct you that a request requires your approval, the message instructs you to log on to PPM Center and update the request status. The notification should include a link (URL) to the referenced request.

To make them easier to configure and use, PPM Center includes:

- Pre-configured notification templates that you can use to quickly compose messages.
- Ability to compose the body of message as plain text or as HTML.

- Ability to include multiple tokens that resolve to information relevant to the recipient. For example, you can include tokens for the URL to the request approval page, information on request status and priority, and emergency contacts.

To configure the message in a notification:

1. In the Workflow Step window, click the **Notifications** tab.
2. Click **New**.

In the Add Notification for Step window, click the **Message** tab.

3. From the **Notification Template** list, select a template to use for the notification.

The **Body** field content is updated based on the selected template.

4. From the **Notification Format** list, select **HTML**, **Plain Text**, or **Multipart**.

The HTML format allows more flexibility in the look and feel of the notification. You can use any HTML editor to write and test the HTML code, and then copy and paste this content to the **Body** field.

With multipart formatting, the content type is set to "multipart/alternative" and the message body is automatically populated with HTML and plain text part markers. This enables you to send single email messages with both HTML and plain text formats so that email clients can choose which message format to use.

5. Use the **Choose** buttons to locate and select values for the **From** and **Reply to** fields.
6. In the **Subject** box, you can leave the default text (Project and Portfolio Management - Demand Management Alert), or replace it.
7. Construct the body of the message. As you do, consider using the following:
 - Token for the URL to the Request Detail page. See "[Table 3-2.Smart URL tokens](#)" on page 68 for a list of these tokens.
 - Token for the URL to the package (PPM Workbench or standard interface). See "[Table 3-2.Smart URL tokens](#)" on page 68 for a list of these distributions.
 - Tokens in the body of the message. Click **Tokens** to access the Token Builder window where you can add tokens to the message body. For information, see "[Using Tokens in the Message Body](#)" on the next page.
 - Tokens related to specific package lines or request detail fields. Add tokens that resolve information related to the individual package line or request detail field to the **Linked Token** field.

8. Click **OK**.
9. From the **Notifications** tab, click **OK**.

Using Tokens in the Message Body

You can select any of the available tokens available in the Token Builder window to include in the body of your message. However, not all tokens will resolve in all situations. As a rule, tokens associated with the request or workflow will resolve.

Note: If you include tokens of custom date fields in the body of the message, the email always displays date values in long format, even if short or medium format is specified for the date field.

Including URLs (Smart URLs)

You can include links to the items that require the recipients' attention in your notifications. You can configure notifications to include the Web address (URL) for the following entities:

- Packages
- Requests
- Request types
- Projects
- Tasks
- Workflows
- Validations
- Object types
- Environments

A user viewing email with a web-based mail reader (such as Microsoft® Outlook) can click the URL in the notification to go directly to the referenced entity.

For workflows, request types, validations, object types and environments the notification can use the entity ID or the entity name as the parameter in the URL. This will bring you to the correct window in the PPM Workbench and open the detail window for the specified entity.

"Table 3-2. Smart URL tokens" below lists the most commonly used smart URL tokens for packages and requests.

Table 3-2. Smart URL tokens

Smart URL Token	Description
PACKAGE_URL	Provides a URL that loads the package details page in the standard interface.
WORKBENCH_PACKAGE_URL	Provides a URL that loads the package window in the PPM Workbench.
REQUEST_URL	Provides a URL that loads the request details page in the standard interface.

If you use an HTML-formatted message, you must use an alternate token to provide a link to requests. (You can also use this token in plain-text formatted notifications.) The smart URL token (for HTML format) for requests is REQUEST_ID_LINK.

The REQUEST_ID_LINK token provides a link that loads the request detail page in the standard interface. This token resolves to the following format:

```
<a href="http://URL">Request Name</a>
```

In the notification, the link is displayed as a linked entry.

Configuring Timeouts for Workflow Steps

Timeouts determine how long a workflow step can remain eligible before generating an error. The **Timeout** tab in the Workflow Step window is used to set a timeout for the workflow step. See the **Timeout** field in the Workflow Step Worksheet (see "[Appendix A: Worksheets](#)" on page 264) for information about how to set the timeout.

To set timeouts for a workflow step:

1. On the PPM Workbench shortcut bar, click **Configuration > Workflows**.
2. Open a workflow.
3. In the Workflow window, click the **Layout** tab.
4. Right-click a workflow step, and then click **Edit** on the shortcut menu.
The Workflow Step window opens.
5. Click the **Timeout** tab.

6. To configure the timeout for the workflow do one of the following:
 - o To use the default timeout of the workflow step source, select **Use Workflow Step Source**. (This is the default.)
 - o To specify a token to resolve to the workflow step timeout value (instead of the default timeout of the workflow step source):
 - i. Select **Specific Value**.
 - ii. From the **Timeout Type** list, select **Token**.
 - iii. Click **Tokens**, and then use the token builder to specify the token.
 - o To specify a constant workflow step timeout value (instead of the default timeout of the workflow step source):
 - i. Select **Specific Value**.
 - ii. In the **Timeout Type** list, leave **Constant** selected.
 - iii. In the **Timeout** box, type the number of minutes, hours, days, weeks, weekdays (Monday through Friday) for the timeout to last.
 - iv. In the timeout unit list to the right, select Minutes, Hours, Days, Weeks, or Weekdays (Monday through Friday).
7. Click **Apply**.

Configuring Transitions for Workflow Steps

Transitions are the rules that logically connect workflow steps. The transitions you add to a workflow to establish the direction a process should take, based on the available results of the previous workflow step. For example, a user submits a request into a request resolution system. The first step in the workflow is Review Request. From this workflow step, the request might be Approved or Not Approved. Both Approved and Not Approved are transitions from the Review Request workflow step.

Note: You can define multiple transitions for the same result, which leads to parallel workflow branches becoming active at the same time.

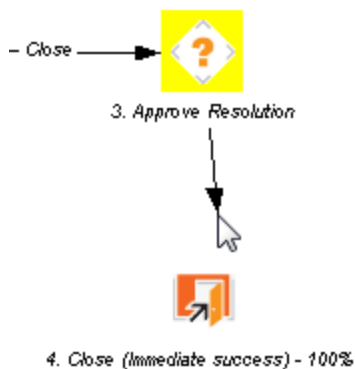
Transitions are added to a workflow after a workflow step had been dragged and dropped from the Workflow Step Source window to the **Layout** tab in the Workflow window. You can choose a transition between workflow steps based on the following workflow step results:

- **Specific result.** The specific result follows this transition. The specific results is the default workflow step result. Specific results are based on the validation specified in the workflow step source for this step. For more information about workflow step sources, see "[Configuring Workflow Components](#)" on page 208.
- **Other results.** Any other results that do not have specific transitions set follow this transition.
- **All results.** All results follow this transition.
- **Specific Event.** The specific event follows this transition. Specific events are based on the workflow step's validation. Used only for the Demand Management IT solution.
- **Specific Error.** The specific error follows this transition.
- **Other Errors.** All other errors that do not have transitions set follow this transition.
- **All Errors.** All errors follow this transition.

Adding Transitions Based on Specific Results

To add a Specific Result transition:

1. On the PPM Workbench shortcut bar, click **Configuration > Workflows**.
The Workflow Workbench opens.
2. Open a workflow.
3. In the Workflow window, click the **Layout** tab.
4. Right-click a workflow step, and then select **Add Transition** on the shortcut menu.



5. Click the destination workflow step for the transition.

On the **Layout** tab, a line with an arrowhead is displayed between the workflow steps. The Define Transition and Step Transitions windows opens.

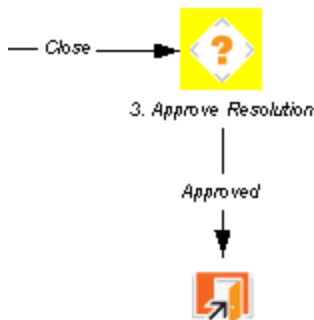
Note: The most common transition is Specific Results. For information about other transitions, see ["Adding Transitions not Based on Specific Results"](#) below.

6. From the **Specific Results** list, select the appropriate operator.
7. From the second **Specific Results** list, select the result required to transition to the destination workflow step.
8. To require the assigned resource to submit a note when acting on the "from" workflow step, select the **Require Notes on Transition** checkbox.
9. Click **OK**.

The Step Transitions window displays for the new transition.

10. Click **Apply** or **OK**.

The Layout tab displays the new transition between the "from" and "to" steps.



11. To add another validation to the transition, in the Step Transitions window, click **New**, and then add another transition value. Click **OK** to add the transition value and close the Step Transitions window. The defined transition name is added to the transition line.
12. Click **Save**.

Adding Transitions not Based on Specific Results

Transitions are added to a workflow after a workflow step had been dragged and dropped from the Workflow Step Source window to the **Layout** tab of the Workflow window. "Specific results" is the default transition value for the transition.

The possible transition values are:

- Specific results
- Other results
- All results
- Specific Events
- Specific Error
- Other Errors
- All Errors

Adding Transitions Based on Values in Fields

You can transition a request based on the value in a particular field of in the request. This can be a general field in the request header, such as **Priority**, **Assigned To**, or **Request Group**, or a custom field specified in the request or package line.

For example, if the **Priority** field for the request is set to Critical, then you might want the request to follow a different, more robust process. This is done by resolving a field token in a workflow execution step. The workflow engine evaluates the field's value at a specific step and then routes the request accordingly.

To transition a request based on a value in a field, you must:

- Configure an immediate execution workflow step.
- Configure the transition for the immediate execution workflow step.

To transition based on the value in a field:

1. On the PPM Workbench shortcut bar, click **Configuration > Workflows**.
The Workflow Workbench opens.
2. Open a workflow.
The Workflow window opens to the **Layout** tab.
3. Configure an immediate execution workflow step, as follows:
 - a. In the Workflow Step Sources window, copy an existing immediate execution workflow step.
The Execution window opens.

- b. Complete the fields in the Execution window as specified in the following table.

Field Name	Description
Workflow Scope	Requests for request tracking and resolution systems, Packages for deployment systems, Release Distribution for release systems.
Execution Type	Select Token .
Processing Type	Select Immediate . Immediate steps are automated. They execute the commands that are configured automatically and move the workflow to the next eligible step without user intervention. Note: If the previous step is an execution step and the Processing Type is set to Immediate , the status dependencies, such as Clear, will not be triggered in the current step. It requires user interaction for these types of status dependencies.
Validation	Use the auto-complete to select a validation that includes all of the possible values of the resolved token. For example, if you plan on branching based on the Priority field, use the [REQ.PRIORITY_CODE] token and the CRT - Priority - Enabled validation. The validation contains all possible values of the token.
Execution	Provide the token for the value that you would like to transition based on. To find the name of the token, below the Execution field, click Tokens . The Token Builder opens. You can use the Token Builder to help you find the token (for example [REQ.PRIORITY_CODE]), but you must manually type the name of the token in the Execution field.
Enabled	Yes

- c. Click **OK**.

- Add the new immediate execution workflow step to the workflow.
- Right-click the immediate execution workflow step, and then, from the shortcut menu, select **Add Transition**.

The menu window closes. The step remains highlighted.

- Select the destination workflow step for the transition.

A line with an arrowhead is displayed between the workflow steps. The Define Transition window and the Step Transitions window open. The Define Transition window provides several options you can use to define the transition.

- In the Define Transitions window, in the **Specific Results** field, select the transition.

8. Click **OK**.
9. In the Step Transitions window, click **OK**.
10. Click **Save**.

Adding Transitions Based on Data in Tables

You can transition based on information stored in a table. To transition using this method, use a workflow execution step with an execution type of SQL.

When transitioning from a properly configured execution step (Execution Type = SQL Statement), transition based on a specific result. The possible results are defined in the workflow step source's validation. The values in this field are determined by a SQL query of a database table.

As with any execution step, configure this transition as an immediate or a manual step.

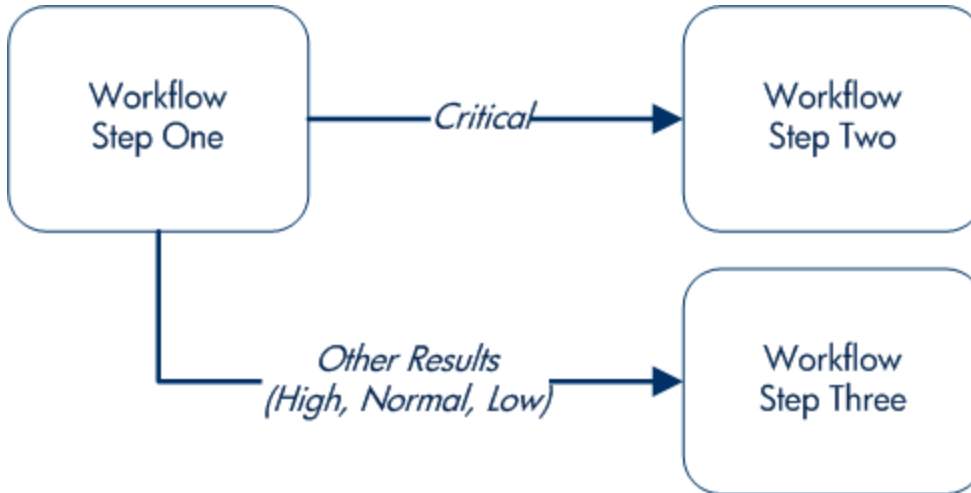
Adding Transitions Based on All But One Specific Value

You can transition based on all but one specified value. You can use Other Results when multiple transitions exit a single step. Other Results acts as the transition if none of the other explicit transition conditions are satisfied.

For example, you might want to transition all Critical requests one way and all other results (High, Normal, Low) in a different way.

To add a transition based on all but one specific value, create a transition from a workflow step based on a value in Specific Results. Create a second transition from the same workflow step. For the second transition, specify Other Results in the Define Transition window.

Figure 3-8. Transitions using other results



Adding Transitions Based on All Results

You can define a request to transition regardless of the step's actual results. For example, you may want to run a subworkflow to perform server maintenance after the on-call server contact is identified. To do this, add a transition from the Specify Contact step to the subworkflow. Because the next step in the process does not depend on the result of the step, it is appropriate to use the All Results transition. To do this, define a transition from the step, and then select **All Results**.

Consider using an All Results transition to start a sub-process. Note that you can still define transitions based on Specific Results or errors when you select **All Results**. Later, you can use an AND condition workflow step to bring the process together.

Adding Transitions Based on Specific Events

Demand Management includes an additional method for transitioning out of a workflow decision step that coincides with a demand scheduling event. Select **Specific Event** in the Define Transition window. You can then specify the specific event for the transition.

Demand Management supports the following events:

- Assignment
- Schedule Demand
- Reject Demand

An Demand Management event does not occur if:

- There is required look-ahead for the transition. The exception to this exception is when the look-ahead requires that you provide an "Assigned To" user during demand assignment.
- You do not have the correct security permissions (request type and workflow step) to transition out of the workflow step.
- The request is locked (being edited by another user).

If the scheduling, assignment, or rejecting event does not work, an error message is returned.

Adding Transitions Based on Errors

You can transition based on a specific error that occurs during an execution step. You can then branch the business process based on likely execution errors such as Timeout, Command execution, or Invalid token (see "[Table 3-3. Workflow transition errors](#)" below). As you add a transition, select the **Specific Error** option in the Define Transition window, and then select the error.

Table 3-3. Workflow transition errors

Transition Option	Meaning
Multiple Return Results	The package level subworkflow receives multiple results from package lines that traversed it.
No consensus	All users of all security groups, or users linked to the workflow step need to vote, and there is no consensus.
No recipients	None of the security groups linked to the workflow step has users linked to it. No user can act on the workflow step.
Timeout	The workflow step times out. Used for executions and decisions.
Invalid token	Invalid token used in the execution.
ORACLE error	Failed PL/SQL execution.
NULL result	No result is returned from the execution.
Invalid integer	Validation includes an invalid value in the Integer field.
Invalid date	Validation includes an invalid value in the Date field.
Command execution error	Execution engine has failed or has a problem.
Invalid Result	Execution or subworkflow has returned a result not included in the validation.
Parent closed	For wf_receive or wf_jump steps, a package line is expecting a message

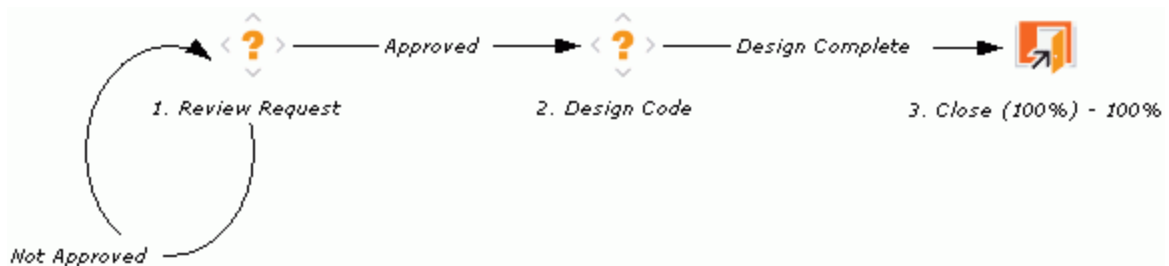
Table 3-3. Workflow transition errors, continued

Transition Option	Meaning
	from a request that is cancelled or closed.
Child closed	For wf_receive or wf_jump steps, a request is expecting a message from a package line that is cancelled or closed.
No parent	For wf_receive or wf_jump steps, a package line is expecting a message from a request that has been deleted.
No child	For wf_receive or wf_jump steps, a request is expecting a message from a package line that has been deleted.
Multiple jump results	For wf_jump steps in a package line, different result values were used to transition to the step.

Adding Transitions Back to the Same Step

You can keep the option of resetting failed execution workflow steps, rather than immediately transition along a failed path. This is often helpful when troubleshooting the execution ("Figure 3-9. Transitioning back to the same step" below).

Figure 3-9. Transitioning back to the same step



If the commands execute successfully, they follow the Success transition path. However, if the commands fail, they do not transition out of the step because no transition has been defined for the FAILED result. The user must manually select the workflow step, and then select FAILED - RETRY. The execution is re-run.

Do not use an immediate execution workflow step if a FAILED result is feeding directly back into the execution workflow step. This results in a continual execution-failure loop.

To transition a request or package line based on a value in a field, you must:

- Configure an execution workflow step.
- Configure the transition for the execution workflow step.

To transition back to the same execution step:

1. On the PPM Workbench shortcut bar, click **Configuration > Workflows**.
The Workflow Workbench opens.
2. Open a workflow.
3. In the Workflow window, click the **Layout** tab.
4. Configure an immediate execution workflow step, as follows:
 - a. In the Workflow Step Source window, copy an existing immediate execution workflow step.

In the Execution window, complete the information described in the following table.

Field Name	Description
Workflow Scope	Requests for request tracking and resolution processes, Packages for deployment processes, or Release Distributions for release processes.
Execution Type	Select Token .
Processing Type	Select Immediate . Immediate steps are automated. They execute the commands that are configured automatically and move the workflow to the next eligible step without user intervention. Note: If the previous step is an execution step and the Processing Type is set to Immediate , the status dependencies, such as Clear, will not be triggered in the current step. It requires user interaction for these types of status dependencies.
Validation	Create a validation with the following validation values. <ul style="list-style-type: none"> • Succeeded • Failed • Failed - Reset • Failed - Rejected For details on how to create a validation, see the <i>Commands, Tokens, and Validations Guide and Reference</i> .
Enabled	Yes

- b. Click **OK**.
5. Add the new execution workflow step to the workflow.
6. Right-click the immediate execution workflow step, and then select **Add Transition**.

7. Select several points near the execution workflow step, and then select the source workflow step.

The Define Transition and Step Transitions windows opens. The Define Transition window provides many options for defining the transition.

8. From the **Specific Results** list in the Define Transitions window, select the transition.

The validations in the **Specific Results** field are the validations created for the execution workflow step. For example, select **Failed - Reset**.

9. Click **OK**.

10. In the Step Transitions window, click **OK**.

The defined transition name is added to the transition line.

11. Click **Save**.

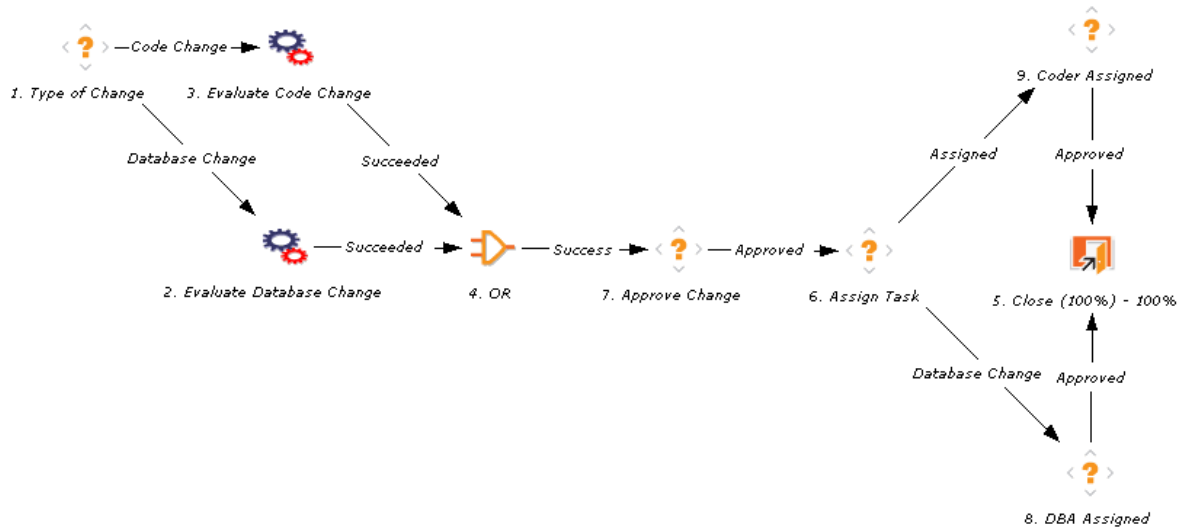
Adding Transitions Based on Previous Workflow Step Results

You can use workflow parameters to store the result of a workflow step. This value can then be used later to define a transition. The basic steps of adding a transition based on a previous workflow step result are:

1. In the Workflow window, on the **Workflow** tab, create a workflow parameter.
2. Create a token execution step to resolve the value in the workflow parameter.
3. For a workflow step, on the **Properties** tab of the Workflow Step window, in the **Workflow Parameter** field, type the workflow parameter name.

"[Figure 3-10. Add a transition based on a previous workflow step](#)" below shows an example process. One step requires the user to route the request based on the type of change (code or database). The decision made at this step is considered later in the process to correctly route rework of the specific type.

Figure 3-10. Add a transition based on a previous workflow step



To add a transition based on a previous workflow step:

1. On the PPM Workbench shortcut bar, click **Configuration > Workflows**.

The Workflow Workbench opens.

2. Open a workflow.

The Workflow window opens to the **Workflow** tab.

3. Create a workflow parameter, as follows:

- a. In the parameters section, click **Add**.

The Workflow Parameters window opens.

- b. Complete the fields.

- c. Click **OK**.

4. Click the **Layout** tab.

5. Configure an execution workflow step with a token that resolves the value in the workflow parameter.

Note: The validation used in this step must contain the same values as the validation specified in the Type of Change decision step.

- a. From the Workflow Step Source window, copy an existing execution workflow step.

The Execution window opens.

- b. Configure the workflow step.
 - c. Click **OK**.
6. Add the new execution workflow step to the workflow, as follows:
 - a. Add a workflow step to the workflow.

The Workflow Step window opens.
 - b. In the Workflow Step window, on the **Properties** tab, select the workflow parameter from the **Workflow Parameter** field.
 - c. Click **OK**.
7. Add the steps and transitions as shown in "[Figure 3-10. Add a transition based on a previous workflow step](#)" on page 79.
8. Click **OK**.

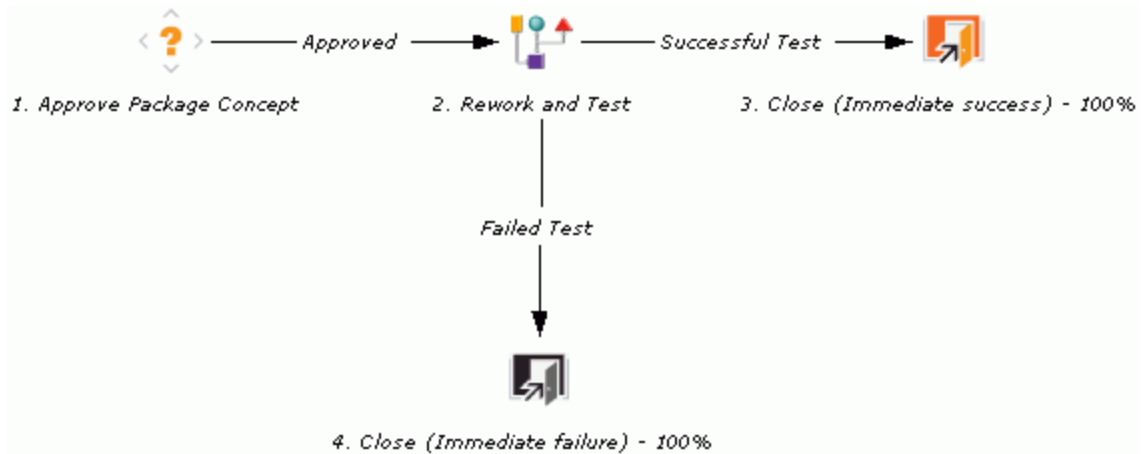
Adding Transitions To and Removing them From Subworkflows

A transition to a subworkflow step is made in the same way as a transition to any other workflow step (execution, decision, or condition). The transition is graphically represented by an arrow between the two steps. The package line or request proceeds to the first step designated in the subworkflow definition.

When the package or request reaches the subworkflow step, it follows the path defined in that subworkflow. It either closes within that workflow (at a Close step) or returns to the parent workflow.

For a package line or request to transition back to the parent workflow, the subworkflow must contain a return step. The transitions leading into the return step must match the validation established for the subworkflow step. In the following example, the transitions exiting the Rework and Test step (Successful Test and Failed Test) match the possible transitions entering the subworkflow's return step.

Figure 3-11. Transitioning to and from subworkflows



Users must verify that the validation defined for the subworkflow step is synchronized with the transitions entering the return step. The subworkflow validation is defined in the Workflow window.

Users typically define the possible transitions from the subworkflow step during the subworkflow definition.

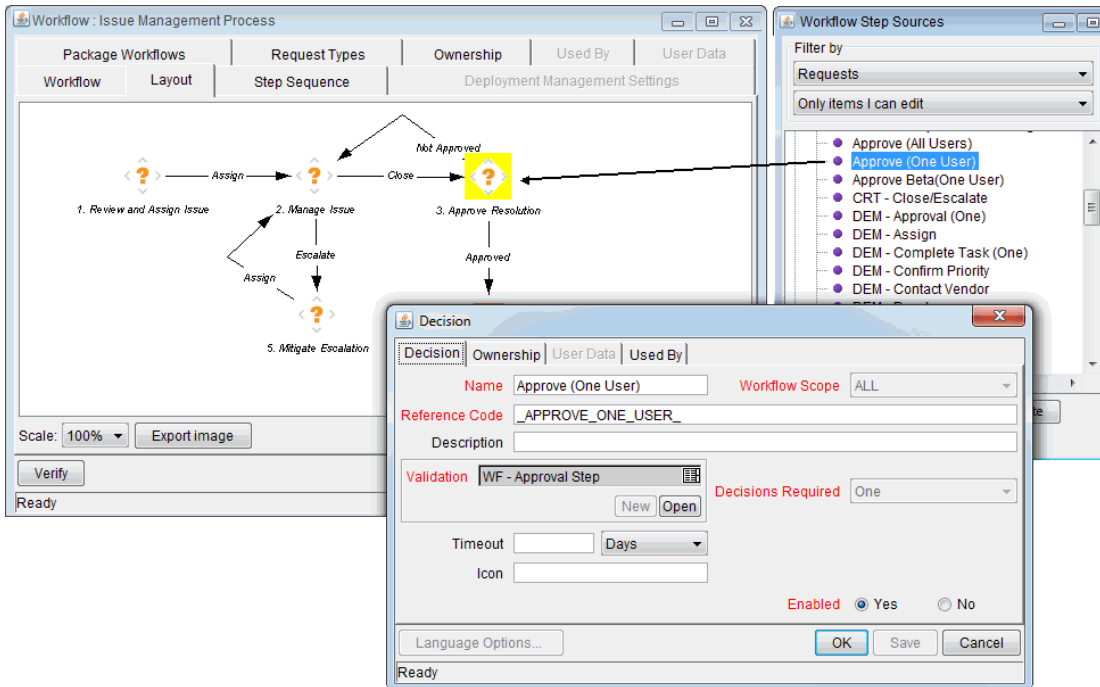
The subworkflow step validation cannot be edited if the subworkflow is used in another workflow definition. You cannot edit the subworkflow field if the subworkflow is used in another workflow definition.

Configuring Validations for Workflow Steps

Validations determine the acceptable values for fields. They maintain data integrity by ensuring that the correct information is provided in a field before that value is saved to the database. For workflow steps, validations ensure the correct transitions are associated with the correct workflow step.

Validations are defined for each workflow step found in the Workflow Step Source window from which the step is derived. "Figure 3-12. Workflow step sources and validations" below shows the Decisions window of the Approve (One User) decision workflow step validation. The validation for this workflow step validation is WF - Approval Step.

Figure 3-12. Workflow step sources and validations



If you open the WF - Approval Step validation, (on the **Decision** tab in the Decision window, click **Open**), you can see that steps derived from this workflow step source can have one of two valid values: "Approved" and "Not Approved".

After you add an Approve (One User) decision step to a workflow, and you add a transition to the next step, the Approved and Not Approve values are listed in the **Specific Results** list in the Define Transition dialog box. (See ["Adding Transitions Based on Specific Results" on page 70.](#))

Validations and Execution Type Relationships

There is a correlation between the validation and the execution type. For data-dependent transitions (token, SQL, PL/SQL), the validation must contain all possible values of the query or token resolution. Otherwise, the execution step could result in a value that is not defined for the process, and the request or package line could become stuck in a workflow step.

For most built-in workflow events and executions that run commands, the validation often includes the standard workflow results (Success or Failure). If the commands or event execute without error, the result of Success is returned, otherwise, Failure is returned.

["Table 3-4. Relationship between validation and execution types" on the next page](#) summarizes the relationship between validations and execution types.

Table 3-4. Relationship between validation and execution types

Execution Types	Validation Notes
Built-in workflow event and workflow step commands	Typically use a variation of the WF - Standard Execution Results validation (Succeeded or Failed). A few of the workflow events have specific validation requirements: <ul style="list-style-type: none"> • wf_return • wf_jump • wf_receive
PL/SQL function	Validation must contain all possible values returned by the function.
Token	Validation must contain all possible values for the token.
SQL statement	Validation must contain all possible values for the SQL query. You can use the same SQL in the validation (drop-down or auto-complete) minus the WHERE clause.

Adding Color to Workflow Steps

To make it easier to distinguish between steps in a workflow, or simply change the look and feel of a workflow in graphical view, you can apply fill color to one or more steps.

To add fill color to workflow steps:

1. In the Workflow window for an open step, click the **Layout tab**.
2. Do one of the following:
 - To select a single workflow step to fill with color, right-click the step, and then click **Edit** on the shortcut menu.
 - To select multiple steps to fill with the same color, press **Ctrl**, click all of the steps to which you want to add color, and then click **Edit** on the shortcut menu.
3. In the Workflow Step window, click the **Display Settings tab**.
4. Use the color selection features on the **Swatches** tab, the **HSB** tab, or the **RGB** tab to specify a fill color for the step.
5. Click **OK**.
6. Click anywhere on the **Layout tab**.

The selected steps fill with the color you specified.

Specifying Font Size for Workflow Layout Images

To specify a font size for displaying larger or smaller characters in workflow layout images, you can set value for the `FONT_SIZE_OF_GRAPHIC_WORKFLOW` server configuration parameter. The default value is 9.

Configuring Segregation of Duties for Workflow Steps

In some cases it may be important to ensure that multiple process approvals are made by distinct users. You can use the **Segregation of Duties** tab to configure additional restrictions on who can approve a given step, based on who has already approved previous steps.

To set segregation of duties for a workflow step:

1. From the Workflow Workbench, open a workflow.
The Workflow window opens.
2. Right-click a workflow step, and then click **Edit** on the shortcut menu.
The Workflow Step window opens.
3. Click the **Segregation of Duties** tab.
4. Click **New**.
The SOD - Source Step window opens.
5. To define a segregation source for the current workflow step, do one of the following:
 - To segregate this step from another workflow step, leave the **Workflow Step** option selected, and then select the other step from the list.
 - To prevent the user who created the request from acting on this step, select the **Workflow Instance Creation** option.
6. To add the segregation source to the **Segregation of Duties** tab, click **OK**.
7. In the Workflow Step window, click **OK**.
Changes are saved to the workflow.

Note: All users who can act on a segregated step are prevented from acting on the current workflow step.

Integrating Request Types and Workflows

This section details the ways in which workflows and request types can integrate to work together.

Integrating Workflows and Request Types through Workflow Step Properties

You can direct several aspects of request behavior through setting the following workflow step properties:

- **Request status.** Use this to specify the status to set on the parent request when a request reaches this workflow step.
- **Current % Complete.** Use this to set the value to display for the parent request's percent complete when a request reaches this workflow step.
- **Parent Assigned To User.** Use this to specify the user to which the parent request is to be assigned when a request reaches this step.
- **Parent Assigned To Group.** Use this to specify the security group to which the parent request should be assigned when the request reaches this step.

Specifying the request status (linking request status values to workflow steps) is the most important means of integrating request types and workflows. As a request progresses through the workflow, it takes on the status assigned at each workflow step.

Instructions for linking request status values to workflow steps are provided here. For instructions on how to configure other workflow step properties, see ["Configuring Properties of a Workflow Step" on page 52](#).

To assign request status values to workflow steps:

1. On the PPM Workbench shortcut bar, click **Configuration > Workflows**.

The Workflow Workbench opens.

2. Open a workflow.

The Workflow window opens to the **Workflow Layout** tab.

3. Right-click a workflow step, and then click **Edit** on the shortcut menu.

The Workflow Step window opens to the **Properties** tab.

4. Use the **Request Status** auto-complete to select the request status.

5. Repeat as necessary with all required workflow steps.

6. On the **Layout** tab, click **OK**.

As the request progresses through this workflow, it acquires the status assigned at each workflow step. Not all workflow steps require that a request status be assigned. A request type retains the last-encountered status.

Integrating Request Type Commands and Workflows

Request type commands define the execution layer within request management. While most of the resolution process for a request is analytically based, cases may arise for specific request types where system changes are required. In these cases, you can use request type commands to make these changes automatically.

Request type commands are tightly integrated with the workflow engine. The commands included in a request type are triggered at execution workflow steps.

It is important to note the following restrictions regarding command and workflow interactions:

- To execute request type commands at a particular workflow step, the workflow step must be configured as follows:
 - Workflow step must be an execution type
 - Workflow Scope = Requests
 - Execution Type = Built-in Workflow Event
 - Workflow Command = `execute_request_commands`
- When the request reaches the workflow step (with Workflow Command = `execute_request_`

commands), all commands whose conditions are satisfied are run in the order they are specified in the request type's **Command** field (on the request type's **Commands** tab).

You can configure the request type to run only certain commands at a given step. To do this, specify command conditions. For information about command conditions, see ["Command Conditions" on page 168](#).

Integrating Request and Package Workflows

Requests (Demand Management) and package workflows can be configured to work together, communicating at key points in the request and package processes. A request workflow step can jump to a preselected package workflow step. The package workflow step can receive the request workflow step, and then act on it to proceed the next step in the process.

You can also integrate packages and requests at a level that does not rely on the workflow configuration by attaching them to each entity as references. You can then set dependencies on these references to control the behavior of the request or package. For example, you might specify a request as a predecessor to a package, so that the package cannot continue until the request closes.

Two built-in workflow events facilitate this cross-product workflow integration. These workflow steps are *jump* workflow steps (`wf_jump`) and *receive* workflow steps (`wf_receive`). These steps are used at the points of interaction between workflows. Workflows can communicate through these jump and receive workflow step pairs.

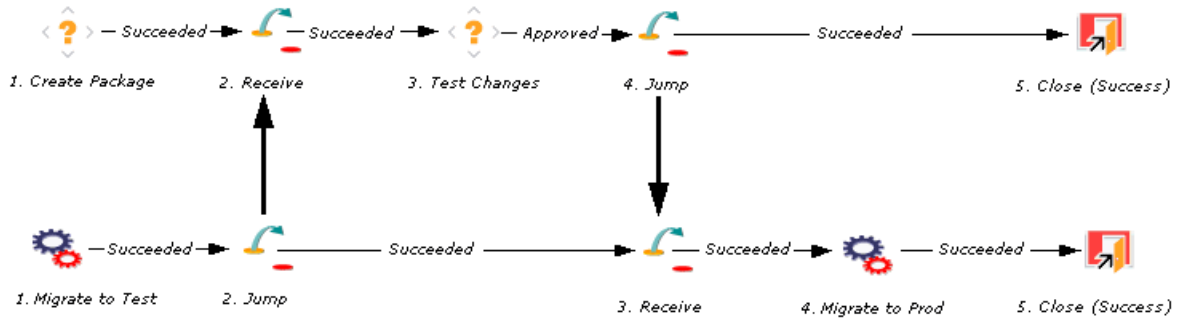
The following example illustrates how this cross-product workflow integration can be useful.

1. A request spawns a package for migrating new code to the production environment.
2. The newly-spawned package must go through an Approval step.
3. After the Approval step succeeds, the process jumps back to, and is received by, the request. The request then undergoes more testing and changes in the QA Environment.
4. After successfully completing the QA Test step, the process jumps from the request and is received by the package.
5. Because the QA Test step was successful, the process can now migrate the code changes to the production environment.

["Figure 3-13. Jump and receive workflow step pairs" below](#) illustrates this process.

Figure 3-13. Jump and receive workflow step pairs

Request Workflow



Package Workflow

The jump and receive workflow step pair must be carefully coordinated. Each jump workflow step must have an associated receive workflow step, linked together by a common jump and receive workflow step label defined in the Workflow Step window. The transition values used to enter and exit the jump and receive workflow steps must also be coordinated.

To establish communication between request and package workflows:

1. Set up the **WF - Jump/Receive Step Labels** validation for use in the Workflow Step window.
This validation is used to join a jump and receive workflow step pair. The selected **WF - Jump/Receive Step Labels** must match in the paired jump and receive Workflow Step windows.
See ["Step 1. Setting Up WF - Jump/Receive Step Label Validations "](#) below.
2. Use the **wf_jump Built-in Workflow Event** to create a jump workflow step.
See ["Step 2. Generating Jump Step Sources"](#) on the next page.
3. Use the **wf_receive Built in Workflow Event** to create a receive workflow step.
See ["Step 3. Generating Receive Step Sources"](#) on page 91.
4. Verify that both the jump and receive workflow steps specify the same entry in the **WF - Jump/Receive Step Labels** field and that the entry matches the transition value between the two steps.
See ["Step 4. Including Jump and Receive Workflow Steps in Workflows"](#) on page 92.

Step 1. Setting Up WF - Jump/Receive Step Label Validations

To set up the WF - Jump/Receive Step Labels validation:

1. On the PPM Workbench shortcut bar, click **Configuration > Validations**.
The Validation Workbench opens.
2. Click **List**, and then open the **WF - Jump/Receive Step Labels** validation.
3. To define a new validation value to use to link two workflows, in the Validation window, click **New**.
The Add Validation Value window opens.
4. Type the validation code, its meaning, and optionally, a description.
5. Click **OK**.
6. In the Validation window, click **Ownership**.
7. In the **Ownership** window, specify the security groups whose members can edit this validation.
8. Click **OK**.

The new validation value is now included in the **Jump/Receive Step Label** field in the Workflow Step window.

For more information about how to configure validations, see the *Commands, Tokens, and Validations Guide and Reference*.

Step 2. Generating Jump Step Sources

To create a jump step using the wf_jump built-in workflow event:

1. On the PPM Workbench shortcut bar, click **Configuration > Workflows**.
The Workflow Workbench opens.
2. Open a workflow.
The Workflow and Workflow Step Sources windows open.
3. In the Workflow Step Sources window, select the **Executions** folder, and then click **New**.
The Execution window opens.
4. In the **Name** field, type a name for the jump step.
5. From the **Workflow Scope** list, select **Requests**.

Note: Package-level subworkflows and Release Distribution workflows cannot include jump and receive steps.

6. In the **Execution Type** list, leave **Built-in Workflow Event** selected.
7. From the **Workflow Event** list, select **wf_jump**.
8. Use the **Validation** auto-complete to select a validation to use to transition out of this workflow step.

Note: The validation values that exit the jump workflow step must match the possible validation values used to enter the jump workflow step.

9. Provide all required information and any optional information you want to include.
10. Click the **Ownership** tab.
11. Specify the security groups whose members can edit this execution workflow step.
12. Click **OK**.

The **Executions** folder in the Workflow Step Sources window now includes the new jump workflow step.

This workflow step can now be used in any new or existing workflow within the defined workflow step scope. Keep in mind that every jump step must have a paired receive step in another workflow.

Step 3. Generating Receive Step Sources

To create a receive step using the wf_receive built-in workflow event:

1. On the PPM Workbench shortcut bar, click **Configuration > Workflows**.
The Workflow Workbench opens.
2. Open a workflow.
The Workflow and Workflow Step Sources windows open.
3. In the Workflow Step Sources window, select the **Executions** folder, and then click **New**.
The Execution window opens.
4. In the **Name** field, type a name for the new execution step.
5. From the **Workflow Scope** list, select either **Packages** or **Requests**, depending on how you plan to apply the workflow.
6. PPM Center automatically populates the **Reference Code** box based on the value you entered in

the **Name** box. You can accept the default, or replace it with a different reference code to be used to reference this receive step.

7. In the **Execution Type** list, leave **Built-in Workflow Event** selected.
8. From the **Workflow Event** list, select **wf_receive**.
9. Use the **Validation** auto-complete to select a validation to use to transition out of this workflow step.

Note: The validation values that exit the receive workflow step must match the possible validation values used to enter and exit the jump workflow step.

10. Provide all required information and any optional information you want included.
11. Click the **Ownership** tab.
12. Select the security groups whose members can edit this execution workflow step.
13. Click **OK**.

The **Executions** folder in the Workflow Step Sources window now includes the new receive workflow step.

This workflow step can be used in any new or existing workflow within the defined workflow scope. Keep in mind that every receive step must correspond to a jump step in another workflow.

Step 4. Including Jump and Receive Workflow Steps in Workflows

After you create the jump workflow and receive workflow steps ("[Step 2. Generating Jump Step Sources](#)" on page 90 and "[Step 3. Generating Receive Step Sources](#)" on the previous page), you can now include them in a workflow. The **Jump/Receive Step Label** field is the key communication link between separate workflows. The communicating jump and receive workflow steps must have a matching **Jump/Receive Step Label** field entry. The **Jump/Receive Step Label** field entry must be unique for any given jump and receive workflow step pair.

To include a jump and a receive workflow step pair in a workflow:

1. On the PPM Workbench shortcut bar, click **Configuration > Workflows**.

The Workflow Workbench opens.

2. Open a workflow.

The Workflow and Workflow Step Sources windows open.

3. In the Workflow Step Sources window:

- a. Expand the **Executions** folder.
- b. Drag your jump workflow step to the **Layout** tab in the Workflow window.

The Workflow Step window opens.

- c. From the **Jump/Receive Step Label** list, select an item.

For example, **Migrate to Production**. This item must be the same for a paired jump and receive workflow step. The **Jump/Receive Step Label** field is the key communication link between separate workflows. The communicating jump and receive workflow steps must have a matching Jump/Receive Step Label field. The Jump/Receive Step Label field must be unique for any jump and receive pair.

- d. Provide any additional workflow step information you want included, and then click **OK**.

4. In the Workflow Step Sources window:

- a. Expand the **Executions** folder.
- b. Drag your receive workflow step to the **Layout** tab in the workflow window.

The Workflow Step window opens.

- c. In the **Jump/Receive Step Label** field, select an item.

For example, **Migrate to Production**. This item must be the same for a paired jump and receive workflow step. The **Jump/Receive Step Label** field is the key communication link between separate workflows. The communicating jump and receive workflow steps must have matching jump/receive step labels. The **Jump/Receive Step Label** field value must be unique for any jump and receive pair.

- d. Provide any additional workflow step information you want included, and then click **OK**.

5. Add a transition between the jump workflow step and the receive workflow step.

The transition must be set to the **Jump/Receive Step Label** field value you selected (for example **Migrate to Production**).

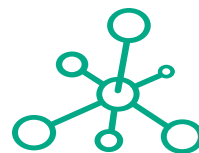
6. To save and close the workflow, click **Save**.



Chapter 4: PPM Center Mobility Access

- "Overview of PPM Center Mobility Access" on the next page
- "Mobility Access Best Practices" on page 97
- "Mobility Access Limitations" on page 97
- "Mobility Access Deployment" on page 98

- "Mobility Access Configuration" on page 99



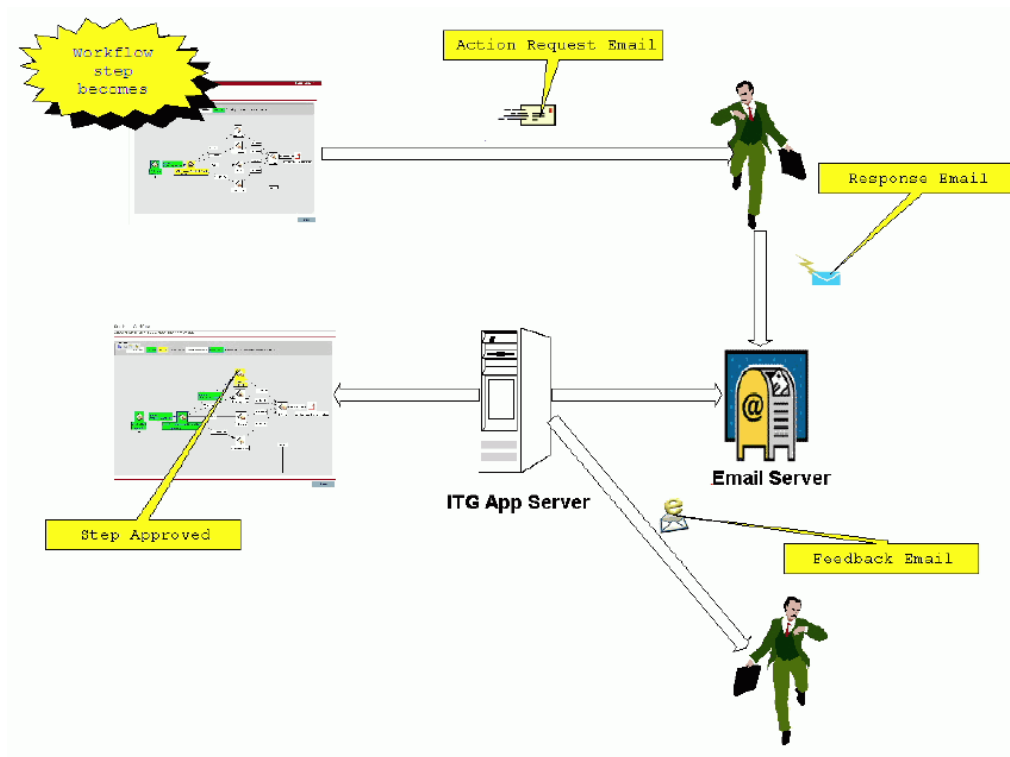
**Hewlett Packard
Enterprise**

Overview of PPM Center Mobility Access

Project and Portfolio Management Center Mobility Access (Mobility Access) is an HPE smart add-on that enables PPM Center users to process approval workflow steps from desktop email or any PDA device. Resources who are working outside of an office or without VPN access can act on approval workflow steps without first logging on to PPM Center.

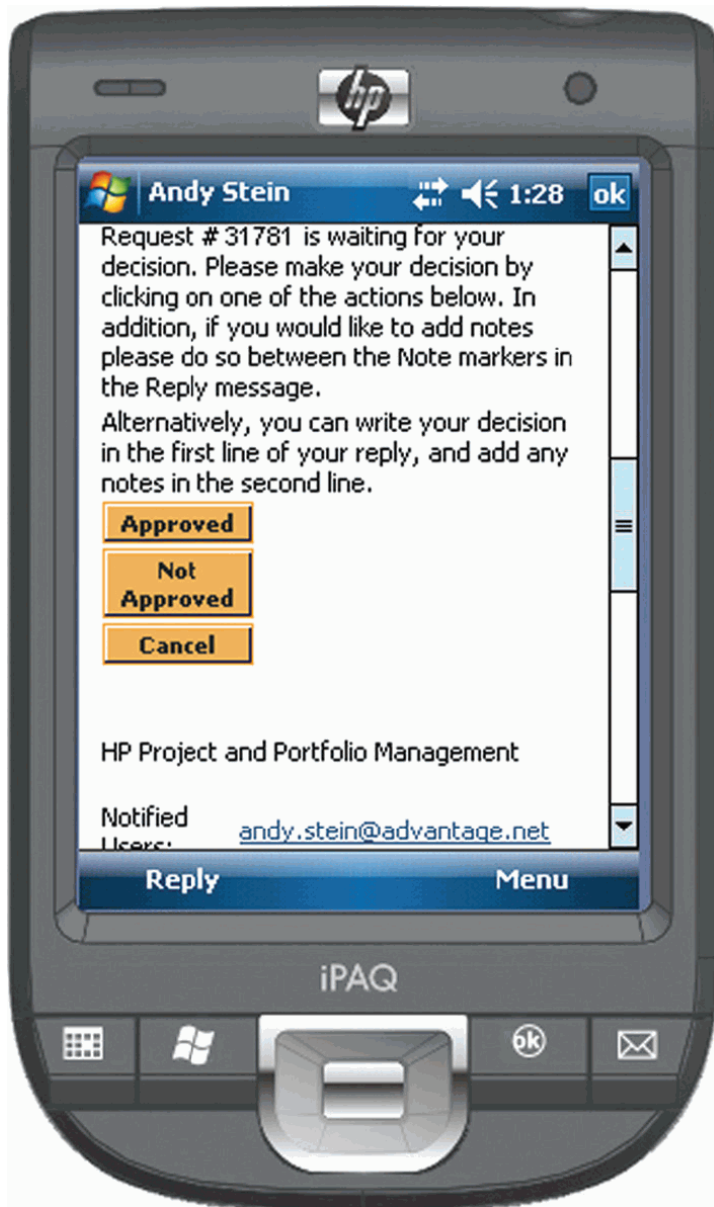
Note: Mobility Access Service notifications are available only for workflow decision steps with lookup validations of both the Demand Management and Deployment Management modules. Mobility Access can also be used with project request workflows.

Users receive email alerts to inform them that a workflow approval step is pending and requires their action. The user can act on the workflow step using the buttons available in the email. The user can also create a note to append to the PPM Center entity (request or package) being processed.



"Figure 4-1. Mobility Access alert about an approval workflow step pending for a PPM Center request" below shows a Mobility Access email alert displayed in a PPM Center user's PDA device.

Figure 4-1. Mobility Access alert about an approval workflow step pending for a PPM Center request



The Mobility Access background service running on the PPM Server periodically connects to the email account (at the frequency you configure for the service) and reads user response emails. The service verifies that the email response is from the email address to which the notification was sent, and that the email address is for a valid PPM Center user.

Finally, the Mobility Access service parses the response email and applies the selected action to the workflow step (if the user is authorized to act on the step). You can also configure the system to send feedback email to notify the user of the success or failure of the selected action.

Mobility Access Best Practices

HPE recommends the following best practices in for Mobility Access use:

- Use Mobility Access Service for only a small subset of important workflow decision steps (for example, steps that require executive approval by email only). For typical day-to-day workflow actions, use the PPM Center standard interface.
- Overall performance of the Mobility Access Service depends on the performance of the email server and the network bandwidth between the email server and the PPM Server. For best performance, place the PPM Server and email server on the same local area network (LAN). The Mobility Access service can process about 1,000 emails per hour in a medium-scale deployment that is set up according to HPE recommendations based on test conditions.
- Before you deploy Mobility Access in a production environment, deploy it in a staging environment so that you can assess system setup and configuration.
- Install Multilingual User Interface (MLU) in PPM Center if you plan to use Mobility Access service in non-English-speaking regions. As a result, MLU can format regional sensitive data such as date, currency, and number.

Mobility Access Limitations

The Mobility Access add-on is subject to the following limitations:

- Mobility Access does not support delegation of approvals. If a user delegates the approval of a step to a different user and forwards the notification received from the system to the delegate, the delegated user cannot act on the workflow step on behalf of the original recipient.
- Mobility Access Service notifications are available only for workflow decision steps with lookup validations of both the Demand Management and Deployment Management modules. Mobility Access can also be used with project request workflows.
- Email addresses sharing between PPM Center users is not supported. Users selected to receive email approval notifications must not share their PPM Center email address with another PPM Center user.

Mobility Access in a Multilingual User Interface

Context

If you plan to use Mobility Access in a PPM Center system that supports multiple languages, consider the following:

- To use Mobility Access with languages other than English, you must provide translated content. For information about the translation tools that PPM Center provides and instructions on how to use them, see the *Multilingual User Interface Guide*.
- The notification locale must match the Windows regional settings on the recipient's system. If the notification locale is different than the user's locale, the notification message display is corrupted, and the recipient cannot respond to the notification.

You can configure the user's locale using the regional and language options settings on Control Panel. For example, for a Windows Server 2003-based computer, on Control Panel, select **Regional and Language Options**. In the Regional and Language Options window, click the **Advanced** tab, and then, in the **Language for non-Unicode programs** section, select a language from the list.

Mobility Access Deployment

To deploy the Mobility Access feature on your PPM Center instance, you need to download and install the software, and then enable and schedule the Mobility Access service in PPM Center. This section provides instructions on how to perform both of these tasks.

Note: Before you deploy the Mobility Access, you must purchase a Mobility Access *site* license key for your PPM Center instance. Because this is a site license rather than a user license, you are not required to assign the license to PPM Center users. Simply make sure that the license key is in the `license.conf` file in the `<PPM_Home>/conf` directory.

Installing the Mobility Access

To install the Mobility Access:

1. Stop the PPM Server.
2. Make sure that your `license.conf` file contains your Mobility Access license key.

License key example: `com.kintana.core.server.MOBILITY_ACCESS_LICENSE_KEY=a54x051fmm67e64325839055039e395c8`

3. Download the `ppm-940-mobility-access.jar` file from the My software updates portal (<https://h20575.www2.hp.com/usbportal/softwareupdate.do>)

Note: The `ppm-940-mobility-access.jar` file is available on the HPE Software Support site only if you have a Mobility Access license key.

4. Navigate to the `<PPM_Home>/bin` directory, and then run the following command.

```
./kDeploy.sh -mobility-access
```

After the `kDeploy.sh` script run finishes, a message is displayed to advise you that the installation was successful.

5. Start the PPM Server.

Mobility Access Configuration

Mobility Access configuration involves the following tasks:

- Enable the Mobility Access Service
- Configure an email server
- Configure user-defined markers
- Configure Mobility Access Service logging

The following sections provide instructions on how to perform each of these tasks.

Enabling and Scheduling the Mobility Access Service

To enable and schedule the Mobility Access service:

1. Log on to PPM Center.
2. On the **Open** menu, click **Administration > Schedule Services**.

The Schedule Services page lists all of the available services, and shows the typical load each service manages, whether the service is enabled, the type of expression used to schedule the service, and the current run schedule.

3. Click the table row that displays the Mobility Access Service.
4. From the **Status** list, select **Enabled**.
5. To select the type of expression to use to schedule the service, from the **Schedule Type** list, select either **Simple** or **Cron**. (The default is **Simple**, and is set to 30 seconds.)

Caution: If you use a cron expression to schedule a service, keep in mind that cron expressions take into account the `TIME_ZONE` parameter setting for the PPM Server on which the service runs. In a server cluster environment, servers can be running on machines located in different time zones.

6. In the **Schedule** column, do one of the following:
 - To schedule the service using a simple expression, type a number in the first field and, from the list on the right, select the time unit (**seconds**, **minutes**, or **hours**.)
 - To schedule the service using a cron expression, type the expression in the text field. For detailed help on how to compose a cron expression, under the **Schedule Type** column heading, select the Help icon (?).

Note: If you use a cron expression to schedule the service, keep in mind that the value you type in the **Schedule** field cannot exceed 40 characters.

7. Click **Save**.

Your changes take effect immediately after you save them. There is no need to restart the PPM Server. You can now configure decision workflow step notifications for mobile access, and PPM Center users who receive such notifications can now access decision step approval functionality from their PDAs and email inboxes.

Mobility Access Server-Side Configuration Settings

Installation adds the `mobility_access.xml` file to the `<PPM_Home>/conf` directory. The `mobility_access.xml` file stores the server-side configuration parameter settings for the Mobility Access Service.

Configuring the Email Server

After you enable the Mobility Access Service, you must configure an email account used to fetch and process the emails. This email account must meet the following requirements:

- The account must not be shared with any other process or used for any other purpose.
- The account must not be configured as the email address for a PPM Center user.
- The account must be configured to receive notification response email messages no larger than 1 MB.

Note: Any email sent to the configured account that is not a notification response email for the Mobility Access Service is considered SPAM and is deleted.

The `mobility_access.xml` file contents are as follows:

```
<email_service_config>
  <email_servers>
    <inbound_email_server>
      <hostname>imap.mail.hp.com</hostname>
      <enabled>true</enabled>
      <protocol>imap</protocol>
      <email_address>ppm750_email_service@hp.com</email_address>
      <email_account>ppm750_email_service </email_account>
      <password><![CDATA[#!#3E7i:C/Vp}lhSN41)
L~YLhFk-:w5tzJpR1ua~q`ekTD^ChnJ<>4UNxc51x7ip`6x~4`qZlx^3_18EAhzJtZ(b90/RE&+
{A,68156ApEcFqQpv9Kg{Rfx^&~ep_VtLPC(:nkk=:<A85x91v(A*(mk3{$kJcbrjlk@k)L{`|8$)
<KPLxI@ 2R13^sL1p)i7#!#]]></password>
      <default_folder>INBOX</default_folder>
      <mail_archive_folder>PPM_PROCESSED_MAILS</mail_archive_folder>
      <archive_messages>Y</archive_messages>
      <send_success_feedback>N</send_success_feedback>
      <send_failure_feedback>Y</send_failure_feedback>
      <send_not_applicable_feedback>N</send_not_applicable_feedback>
      <days_to_retain_messages>2</days_to_retain_messages>
      <action_processor><value>com.kintana.sops.emailprocessor
.server.WorkflowActionProcessor</value></action_processor>
      <notes_logging>ONLY_EMAIL_MESSAGE</notes_logging>
      <package_lines_bulk_approval>Y</package_lines_bulk_approval>
    </inbound_email_server>
  </email_servers>
</email_service_config>
```

To configure the email server:

1. Navigate to the `<PPM_Home>/conf` directory and open the `mobility_access.xml` file in a text editor.
2. Provide the information described in the following table, which lists descriptions of the Mobility Access Configuration parameters in the `mobility_access.xml` file, along with their valid values and default values.

Note: You must provide values for the parameter names that are marked with an asterisk in the table.

Parameter Name	Default	Description
*hostname		Hostname of email server. Example: <code>imap.mail.hp.com</code> You can also specify an IP address for this parameter.
*enabled	true	If set to true, indicates that this email server is enabled for processing emails. Emails are fetched and processed only from the servers for which this parameter is set to true.
*protocol	imap	Mail protocol of the email server. Only IMAP and POP3 protocols are supported. Valid values are as follows: <ul style="list-style-type: none"> ◦ <code>imap</code> ◦ <code>pop3</code> ◦ <code>imaps</code> ◦ <code>pop3s</code>
*email_address		Email address from which to fetch and process the emails. Example: <code>ppm800_email_service@hp.com</code> Note: Make sure that no PPM Center users account specifies this address as the user email address.
*email_account		Email account from which to fetch and process the emails. The value you specify depends on your server and could be just the account name (with domain name) or the email address. Examples: <code>ppm800_email_service</code>

Parameter Name	Default	Description
*Required		AMERICAS\ppm800_email_service ppm800_email_service@hp.com
password		<p>Both plain text password and passwords encrypted using kEncrypt.sh are supported. Passwords must be enclosed between <![CDATA[and]]> xml tags to escape special characters.</p> <p>Note: All encrypted passwords must be enclosed between #!#.</p> <p>Plain text password example:</p> <pre><password><![CDATA[Welcome123]]> </password></pre> <p>Encrypted password example:</p> <pre><password><![CDATA[#!#3E7i:C/Vp}1hSN41) L~YLhFk-:w5tzJpR1ua~q `ekTD^ChnJ<>4UNxc51x7ip`6x~4`qZ1x^3_18EAhzJtZ(b90/RE&+ {A, 68156ApEcFqQpv9Kg{Rfx^&~ep_VtLPC(:nkk=:<A85x91v(A (mk3 {\$kJcbrjlk@k)L{` 8\$)<KPLxI@ 2R13^sL1p)i7#!#}] ></password></pre>
mail_archive_folder		<p>Determines the name of the folder in which incoming mails are to be stored after processing. (Applies only to email servers that support IMAP protocols.)</p> <p>Example: PPM_PROCESSED_MAILS</p> <p>Note: If a folder does not exist, the Mobility Access Service creates the folder.</p>
*archive_messages	Y	<p>Applies to email servers using the IMAP protocol. If set to Y, all the incoming emails and outgoing feedback emails are logged into the PPM_EMAIL_PROCESSED_MSGS table.</p> <p>If the email server supports the IMAP protocol, and if the mail_archive_folder parameter is set to a valid value, then all the incoming emails are moved to the archive folder after processing.</p>
*days_to_retain_messages	180	<p>Applies to email servers using the IMAP protocol. Number of days from the mail received date that email messages are to remain in the PPM_EMAIL_PROCESSED_MSGS table and mail_archive_folder.</p>
*action_processor		<p>Intended for possible future use, this parameter is set to the following value and must NEVER be changed:</p>

Parameter Name	Default	Description
*Required	Default	Description
		com.kintana.wf.mobilityaccess. server.WorkflowActionProcessor
Action Processor Configurations		
notes_ logging	ONLY_ EMAIL_ MESSAG E	Determines what gets saved in the notes for requests and packages. The value can be one of the following: <ul style="list-style-type: none"> ONLY_EMAIL_MESSAGE Only the email message is saved in the notes. HEADERS_AND_EMAIL_MESSAGE Email message and headers are saved in the notes.
*send_ success_ feedback	N	Determines whether a feedback message is to be sent to the user if the selected workflow action was completed successfully.
*send_ failure_ feedback	Y	Determines whether a feedback message is to be sent to the user the selected workflow action failed.
*send_ not_ applicabl e_ feedback	N	Determines whether a feedback message is to be sent to the user if the selected workflow action was not applicable.
mobility_ access_ hide_ initial_ message	false	Controls whether to hide or display the initial text in an Mobility Access email notification. Setting the parameter to true hides the initial text.

3. Save, and then close the `mobility_access.xml` file.

Configuring User-Defined Markers

Any markers that PPM Center users define in email responses are added to the named `mobility_access.properties` resource file, which is located in the `<PPM_Home>/conf/custom_resources/mobility_access` directory. You can edit this file to customize these markers.

Note: If your PPM Center instance supports multiple languages, note that, while HPE supplies translations for the message content markers it delivers. If you change these marker definitions, then you must create and maintain the translations for them.

The `mobility_access.properties` file contains the following code:

```
#A customer-updatable resource file for Mobility Access markers
#####
### Non-Translatable Resources ###
#####
subjectResponseBeginSuffix=:
subjectResponseEnd=;
notesMarkerPrefix=<
notesMarkerSuffix=>
controlDataResponseBeginSuffix=:
controlDataResponseEnd=~::~
#####
### Translatable Resources ###
#####
subjectResponseBegin=User Action
userNotesResponseBegin=Notes Begin
userNotesResponseEnd=Notes End
controlDataResponseBegin=PPM Reference
```

Adding Mobility Access to Decision Step Notifications

After you deploy Mobility Access, you can configure the decision steps in your workflows to be acted upon by assigned recipients from their email inboxes and PDAs.

Caution: If you create a validation for a workflow step source, keep in mind that the Mobility Access feature does not support the use of single quote (`) characters in validation names.

To configure a notification with mobile access on a decision workflow step:

1. On the PPM Workbench shortcut bar, click **Configuration > Workflows**.
2. Open a workflow.
3. On the **Layout** tab in the Workflow window, double-click a decision step.
4. In the Workflow Step window, click the **Notifications** tab.

5. Click **New**.

The Add Notification for step window opens to the **Setup** tab.

6. Select the **Enable Decision by Email** checkbox.
7. From the **Event** list, select **Eligible**.

Caution: HPE recommends that you *not* specify **All**, **Specific Result**, or **Specific Error** as the triggering event.

Note: Selecting the **Enable Decision by Email** checkbox defaults the **Interval** list value to **Immediate**, which is the only interval that Mobility Access supports.

8. In the **Recipients** section, do one of the following:
 - Click **New**, and then use the Add New Recipient window to select the notification recipients (users, security groups, or tokens).
 - To specify the users or groups listed on the **Security** tab for the step as notification recipients, click **Copy Security**.

You can only specify PPM Center users who have permission to act on the workflow step as notification recipients.

Note: If you have installed and enabled the Mobility Access add-on, the **Enable Decision by Email** checkbox is available. In this case, you can configure notifications for decision steps to be acted on by PPM Center users from their email inboxes. For details, see "[PPM Center Mobility Access](#)" on page 94.

9. Click the **Message** tab and configure the body of the notification (for details, see "[Configuring Message Content](#)" on page 65).
10. Click **OK**.

Once an entity using the workflow advances to the workflow step for which you have set up the notification, the specified recipients can act on the step from their email inboxes or PDAs.

Configuring Mobility Access Service Logging

The Mobility Access Service writes to the `mobility_access_log.txt` file, which is located in the `<PPM_Home>/server/<PPM_ServerName>/log` directory. If you encounter problems related to Mobility Access, you can use this log file to help you troubleshoot.

Mobility Access Service logging is defined in the `logging.conf` file, which is located in the `<PPM_Home>/conf` directory. The file specifies that any error related to Mobility Access Service is to be logged in the `mobility_access_log.txt` file.

To turn on debugging, open the `logging.conf` file and change only the first line in the Mobility Access Service Logging configuration to the following:

```
# Mobility Access logging
log4j.logger.com.kintana.wf.emailprocessor=DEBUG, MOBILITY_ACCESS_LOG
```

Caution: To ensure that only valid actions are taken based on email responses, each email approval message contains a system-generated key that PPM Center uses to validate the response. Any user who responds to a PPM email approval message must include this key in the response.

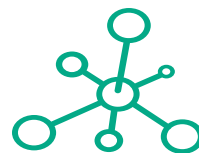
However, note that HPE does not guarantee non-repudiation for email approvals, and recommends that you check for compliance with internal and other applicable policies for non-repudiation in approval processes within your enterprise before you enable this feature in PPM Center.

To reduce the risk of accepting approval messages from unauthorized users, make sure that email messages are encrypted both in transit and at rest. To further reduce the risk of exposing the system-generated keys in PPM Center email approval messages, make sure that the user accounts of potential approvers are on an internal enterprise mail server, and that only these email accounts are associated with the corresponding PPM Center users. These messages must not be forwarded to external mail servers, and any mobile devices used by potential approvers must directly exchange messages with the enterprise mail server using secure channels. These risk mitigation suggestions still cannot guarantee non-repudiation of approvers.



Chapter 5: Configuring Request Types and Request Header Types

Overview of Request



Hewlett Packard
Enterprise

Types

Requests are the fundamental work unit of Demand Management. Users create requests and submit them along a resolution process defined in the workflow. The request details page (see "[Figure 5-1. Details page for a DEM - Application Enhancement type request](#)" below) contains the form used to capture all of the information required to complete a specific business process.

Each request is derived from an associated *request type*, which determines the fields the request includes and much of the request-specific logic. PPM Center includes predefined request types such as Bug, Project Scope Change, and Enhancement request types. These serve as examples that you can use as starting points on which to configure the processes required by your business. You can also create and configure your own request types from scratch from the Request Type Workbench.

Figure 5-1. Details page for a DEM - Application Enhancement type request

Create New DEM - Application Enhancement

Submit Cancel

Expand All | Collapse All

Summary

Requested By:
Admin User

Request Status:
Unreleased

Workflow:
DEM - Enhancement Request Process

Assigned To:

Assigned Group:

Requestor Department:

Priority:
High

Application:
HR Application

Description:
Create a new module for onboarding in Singapore office.

Work Item Fields

Scheduled Start Date:

Scheduled Finish Date:

Scheduled Duration:

Scheduled Effort:

Workload?
 Yes No

Role:

Actual Start Date:

Actual Finish Date:

Actual Duration:

Actual Effort:

Workload Category:

Enhancement Details

Enhancement Name:

Detailed Description:

New Enhancement:
 Yes No

Suite:

Requested By:

Requestor Location:

Business Initiative:

Analysis

Estimated Completion Date:

Demand Management SLA Fields

SLA Level:

SLA Violation Date:

Service Requested Date:
January 25, 2013

Service Satisfied Date:

Demand Management Scheduling Fields

Estimated Start Date:

Estimated Effort:

Reject Date:

Demand Satisfied Date:

+ Resources (No Resources Exist)

+ Notes

+ References

Request Type Components and Configuration Options

Request types have several categories of configurable components. All of these can be viewed and configured from the Request Type window.

The main components of a request type are as follows:

- **General information.** General information includes basic request type data such as the name and request type category. For detailed information on these component and how to configure them, see ["Configuring General Information for Request Types" on page 116.](#)
- **Fields.** Every request type is associated with a *request header type* that defines a set of fields (such as **Priority**, **Submitted By**, and **Assigned To**) for the request type. You can use the **Fields** tab to view these default request header type fields and to create additional fields for the request type. For detailed information on request type fields and how to configure them, see ["Creating and Configuring Request Type Fields" on page 118.](#)
- **Layout.** After you create all of the fields for a request type, you can use the **Layout** tab to configure their display on request details pages. For information about how to configure the field layout for a request type, see ["Configuring Layouts for Request Types" on page 127.](#)
- **Display Columns.** Use this tab to configure the request type columns that are available for display in portlets. For instructions, see ["Configuring Display Columns for Request Types " on page 131.](#)
- **Request Status.** A request usually acquires different status values as it progresses along its workflow. You can set up these status values to drive field behavior, by linking workflow processes to specific information in the request. For information about how to work with request statuses, see ["Configuring Request Statuses for Request Types " on page 133.](#)
- **Status Dependencies.** The different status values that a request acquires as it progresses toward resolution can be used to control field behavior. For example, a read-only field can become editable following changes that affect request status. For more information, see ["Configuring Request Field Status Dependencies" on page 136.](#)
- **Rules.** Use this tab to configure request rules that can drive simple or complex interactions between fields in a request. For example, you can configure a rule to set up the automatic population of fields based on dependencies. For more information, see ["Request Type Rules" on page 139.](#)

- **Commands.** Use this tab to control certain behavior of request type fields. At specific workflow execution steps in a request tracking and resolution process, you can select to run the commands stored in the request type. These commands can then manipulate the data inside a request type field.

For example, you can construct a command to evaluate several parameters, and then set a default value for the field based on those parameters. This provides an advantage over the defaulting features on the **Field** tab, which can only default based on a single parameter stored in the same request type. For detailed information on how to set up commands for request types, see ["Configuring Commands for Request Types " on page 165.](#)

- **Sub-Types.** Use this tab to create valid subtypes for the request type. For example, a defect request type might have hardware, software, and documentation subtypes. For more information, see ["Configuring Sub-Types for Request Types " on page 169.](#)
- **Workflows.** Use this tab to specify which workflows can be used with a request type. For instructions, see ["Configuring Request Types to Work with Workflows" on page 171.](#)
- **User Access.** Use this tab to set up rules that govern which users can access requests of this type. (The set of users who can access a request is referred to as *request participants*.) You can give the participants varying levels of access rights to requests. For details, see ["Configuring Participants for Requests" on page 172.](#)
- **Notifications.** Use this tab to configure emails to be sent if specific fields in the request type are completed. For details, see ["Configuring Notifications for Request Types" on page 187.](#)
- **User Data.** This tab displays information captured by user data fields, which are custom fields that you create to capture specific information that is not captured by standard PPM Center fields. If no user data fields are defined for the request type, the **User Data** tab is disabled. For detailed information on user data and how to create user data fields, see ["Configuring User Data" on page 245.](#)
- **Ownership.** Use this tab to specify who can edit the request type configuration. For details, see ["Configuring Ownerships of Request Types" on page 194.](#)
- **Resources.** Use this tab to:
 - Enable tracking of actuals in Time Management.

Note: Actual values, or *actuals*, represent the amount of time (in hours) that a resource has worked on an activity or request, the amount of time that is estimated to remain, and the percent of work that is completed.
 - Enable tracking of resources assigned to requests of this type.

- If resource tracking is enabled, configure resource security on the request type. (Determine who can access and modify the resources and actuals displayed in the **Resource** section of the request details page in the PPM Dashboard.)

For details on how to use the **Resources** tab, see ["Configuring Resource Tracking" on page 182](#), ["Tracking Resources Assigned to Requests" on page 182](#), and ["Configuring Request Types for Use with Time Management" on page 185](#).

- **Help Content.** Use this tab to add help content to fields, sections and request types. For details, see ["Configuring Help Contents for Request Types " on page 195](#).

Controlling Request Field Behavior

You can control the behavior of request fields in PPM Center by setting up *request type rules* and *status dependencies*. Because these two methods of determining field behavior have functional overlap, they can produce unexpected results when used together. Therefore, it is important to understand what each does and how they can interact to propagate changes through the system.

Status Dependencies

Status dependencies can be used for status-based business logic. You can configure status dependencies to change request type field attributes (visible, editable, or required, cleared, and so on) whenever a request moves to a new status.

When a user acts on a request, the PPM Center system does the following:

- The system looks at the workflow and computes the next status that the request would have as a result of the action.
- Based on status dependencies configured for the next status, the system "looks ahead" to determine whether any fields will be required or need to be reconfirmed for that status.
- If status dependencies dictate that a field must be reconfirmed, or that it will be required for the next status, and currently contains no value, then the user is directed to an intermediate page, which is referred to as the *look-ahead page*. The look-ahead page displays all required fields that are empty, and all fields that contain values that must be reconfirmed. The user must either provide the required values so that the request can transition to the next status, or cancel the action.

With status dependencies, a request can transition one way in a workflow to make some fields required, but transition a different way and make no fields required. In some organizations, users must

complete specific required fields to cancel a request. The "look-ahead" functionality of status dependencies enables you to avoid this kind of complication.

For detailed information on how to configure status dependencies, see ["Configuring Request Field Status Dependencies" on page 136](#).

Request Type Rules

Request type rules are used to drive dynamic behavior of request fields directly from the request detail page, independent of status changes. This is often critical for ensuring the usability of complex request forms, and enables you to add advanced logic into a request to help guide the user, simplify data entry, and minimize misunderstandings.

You can use request type rules to define dependencies between fields, and use these dependencies to set default values in any field, show or hide specific fields, make other fields required or optional, change the styling of a field, and other dynamic behavior. Each request type can contain as many rules as necessary.

As an example, consider a request type designed to handle a project proposal process. Among the fields it contains are a **Projected Cost** field and a **# Resources** field. The request type also contains a **Project Size** field, which is to be used to qualitatively categorize a proposed project as "small" or "large," which the workflow depends on later in the process. Rather than forcing users to make a subjective judgement on what constitutes a "small" or "large" project, the **Project Size** field can be hidden and automatically defaulted with request type rules. A rule can be defined to set the **Project Size** to "small" if the **Projected Cost** and **# Resources** fall below specified values, and to "large" otherwise.

For detailed information on how to configure request type rules, see ["Request Type Rules" on page 139](#).

Opening the Request Type Workbench

To open the Request Type Workbench:

1. Log on to PPM Center.
2. On the **Open** menu, click **Administration > Open Workbench**.

The PPM Workbench opens.

3. On the shortcut bar, click **Demand Mgmt > Request Types**.

The Request Type Workbench opens.

Setting Request Type Defaults

You can select a default request header type and a default workflow for a request type and the default value for the maximum number of fields in a request type.

To set the default request header type and workflow:

1. On the PPM Workbench shortcut bar, click **Demand Mgmt > Request Types**.

The Request Type Workbench opens.

2. Click the **Results** tab.

3. Click **Setup Request Header**.

The Request Header Setup Dialog window opens.

4. Provide the information specified in the following table.

Field Name	Description
Default Workflow	Use this auto-complete to select a default workflow. This default workflow is used for all new request types, unless the associated request type has a defaulting rule for the workflow.
Default Request Header Type	Use this auto-complete to select a default request header type. This request header type is used for all new request types, unless a different request header type is specified in the individual request type.

5. Click **OK**.

The selected workflow and request header type are now defaults.

To change the default number of fields for a request type:

1. On the PPM Workbench shortcut bar, click **Configuration > Validation**.

The Validation Workbench opens to the **Query** tab.

2. Click **List**.

The **Results** tab lists all validations.

- Find, and then open the **CRT- Max Custom Fields** validation.
- Click **New**.

The Add Validation Value window opens.

- Provide the information described in the following table.

Field Name	Description
*Required	
*Code	Type the validation value in this field. Validation values are expressed in increments of 50. The Code and Meaning fields must display the same value.
*Meaning	Type the validation value in this field. (The Code and Meaning fields must contain the same value.)
Desc	You can type a short description of the validation in this field.
Enable?	To keep the validation value available to the system, leave this checkbox selected (the default).
Default	To specify this value as the default validation value, select this checkbox.

- Click **OK**.

The Validation window lists the new validation.

- Click **OK**.

Configuring General Information for Request Types

To configure the general information for a new request type:

- On the PPM Workbench shortcut bar, click **Demand Mgmt > Request Types**.

The Request Type Workbench opens.

- Click **New Request Type**.

The Request Type window opens.

3. Complete the information described in the following table, as necessary.

Field Name	Description
Request Type Name	Type the name of the request type.
Request Header Type	Use the auto-complete to select a request header type to use with this request type, or to create a new request header type, click New .
Creation Action Name	Optional. You may just leave it empty. You may also provide a value for it, which is displayed under the Create Based On Desired Action section as Most Recently Created on the Create New Request page instead of the request type name itself.
Category	From this list, select the category that includes the request type. You can use the Validation Workbench to create categories (for example, Sales and Support or General Administration) based on your business needs. The categories you create are displayed in the Create New <Request_Type> window in the standard interface. [Validation = CRT - Request Type Category]
Extension	For request types created for an Deployment Management extension, select the extension from the list.
Description	Type a clear description of how the request type is to be used.
Meta Layer View	The reporting meta layer (RML) contains a database view for each request type, which displays data columns for each field defined for the respective request type. This eliminates the need for a report writer to navigate the generic transactional data model when creating an Demand Management report. A request type must have a corresponding meta layer view name that is unique to the system, and that has the format MREQ_<Request_Type_Name>. (Use uppercase text strings and underscore characters only.) For detailed information about (reporting meta layer) views for Demand Management, see the <i>Reporting Meta Layer Guide and Reference</i> .
Max Fields	From this list, select the maximum number of fields the request type can have. See " Setting Request Type Defaults " on page 115.
Enabled	To keep this request type available to PPM Center, leave Yes selected (the default).

4. Do one of the following:

- To save the changes and close the Request Type window, click **OK**.
- To save the changes and leave the window open, click **Save**.

Creating and Configuring Request Type Fields

This section provides an overview of request type fields, and information about how to create and configure fields for request types.

Overview of Request Type Fields

Each request type field has the following three associated attribute criteria:

- Criteria for visible fields
- Criteria for editable fields
- Criteria for default fields

The following sections provide information about these attributes and instructions on how to set them for your request type fields.

Criteria for Visible Fields

You can specify that a request type field be visible to or hidden from users. "[Table 5-1. Criteria for visible fields](#)" below lists the methods you can use to do this.

Table 5-1. Criteria for visible fields

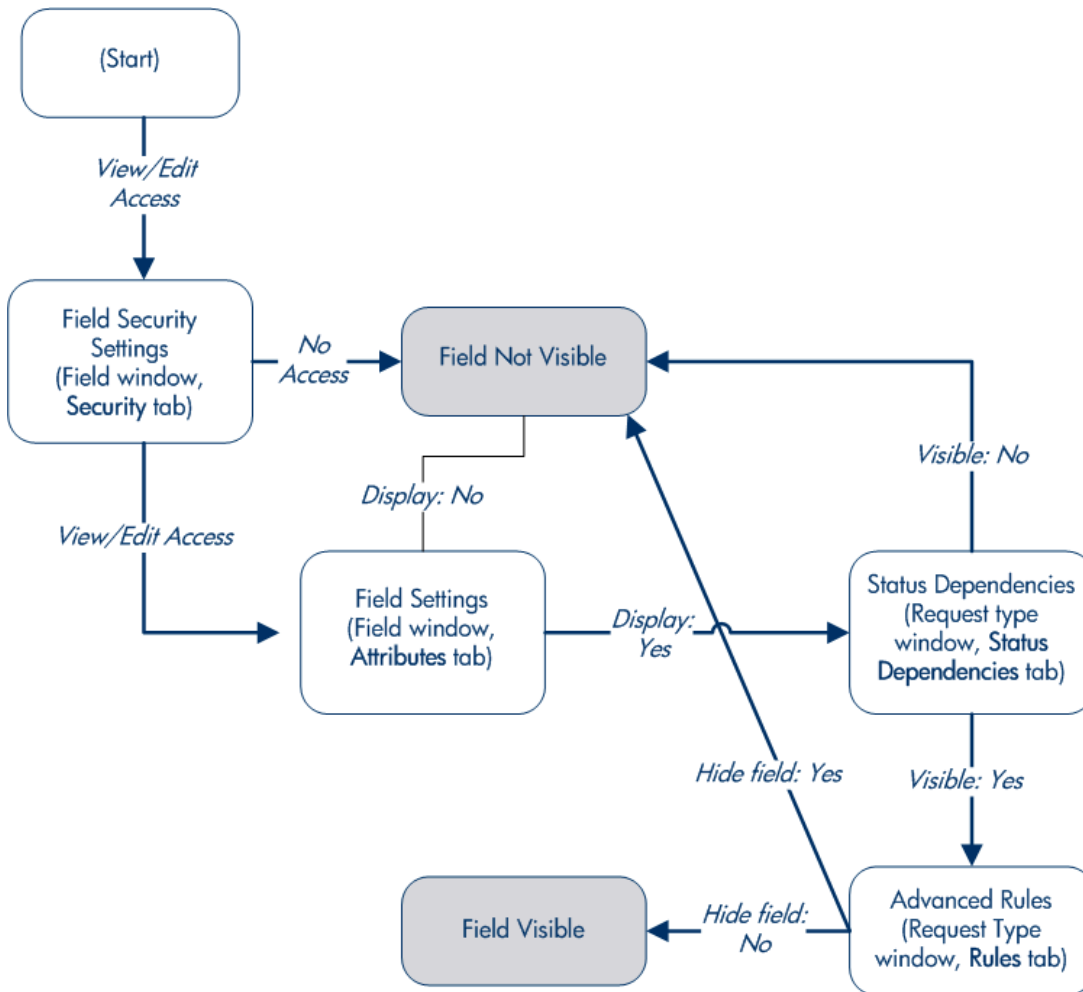
Criteria	Description
Field attributes	Use the Attributes tab in the Field window to designate a field as always visible or always hidden. For details, see " Creating Fields for Request Types " on page 121 .
Request status	You can specify field visibility based on request status (linked to the workflow step). For details, see " Configuring Request Statuses for Request Types " on page 133 .
Field security	You can use the Security tab in the Field window to specify field visibility for particular users or security groups. For details, see " Creating Fields for Request "

Table 5-1. Criteria for visible fields, continued

Criteria	Description
	Types" on page 121.

"Figure 5-3. Field visibility interactions" below shows how Demand Management determines field visibility for a particular user. In this diagram, the user is assumed to have permission to view the requests, which requires the correct license, access grants, and settings on the **User Access** tab in the Request Type window.

Figure 5-3. Field visibility interactions



Criteria for Editable Fields

You can configure request type fields to become read-only or editable based on request status or users and user groups. "Table 5-2. Criteria for editable fields" on the next page lists the methods you can use

to determine field editability.

Table 5-2. Criteria for editable fields

Criteria	Description
Request status	You can specify that a field is read-only based on request status. For details, see "Configuring Request Statuses for Request Types " on page 133.
Field security	Use the Security tab in the Field window to designate fields as read-only for specific users or security groups. For details, see "Creating Fields for Request Types" on the next page.
Advanced UI rules	Advanced UI rules can be used to make a field editable or read-only, based on dependencies that have been configured. Even if request status and field-level security dictate that a user can edit a field, that user will not be able to edit the field if an advanced rule is triggered based on some other dependency that makes it view-only. For details, see "Advanced Rules for Request Types" on page 146. Note: You cannot use special commands to trigger rules.

Criteria for Default Fields

You can configure a field to automatically update the values in other fields. ["Table 5-3. Criteria for default fields"](#) below lists the configuration methods you can use.

Table 5-3. Criteria for default fields

Criteria	Description
Field defaulting	From the Default tab in the Field window, you can link the value in a field to the value in other fields defined for the same entity. For example, a request type field can default to the username of a specific manager when the value in another field in that request type equals "Critical." For details, see "Creating Fields for Request Types" on the next page.
Request type rules	From the Rules tab in the Request Type window, you can configure a request type to automatically populate one or more fields based on the values in the dependent fields. For example, if a field has the value "Bug Report," then the workflow, contact name, contact phone, and department can be automatically set by corresponding request type rules. For details, see "Request Type Rules" on page 139.
Request type commands	You can use commands to control certain aspects of request type field behavior. You can specify that the commands stored in the request type be run at specific workflow execution steps in the resolution process.

Table 5-3. Criteria for default fields, continued

Criteria	Description
	<p>These commands can then manipulate the data inside a request type field.</p> <p>For example, you can construct a command to consider a number of parameters, and then default a field based on those parameters. This provides an advantage over the defaulting features in the Field window, which can only default based on a single parameter stored in the same request type.</p> <p>Using commands to control field values using commands can be useful for:</p> <ul style="list-style-type: none"> • Storing a value from an execution (You can also use workflow parameters to do this.) • Clearing a field after evaluating multiple parameters. <p>For more information on how to set up commands to control field defaulting, see the <i>Commands, Tokens, and Validations Guide and Reference</i>.</p>

Creating Fields for Request Types

You use the Field window to create and configure request type fields. From the Field window, you can specify:

- Whether the field is hidden displayed
- Whether the field is editable under specific circumstances
- Default field values
- Dependencies tied to values of other fields in the request type

To create a request type field:

1. Log on to PPM Center.
2. On the **Open** menu, click **Administration > Open Workbench**.
The PPM Workbench opens.
3. On the shortcut bar, click **Demand Mgmt > Request Types**.
The Request Type Workbench opens.

- Open a request type.

The Request Type window opens to the **Fields** tab.

- Click **New**.

The Field: New window opens to the **Attributes** tab.

- In the top section of the Field window, provide the information described in the following table.

Field or Option	Description
*Required	Description
*Field Prompt	Type the label for the request type field for display on the request details page.
*Token	Type an uppercase text string to use to identify this field. The token name must be unique for the specific request type (for example, ASSIGNED_TO_USER_ID).
Description	(Optional) Type a short description of the field.
Enabled	If you do not want the field enabled for this request type, click No . Otherwise, leave Yes selected.
*Validation	Use the auto-complete to specify the validation for this field. The validation determines what type of component this field is to be (plain text field, date field, drop-down list, and so on) and what values are valid for the field. For more information on validations, see the <i>Commands, Tokens, and Validations Guide and Reference</i> . Note: When you specify the validation, make sure that the token(s) referenced by the validation already exist.
Multi-Select Enabled	If you selected an auto-complete component for the validation, and you want to allow users to select more than one value, then click Yes . If you selected a non-ACL component for the validation, users cannot select more than one value, regardless of whether or not the Multi-Select Enabled option is enabled and whether or not its value is Yes. This also applies to Web services. Note: Some header fields do not allow the multi-select option.

- On the **Attributes** tab, provide the information described in the following table.

Attribute Name	Description
Section Name	From this list, select the request detail page section that is to display the

Attribute Name	Description
	field. This field is only available if you have created two or more sections to display for requests of this type. For instructions on how to create sections, see "Adding Sections to Request Types" on page 129
Display Only	To make the field read-only, and uneditable even at initial request creation, select Yes .
Transaction History	To turn on transaction auditing for this field, select Yes . Whenever the value in this field changes in a request, that change is logged to a transaction history table.
Notes History	To turn on notes auditing for this field, select Yes . Whenever the value in this field changes in a request, the change is logged in notes for the request.
Display on Search and Filter	To prevent the field from being displayed on Search and Filter pages in the standard interface, select No .
Display	To prevent the Request Type field from being displayed for requests of this type, select No . Note: If you select No , no matter what you select for Display Only, the field is hidden.
Search Validation	Use the auto-complete to specify the logic to use to determine the valid search values in this field.

Note: The total number of fields in a request type cannot exceed 359 if you enabled any one of the following attributes in any field:

- **Notes History**
- **Transaction History**

8. Click the **Default** tab.
9. Provide the following information:
 - a. To specify that the field is to have no default value, in the **Default Type** field, leave **None** selected. To specify that the field is to have a constant as the default value, select **Constant**.

You can also configure a default value that is based on the value in another field or derived from a parameter. To configure these default types, you configure a rule or a command to automatically populate the request type field. For instructions on how to do this, see ["Request Type Rules" on page 139](#) and ["Configuring Commands for Request Types" on page 165](#).

- b. If you selected **Constant** as the default type, then in the **Visible Value** field, type the constant value.

10. Click the **Storage** tab.

On the **Storage** tab, the field is automatically placed into the next available position within the database based on the current field attributes. To locate a field in the database, an administrator can open the Field window for a specific field in a request, and use the **Storage** tab. This is useful for reporting purposes.

If necessary, you can use the **Storage** tab to specify a field location within the database when creating a new field. However, the standard practice is to allow the interface to automatically position the field for the administrator.

The **Storage** tab automatically stores the value for a text field that has a maximum length of 4000 characters in column 41 or higher.

11. Provide the following information:

- a. From the **Max Length** list, select a value to set the maximum number of characters for the field value (either 200 or 4000).
- b. From the **Batch Number** list, select the batch number for the field.

Note: Batch number is based on the number of maximum fields. For every 50 fields, one batch is created. 10 Ten of these the 50 fields in a batch can be contain more than 200 characters in length characters. Enabled This list is enabled only when if there are more than 50 fields (creating more than one batch).

- c. From the **Parameter Col** list, select the internal database column that in which the field value is to be stored.

Note: These values are stored in the corresponding column in the request details table for each batch of the given request type. Information can be stored in up to 50 columns using request type, allowing up to 50 fields per batch. No two fields in a request type can use the same column number within the same batch.

12. Click the **Security** tab.

The Security tab displays the information described in the following table.

Field Name	Description
Visible To	Lists all users, security groups, and linked tokens to which the field is visible. By default, the field is visible to all users.

Field Name	Description
Editable By	Lists all users, security groups, and linked tokens for which this field is editable. By default, the field is editable for all users.

13. To change the default field security settings for the field:

a. Click **Edit**.

The Edit Field Security window opens.

b. To control who can view the field:

i. Clear the **Visible to all users** checkbox.

ii. In the **Select Users/Security Groups that can view this field** list, select **Security Group, User, Standard Token, or User Defined Token**.

iii. Use the auto-complete to select the security group, user(s), or token.

c. To assign your selection editing rights in addition to viewing rights to the field, leave the **Provide Editing Rights** checkbox selected.

d. To add the selection to the list of users and security groups who can view the field, click the **Add** arrow.

e. To hide the field from a selected security group, user, or token now listed on the right, clear the **Visible** checkbox in the corresponding row.

f. To make the field read-only for a selected security group, user, or token, clear the **Editable** checkbox in the corresponding row.

g. To remove field access rights entirely, select the user, security group, or token, and then click **Remove**.

h. After you finish configuring field security, click **OK**.

The **Security** tab is updated with the list of users, security groups, and tokens with viewing or editing rights to the field.

If you add field-level security to fields on a request type that has been used to create requests, the PPM Center database tables are updated with this new configuration. Because of the scope of database changes, you must collect database schema statistics. For information about how to collect database schema statistics, see the *Installation and Administration Guide*. For help with this task, contact your system administrator.

Note: There can only be 500 rows per column, three columns per tab, and a maximum of 20 tabs for each request type.

When taking advantage of the reporting meta layer functionality, those fields contained within the first four batches (200 fields) are available for reporting.

Copying Fields for Request Types

To simplify the process of adding fields to a request type, you can copy the definition of existing fields from other request types.

To copy a request type field:

1. Log on to PPM Center.
2. On the **Open** menu, click **Administration > Open Workbench**.

The PPM Workbench opens.

3. On the shortcut bar, click **Demand Mgmt > Request Types**.

The Request Type Workbench opens.

4. Open a request type.
5. In the Request Type window, click the **Fields** tab.
6. Click **New**.

The Field window opens.

7. Click **Copy From**.

The Field Selection window opens.

8. Specify the search criteria (such as the token name or field prompt) in the header fields in the top section of the window.

Note: You can perform more complex queries. For example, you can list all fields that reference a certain validation or that are used by a certain entity. Because of the large number of fields in the system, use one or more query criteria to limit the number of fields returned.

9. Click **List**.

The Query Results table lists the search results.

10. Select the field to copy, and then click **Copy**.

Note: For security purpose, the security settings will not be copied. If necessary, you can

manually set up the security settings for the newly copied field.

11. In the Field window, make any necessary modifications, and then click **OK**.

The **Fields** tab lists the new field.

Removing Fields from Request Types

To remove a field from a request type:

1. On the PPM Workbench shortcut bar, click **Demand Mgmt > Request Types**.

The Request Type Workbench opens.

2. Open a request type.
3. In the Request Type window, click the **Fields** tab.
4. Select a field, and then click **Remove**.

Note: You cannot remove a request header type field.

5. Click **OK**.

Configuring Layouts for Request Types

The request type layout determines the look and placement of fields on a request details page. The following sections provide instruction on several ways to modify request type layout.

Modifying Field Widths on Request Types

To change the width of a field:

1. On the PPM Workbench shortcut bar, click **Demand Mgmt > Request Types**.

The Request Type Workbench opens.

2. Open the request type that includes the field you want to modify.
3. In the Request Type window, click the **Layout** tab.

4. In the **Sections** section, select a section that contains the field to modify.
5. In the **Selection Section Layout** section, select the field.
6. At the bottom of the tab, from the **Width** list, select a field width.

Fields can have a width of 1, 2, or 3. The field width must correspond to the column location. For example, a field located in Column 2 cannot have a width set to 3. For fields of the Text Area component type, you can determine the number of lines the Text Area will display. Select the field and change the value in the **Height** field.

If the field you select is not of type Text Area, this attribute is blank and you cannot modify it.

7. Click **OK**.

Moving Fields On Request Types

To move a field or a set of fields:

1. On the PPM Workbench shortcut bar, click **Demand Mgmt > Request Types**.
The Request Type Workbench opens.
2. Open a request type.
3. In the Request Type window, click the **Layout** tab.
4. In the **Sections** section, select a section that contains the fields to rearrange.
5. In the **Selection Section Layout** section, select a field or fields to move.
6. Use the arrow buttons at the bottom of the tab (or the corresponding keyboard arrows) to change the position of the fields.

Note: You cannot move a request header type field.

If the field layout for a request type contains multiple sections, you can move fields from one section to another. To move a field to a different section:

- a. In the **Sections** section, select a section that contains a field that you want to move to a different section.
- b. In the **Selection Section Layout** section, select the field to move.
- c. From the list to the right of **Move To**, select a section name.

- d. Click **Move To**.

The field is moved to the section you selected in [step c](#).

7. Click **OK**.

Adding Sections to Request Types

To add a new section to a request type:

1. On the PPM Workbench shortcut bar, click **Demand Mgmt > Request Types**.

The Request Type Workbench opens.

2. Open a request type.
3. In the Request Type window, click the **Layout** tab.
4. Under the **Sections** list, click **New**.

The Input window opens.

5. Type a name (up to 30 characters) for the new section, and then click **OK**.

Before you can save a new section, you must first add fields to it.

Note: If all the fields you add to the section have a width of one column and are all in the same column, all displayed columns automatically span the entire section when a request of the given request type is viewed or edited.

6. Add one or more fields to the new section. You can either move existing fields from a different section to the new section (see ["Moving Fields On Request Types" on the previous page](#)), or create fields for the section from the Fields tab (see ["Creating Fields for Request Types" on page 121](#)).
7. To view how the new section will look to users who process the request, click **Preview**.

The Request Preview window opens and shows how the sections and fields are to be displayed on the request detail page.

Note: Hidden fields are displayed as blanks in the preview window, however, they are removed on the request detail page, and will be replaced by fields located to the right side of the hidden fields on the same row.

Cells with no fields are treated the same as hidden fields.

8. Click **Save**

When a user creates requests of this type, the new section with the defined custom fields is visible.

Changing Section Names on Request Types

You can rename sections you added to a request type. You cannot change the name of sections added to a request type by the request header type.

To change the name of a section:

1. On the PPM Workbench shortcut bar, click **Demand Mgmt > Request Types**.

The Request Type Workbench opens.

2. Open a request type.
3. In the Request Type window, click the **Layout** tab.
4. In the **Sections** section, select a section.
5. Click **Rename**.

The Input window opens.

6. Type a new section name. (Custom section names can contain up to 30 characters.)

After requests are generated for the given request type, the new section with the defined custom fields is visible.

7. To view what the layout will look like to the user processing the request, click **Preview**.

An HTML window opens to shows the fields as they are to be displayed.

If all the fields have a width of one column and are all in the same column, all displayed columns automatically span the entire available section when a request of the given request type is viewed or edited.

Note: Hidden fields are displayed as blanks in the preview window, however, they are removed on the request detail page, and will be replaced by fields located to the right side of the hidden fields on the same row.

Cells with no fields are treated the same as hidden fields.

8. Click **OK**.

Deleting Sections on Request Types

You can delete sections you added to a request type. You cannot delete sections added to a request type by the request header type.

To delete a section:

1. On the PPM Workbench shortcut bar, click **Demand Mgmt > Request Types**.
The Request Type Workbench opens.
2. Open a request type.
3. In the Request Type window, click the **Layout** tab.
4. In the **Sections** section, select a section.
5. Click **Remove**.
6. Click **OK**.

Configuring Display Columns for Request Types

Certain information in a request can provide a useful summary-level description of the request. This can include information such as the request type, a description of the request, and a priority. For each request type, you can control which request columns can be displayed in the following pages:

- Request list portlets
- Request search results page
- Request drill-down pages accessed by clicking on request chart portlets

"[Figure 5-4. Display columns set in the Request Type window](#)" below shows how the settings in the Request Type window control the columns that can be displayed on a request list portlet page.

Figure 5-4. Display columns set in the Request Type window

The top screenshot shows the 'Request Type : Bug' configuration window. The 'Request Type Name' is 'Bug' and the 'Reference Code' is '_BUG'. The 'Request Header Type' is 'Default'. The 'Description' is 'Bug Request type'. The 'Meta Layer View' is 'MREQ_' and 'BUG'. The 'Max Fields' is '50' and 'Enabled' is 'Yes'. The 'Display Columns' tab is selected, showing a list of available columns and a list of columns to be displayed. The 'Display Columns' list includes: Request No *, Request Type, Description, Request Status, Assigned To, Priority, and Created By. A note states: 'Note: The first 5 columns will appear in a narrow portlet. The first 7 display columns will appear in a wide portlet. All display columns will appear in the search results, maximized view and drill down pages.'

The bottom screenshot shows the 'Request List' table with the following data:

Req #	Request Type	Description	Status	Assigned To	Priority	Created By
31323	Bug	Currency setting on time sheets do not reflect locale	New	Leslie Franklin	Critical	John Groom
31311	DEM - Application Bug	Application fails when saving after using new Ops modules	In Further Review	Bridget Holbrook	High	Joseph Banks
31306	DEM - Application Enhancement	Add new purchasing codes for KMM subsidiary	New	Bridget Holbrook	Normal	John Groom
31305	DEM - Application Enhancement	Add new purchasing codes	Pending Functional Spec	Bridget Holbrook	Normal	John Groom
31304	Project Details		In Planning			Joseph Banks

To configure the columns for display in list portlets:

1. On the PPM Workbench shortcut bar, click **Demand Mgmt > Request Types**.

The Request Type Workbench opens.

2. Search for, and then open a request type.
3. In the Request Type window, click the **Display Columns** tab.
4. In the **Available Columns** field, select the columns to display.

Note: You can use the Shift and Ctrl keys to select multiple column headings.

5. Click the right-pointing arrow.

The **Display Columns** field lists the selected items.

6. In the **Display Columns** box, select any columns that you do not want to be available for display.

7. Click the left-pointing arrow.
8. Click **OK**.

These settings determine the default columns displayed in the request portlets. Users can edit portlet preferences to modify column display in the portlets on their dashboard pages. These settings also determine the columns displayed for results returned for advanced searches in the Request List portlet or Request Search Results page.

Configuring Request Statuses for Request Types

A request can acquire different statuses as it progresses along its workflow. These statuses can be used to drive field behavior by linking the workflow processes to specific information in the request.

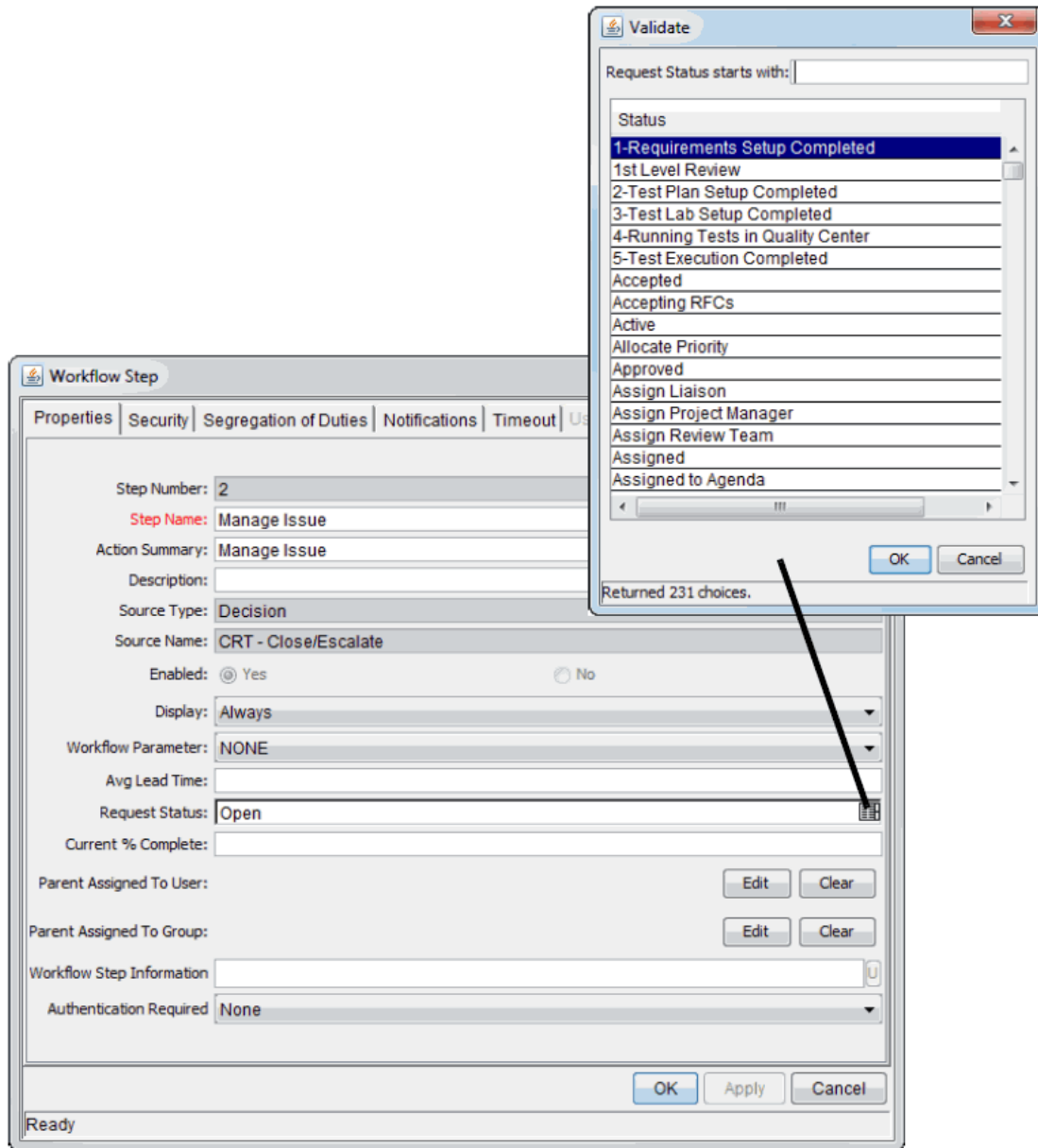
Overview of Request Statuses

A requests can take on different statuses as it progresses through its workflow steps. Demand Management provides over 200 request status values. A few of these are:

- Submitted
- Accepted
- In Design
- Assigned
- In Progress
- On Hold
- Pending Confirmation
- Check Test Completion Status
- Complete

These status values are linked to the workflow steps to drive the request logic. ["Figure 5-5. Request status specified in the Workflow Step window"](#) below shows how status values are linked to workflow steps.

Figure 5-5. Request status specified in the Workflow Step window



As a request moves along its workflow, its status changes at particular steps. Each status can be linked to request field behavior through the **Status Dependencies** tab (from the Request Type Workbench). For more information on linking request statuses to field behavior, see ["Configuring Request Field Status Dependencies"](#) on page 136.

Before you can link request status values to workflow steps, the request type must first have all required status values. You use the **Request Status** tab in the Request Type window (["Figure 5-6. Request Status tab and Request Status List window"](#) below) to configure the list of available status values.

Figure 5-6. Request Status tab and Request Status List window

Status Name	Enabled	Auto Link
1-Requirements Setup Completed	Y	Y
1st Level Review	Y	N
2-Test Plan Setup Completed	Y	Y
3-Test Lab Setup Completed	Y	N
4-Running Tests in Quality Center	Y	Y
5-Test Execution Completed	Y	N
Accepted	Y	N
Accepting RFCs	Y	N
Active	Y	N
Allocate Priority	Y	N
Approved	Y	N

Buttons: New, Edit, Delete, Refresh, Close

231 Request Status Records Loaded

If the **Available Request Statuses** list does not display the value you want, you can create the status value you need. To set the initial status for a request, select a value from the **Initial Request Status** list.

Creating Request Statuses for Request Types

To create a new request status:

1. On the PPM Workbench shortcut bar, click **Demand Mgmt > Request Types**.

The Request Type Workbench opens.

2. Open a request type.

3. In the Request Type window, click the **Request Status** tab.

4. Click **Request Status**.

The Request Status List window opens.

5. Click **New**.

The Request Status: New window opens.

6. Complete the following fields:

- In the **Status Name** field, type a name for the new status.
- The **Reference Code** box is automatically populated based on the status name you typed. You can either accept this default value, or replace it with a different status name.
- To make the status available to the system (and display it in the **Available Request Status** column for all new request types), leave the **Enabled** option set to **Yes**. To make the status unavailable to the system, select **No**.
- To allow the new status to automatically link to all new request types, for **Auto Link**, click **Yes**. Otherwise, leave **No** selected.

7. Click **OK**.

8. In the Request Status List window, click **Close**.

Configuring Request Field Status Dependencies

On a request, field behavior can be linked to the status of the request. For example, a request cannot move to the Assigned status unless the **Assigned To User** field contains a value. In addition, as long as the request has a status of Assigned, a user cannot change the **Assigned To User** field.

To make this work, the **Assigned to** field is configured with the following settings for the **Assigned** status:

- **Visible = Yes**
- **Editable = No**
- **Required = Yes**
- **Reconfirm = No**
- **Clear = No**

You configure field dependencies from the **Status Dependencies** tab in the Request Type window.

To assign field properties based on request status:

1. On the PPM Workbench shortcut bar, click **Demand Mgmt > Request Types**.

The Request Type Workbench opens.

2. Open a request type.
3. In the Request Type window, click the **Status Dependencies** tab.
4. From the **Request Status** list, select one or more request status values.

Note: You can use the Shift and Ctrl keys to select multiple values.

5. In the **Field** table, select the request field for which you want to configure properties based on the selected request status.
6. Under the **Field** section, set the options described in the following table.

Field Name	Description
Visible	Determines whether the field is visible to users while a request is in the selected request statuses. If this option is set to No , the field is hidden while the request is in the selected statuses.
Editable	<p>If the Editable option for a request field is set to No for a specific status, then users cannot edit the field while the request is in that status. If the Required or Reconfirm option for a request field must be set to Yes, then the Editable option must also be set to Yes.</p> <p>At certain stages in a request resolution process, you may want to ensure that specific fields are not changed. For example, when a request of type Vendor Bug has the status Patch Applied, you want to ensure that the Patch Number field does not change. To accomplish this, you set the Patch Number field to be read-only for all request statuses after a certain point in the workflow. (Of course, you would probably make the Patch Number field required in an appropriate previous status, to ensure that it has a valid value before it becomes a read-only field.)</p>
Required	Specifies whether the field is required or not while a request is in the selected request status(es). If a field is required when a request is in the selected status, a user must provide a value for the field before the request can move to that status. When the workflow transitions to the status, the "look-ahead" page is displayed to require the user to fill out the fields to be required for that status (if any of those fields do not already contain values).
Reconfirm	If the Reconfirm option for a field in the request type is set to Yes , the field is presented to the user on the look-ahead page before the request is allowed to transition into this status. The user can review the field value and, if

Field Name	Description
	necessary, change it.
Clear	<p>The Clear field is used in conjunction with other dependencies to remove the content of a field. The clear flag is used as follows:</p> <ul style="list-style-type: none"> ○ If set to Yes, and either or both the Required and Reconfirm options are set to No, the field is not presented to the user on the look-ahead page, but the field is automatically cleared when the request moves to status. ○ If set to Yes, and either or both of the Required and Reconfirm options are also set to Yes, then the field is cleared and displayed on the look-ahead page as the request is moving to this status. If required, the user must provide a valid value in the field before the request can complete the transition to the new status. If only reconfirming, then the user can decide whether or not to provide a value before continuing. <p>Note: To present the Reconfirm field to the user for mass update of records, set the Clear field to Yes.</p>

Note: You can also control field attribute such as Editable and Visible by configuring an advanced request type rule that includes JavaScript-based logic. For details, see ["Advanced Rules for Request Types" on page 146](#).

You can configure multiple fields simultaneously by using the **Ctrl** or **Shift** keys to select the fields and then change the attribute values. You can also select multiple status values and change the same fields if those states require the same attribute values for the same fields.

7. Click **OK**.

Status Dependencies Interactions

["Table 5-4. Status dependencies interactions" on the next page](#) shows the results of different combinations of the **Required**, **Reconfirm**, and **Clear** selections. For each request status within a request type, there can be up to a maximum of 250 fields with a required state and 250 fields with a reconfirm state.

Caution: Please keep in mind that there is some overlap between status dependency functionality and request type rule functionality. It is important that you understand how status dependencies and request type rules can interact and possibly produce unintended results. Plan carefully before configuring either. HPE strongly recommends that you thoroughly review ["Request Type Rules" on the next page](#), and plan your field status dependencies accordingly.

Table 5-4. Status dependencies interactions

Dependencies			Results at Given Status		
Required	Reconfirmed	Clear	Display	Color	Data Shown
No	No	No	No	N/A	N/A
No	No	Yes	No	N/A	N/A
No	Yes	No	Yes	Black	Current Data
No	Yes	Yes	Yes	Black	None
Yes	No	No	Yes, if NULL ^a	Red	None
Yes	No	Yes	Yes	Red	None
Yes	Yes	No	Yes	Red	Current Data
Yes	Yes	Yes	Yes	Red	None

a. If a field configured as required, then it is only displayed if its value is blank (NULL). The user must provide a non-NULL value before he can proceed. If the field contains a value, then it satisfies the “required” dependency, and thus does not need to be displayed on the look-ahead page.

Request Type Rules

Request type rules are a powerful way of configuring complex interactions between fields on a request. You can use them to set up automatic population of request fields, change field attributes, or otherwise affect request field behavior based on various dependencies.

One of the most common rule configurations is to set the default workflow when a user creates a new request. From a practical standpoint, you typically know which workflow is appropriate for a request of a given type, and you do not want to offer the user a choice when he creates a request. So, the **Workflow** field is typically disabled, and a rule is defined to automatically set it to the appropriate value.

Keep in mind that there is some overlap between request type rule functionality and status dependency functionality. It is extremely important that you understand how status dependencies and request type rules can interact (and possibly produce unintended results), and plan carefully before you implement either. HPE strongly recommends that, before you configure request type rules, you thoroughly review ["Configuring Request Field Status Dependencies" on page 136](#), and be familiar with all the way in which rules can affect system behavior. Also see ["Important Considerations for Configuring Advanced UI Rules" on page 146](#).

Types of Request Type Rules

You can configure the following request type rules to control Demand Management system behavior:

- You can use a *simple default* rule to set new values in the **Workflow**, **Assigned To**, and **Assigned Group** fields.
- You can use an *advanced* rule with SQL-default logic to set a new value in any fields in the request, based on a SQL statement. You can specify any SQL statement, as long as it returns a single row with two values (a hidden, internal value and a visible value).
- You can use an advanced rule with UI-rule logic to change the behavior or appearance of any field in a request. These rules run JavaScript functions, which are defined on the server in the `RequestRulesSystemLibrary.js` file, which is in the `<PPM_Home>/server/<PPM_Server_Name>/deploy/itg.war/web/knta/crt/js` directory.

Caution: Because the `RequestRulesSystemLibrary.js` file is redeployed at each service pack installation or product upgrade, it is important that you not modify the `RequestRulesSystemLibrary.js` file. Otherwise, your changes are lost.

You can also define your own custom JavaScript functions to use in UI rules. You must define these custom JavaScript functions in the `RequestRulesUserLibrary.js` file, which is also in the `<PPM_Home>/server/<PPM_Server_Name>/deploy/itg.war/web/knta/crt/js` directory.

Note: The `RequestRulesUserLibrary.js` file is not altered when you apply a service pack or upgrade PPM Center. Your custom functions are preserved after these operations.

Predefined JavaScript Functions (Advanced Rules Only)

Demand Management provides several predefined JavaScript functions, which are listed in "[Table 5-5. JavaScript Functions for use with advanced rules](#)" on the next page. All of these JavaScript functions are dynamically applied in the request detail page when invoked by the respective rule, but most do not have any effect in other request edit interfaces (Quick Edit page, Mass Update page, Web service APIs to create or update requests). However, all of the "setFieldRequired()" rules are evaluated upon save in the other interfaces. Any field that is configured to become required as a result of rules will

ultimately be required when the request is saved, and if such a field does not have a value, then a message is displayed, indicating that the request(s) could not be saved.

Table 5-5. JavaScript Functions for use with advanced rules

JavaScript Function	Description
setFieldRequired(< <i>boolean flag</i> >)	Sets the field(s) as required or not required based on the flag parameter value. The Result Fields section must contain at least one field.
setFieldEditable(< <i>boolean flag</i> >)	Sets the field(s) as editable or read-only based on the flag parameter value. The Result Fields section must contain at least one field.
setFieldVisible(< <i>boolean flag</i> >)	Sets field(s) visibility based on the flag parameter value. The Result Fields section must contain at least one field.
setFieldStyle(< <i>CSS class name</i> >)	<p>Sets the field(s) style to the CSS class specified. The Result Fields section must contain at least one field.</p> <p>Any custom styling you want to apply to request fields must first be defined in the <code>RequestRulesUserCss.css</code> file, which is located in the <code><PPM_Home>/server/<PPM_Server_Name>/deploy/itg.war/web/knta/crt/css</code> directory on the PPM Server.</p> <p>Notes:</p> <ul style="list-style-type: none"> • Not all styles on html elements are supported by Internet Explorer and Mozilla Firefox. Please consult the browser-specific implementation for information on which styles are supported. • <code>border</code> attribute for dropdown list is not supported by Microsoft Internet Explorer. This is a Microsoft Internet Explorer limitation.
showMessage(< <i>string text</i> >, < <i>boolean continueProcessing</i> >)	<p>Show the message specified by the "text" parameter (enclosed in single or double quotes) and continue to process the request rule event based on the <code>continueProcessing</code> parameter value.</p> <p>If <code>continueProcessing = false</code>, then the event that triggered the rule is aborted. You can use this function to display a message that reminds the user that he must perform some action before he saves, and prevent the user from saving until he performs that action.</p>
addResource(< <i>resourceFieldToken</i> >)	Adds the specified resource to the list of resources on the request. This rule only applies to request types that

Table 5-5. JavaScript Functions for use with advanced rules, continued

JavaScript Function	Description
	track resources. For more information about tracking resources, see "Configuring Resource Tracking" on page 182.

Rule Event Behavior on the Request Details Page

Rules can be used to achieve dynamic behavior on the request form itself, independent of workflow actions. ["Table 5-6. Effects of rule events" below](#) lists the rule events you can specify for any request type rule (simple or advanced), along with descriptions of how each is processed and how it affects field behavior on the request detail page.

Caution: All of these rule events are applied dynamically in the request detail page. For other interfaces in which requests are updated (Quick Edit page, Mass Update page, Web service APIs to create or update requests), the rules are not processed dynamically as these events occur. Rather, they are processed when requests are saved. Any field that is configured to be required as a result of any of these rules will ultimately be required when the request is saved, and if such a field does not have a value, then the user sees a message indicating that the request(s) could not be saved.

Table 5-6. Effects of rule events

Rule Event	Rule Event Processing and Field Behavior
Apply on creation	Applied any time a request of this type is created. Note: When users change request type on the request details page, target request type rules defined for this event are triggered.
Apply on copy	Applied any time a request of this type is copied. Notes: <ul style="list-style-type: none"> When selecting the "Apply on copy" rule event, you are able to configure field dependencies. This feature is not supported in the "Apply on creation" rule event. If a request type contains rules of the "Apply on copy" rule event and rules of the "Apply on creation" rule event, only the "Apply on copy" rules are triggered when users copy the requests of that type.
Apply on page load	Rules defined for this event are triggered when the request detail page is loaded in a Web browser; however, these rules are not applied when a request is first created. (Use the "Apply on creation" rule event for that.)

Table 5-6. Effects of rule events, continued

Rule Event	Rule Event Processing and Field Behavior
	<p>Note: Rules defined for this event are not triggered on the look-ahead page.</p>
Apply on field change	<p>Rules defined for this event are triggered any time one of the fields listed in the Dependencies section changes to a new value if all of the dependencies specified for the rule are met.</p> <p>For example, consider a rule that has dependencies on when the Priority field has the value "Critical" and the Assigned To field is null. The rule is evaluated whenever either the Priority field or the Assigned To field is changed, and is triggered only if both of the dependency conditions are true when the rule is evaluated.</p>
Apply before save	<p>When a user clicks Save on the request detail page, or clicks a workflow "action" button (which saves the request before processing the workflow action), then any rules defined for this event are processed before the save request is submitted.</p> <p>The "showMessage()" function can cancel the request save if the rule is not satisfied.</p>
Apply before transition	<p>When the user clicks a workflow "action" button, rules defined for this event are applied and then processed before the transition is taken. For example, if a rule is triggered to show a message and cancel the action, then the user sees the message displayed and the transition is not taken.</p> <p>As another example, if an "Apply before transition" rule is defined to set a timestamp in a particular request field, then the rule is processed and the target field is updated before the transition is taken.</p> <p>The "showMessage()" function can cancel the transition event if the rule is not satisfied.</p>

Note: PPM Center does not support the use of special commands to trigger rules.

When a user clicks either **Save Progress** or **Continue Workflow Action** on the look-ahead page, only "Apply before save" rules are triggered. This is because when the user clicks the workflow "action" button on the Request Details page before going to the look-ahead page, PPM has executed "Apply before transition" rules.

Once a rule is triggered, any dependencies defined for the rule (in the **Dependencies** section of the Rules window) must be met before the rule is executed. A simple default rule has a limited set of dependencies available, but an advanced rule can have dependencies set up to run the rule under very specific circumstances based on the current values of any fields in the request.

Note: Rule dependency is not supported if the dependency field is using the () pattern to present negative numbers. For example, using (1000) to represent -1000.

Considering the fact that Auto Complete List and/or Drop Down List values can be localized, which may then cause consistency issue for the dependencies, HPE recommends you to match Text Field code to the code of Auto Complete List or Drop Down List when you set up request type field dependencies.

Rule Event Precedence

It is possible to configure conflicting rules on different events. In this case, "Apply before save" or "Apply before transition" rules are applied last. These rules take precedence if there is a rules conflict.

For example, consider a request type that has an "Apply on page load" rule that makes the **Priority** field required, and an "Apply before save" rule that makes the **Priority** field optional. When the user saves the request, the **Priority** field is option because the "Apply before save" rule takes precedence.

Typically, rules are processed in the order specified in the request type configuration. For advanced rules, this ordering can include a mix of SQL-based and JavaScript-based rules.

Creating Simple Default Rules for Request Types

Simple default rules are used to automatically populate the **Workflow**, **Assigned To**, and **Assigned Group** fields. These fields can be populated based on the **Rule Event** and **Dependencies** fields. You can use any valid combination of these fields to specify values for the **Workflow**, **Assigned To**, or **Assigned Group** fields. For example, by setting the workflow and the rule event to **Apply On Creation**, you can set the default workflow to be used each time a request of that type is used.

Note: **Workflow** is the only required field for simple default rules.

To add a simple default rule to a request type:

1. Log on to PPM Center.
2. From the **Open** menu, click **Administration > Open Workbench**.
The PPM Workbench opens.
3. On the shortcut bar, click **Demand Mgmt > Request Types**.
4. The Request Type Workbench opens.
5. Open a request type for which you want to create a simple default rule.

6. In the Request Type window, click the **Rules** tab.

7. Click **New**.

The Rules window opens in simple defaults mode.

8. In the **Rule Name** box, type a name for the new rule. (Required)

9. In the **Description** box, you can type a short description of the rule.

10. To enable this rule, leave **Yes** selected for the **Enabled?** option.

11. From the **Rule Event** list, select the event to trigger the rule. For a description of selectable events, see ["Rule Event Behavior on the Request Details Page" on page 142](#).

12. If the results of the new rule might trigger rules defined for the same event (same dependency) that occur later in the rules sequence, and you want to prevent that from occurring, for **Stop processing other rules?**, click **No**.

Note: For example, if you define ten rules for the same field change event, you can specify one of them to stop processing through all subsequent rules.

13. In the **Dependencies** section, you can do the following:

- a. To specify a department to which the rule is to apply, from the **Requestor Department** list, select the department name.
- b. To specify an application to which the rule is to apply, use the **Application** auto-complete to select the application.

14. In the **Results** section, do the following:

- a. Use the **Workflow** auto-complete to select the workflow that the rule assigns to a request of this type. (This step is required.)
- b. You can use the **Assigned To** auto-complete to select a user for this rule to assign to the request.
- c. You can use the **Assigned Group** auto-complete to select a security group for rule to assign to the request.

After you save this rule, any new request of this type that matches the combination of values specified for **Requestor Department**, and **Application** automatically updates the **Workflow**, **Assigned To**, and **Assigned Group** fields to contain the values you specified for those fields.

15. To save the rule and create another rule, click **Add**, and then repeat [step 8](#) through [step 14](#). To save the new rule and close the Rules window, click **OK**.

Note: If more than one rule applies for to a given request, then the system typically processes them in the order in which they are specified in the request type configuration.

Advanced Rules for Request Types

You can set up advanced rules that include logic to automatically populate any request field based on user entries. You can also use advanced rules to automatically change field attributes such as visible or hidden, editable or read-only, required or optional, reconfirm, clear, background color, and so on.

Note: Configuring advanced default rules requires knowledge of SQL or JavaScript.

Caution: HPE recommends that you review ["Important Considerations for Configuring Advanced UI Rules"](#) below and ["UI Rules: Examples"](#) on page 150 before you set advanced UI rules.

Important Considerations for Configuring Advanced UI Rules

Configuring well-designed UI rules requires some planning. Here are some guidelines to help you construct useful UI rules that do not have unwanted results:

- Consider all possible events and outcomes. Make sure you handle what happens when the page is loaded, when field values change, when the page is saved, and so on.
- Consider negative cases. If you set up a rule to be fired whenever a field contains a particular value, make sure you handle what happens when the field is changed to a different value. You may need additional rules for that.
- If you set a field as required based on a specific value in another field, you must ensure that you set the field as not required for other possible field values.
- If you configure SQL rules to set values for auto-complete list, make sure that only valid values are set. Auto-complete component's valid value list is not in client UI, so when using rule to set values for auto-complete component, the values are not validated and invalid values can be set. As a result, when users create a request, a value that is not in auto-complete list can be inserted and saved in an auto-complete field as triggered by the SQL rule.
- PPM Center uses #@# as a separator. Make sure that:

- Values used in SQL rules do not contain & or #@#
- SQL rules do not contain #@#

You can use # and @ separately.

- Rules can be executed on the printable version of Request Details, but with the following limitations:
 - Only SQL rules and UI Rule `setFieldVisible()` will be executed.
 - Only **Apply on page load** rule event is supported.
 - The printable version should be opened from request details page instead of by entering the URL address directly.
- If you set a field's field level security as only visible to and editable by one user, you can not make it visible to or editable by other users by using UI rules.
- If a field's `Display` attribute is set to `No` or the field is set invisible by its status dependency, you can not make it visible on the Web UI by using UI rules.
- **Dependencies** field supports the following validations:
 - Text field
 - Auto-complete list
 - Drop-down list
 - Date field (short, medium, and long)
- **Results** field supports the validations listed in the following table:

Rule Validation	setFieldRequired ¹	setFieldEditable ²	setFieldVisible ³	setFieldStyle
Text field	Yes	Yes	Yes	Yes
Text area	Yes	Yes	Yes	Yes
Auto-complete list	Yes	Yes	Yes	Yes
Drop-down list	Yes	Yes	Yes	Yes
Radio buttons	Yes	Yes	Yes	Not supported
Date field	Yes	Yes	Yes	Yes

Rule Validation	setFieldRequired¹	setFieldEditable²	setFieldVisible³	setFieldStyle
Web address (URL)	Yes	Yes	Yes	Yes
Link	Yes	Not supported	Yes	Not supported
Password	Yes	Yes	Yes	Not supported
Attachment	Yes	Not supported	Yes	Not supported
Table component	Yes	Not supported	Not supported	Not supported
Staffing profile	Yes	Not supported	Yes	Not supported

1. setFieldRequired UI rule does not work for the above validations when the **Display** attribute is set to **No** or the **Visible** attribute in Status Dependencies is set to **No**.

2. setFieldEditable UI rule does not work for the above validations when

- **Display Only** attribute is set to **Yes**, or
- **Display** attribute is set to **No**, or
- **Visible** attribute in Status Dependencies is set to **No**, or
- **Editable** attribute in Status Dependencies is set to **No**.

3. setFieldVisible UI rule does not work for the above validations when the **Display** attribute is set to **No** or the **Visible** attribute in Status Dependencies is set to **No**.

Caution: If a request contains a rule that uses `KNTA_MULTI.Find_User_full_names ([SYS.USER_ID])` from `sys.dual`, saving the request may fail or removing associated entities from the request may fail. Use `KNTA_MULTI.Find_UserNames ([SYS.USER_ID])` from `sys.dual` instead.

Note: Using special commands, for example, `ksc_store`, in a workflow execution step to change field value will not trigger rules.

If a field's `Display` attribute is set to `No` or the field is set invisible by its status dependency, you can not make it visible on the web UI by using request level UI rules.


SQL for Hierarchical Display (tree validation) now supports resolving such static tokens as `[SYS.USERNAME]` and `[SYS.USER_ID]`. Other types of tokens, such as Request Details or Context related tokens, are not supported.

Considerations for Configuring Rules in Table Component

Note the following considerations when you configure SQL rules and/or UI rules for table component:

- Set **Process subsequent rules?** option to Yes or No for different validations.
- **Dependencies** field is not required for **Apply on creation** rule event, but required for **Apply on field change** rule event.
- **Dependencies** field supports the following validations:
 - Text field
 - Auto-complete list
 - Drop-down list
 - Date field (short, medium, and long)
- **Results** field supports the following validations:
 - Text field
 - Text area
 - Auto-complete list
 - Drop-down list
 - Radio buttons (yes/no)
 - Date field (short, medium, and long)
 - Web Address (URL)
 - Link
- UI rules support the following JavaScript functions:
 - `setFieldRequired ()`
 - `setFieldEditable ()`
 - `setFieldVisible ()`
 - `showMessage ()`
 - `setFieldStyle ()`

Note: To modify CSS styles, locate and access the `RequestRulesUserCSS.css` file in the following directory: `/opt/ppm/[instance name]/server/[instance name]/deploy/itg.war/web/knta/crt/css/`.

For more information about these JavaScript functions, see "[Predefined JavaScript Functions \(Advanced Rules Only\)](#)" on page 140 or access the *Request Rules JavaScript Function Help* by clicking the  icon next to the **Logic** field in **Rules** window of the Validations Workbench.

- You can set table component as a dependency to trigger request type rules.
- When using SQL rule(s) on text field with **Apply on creation** rule event, make sure not to use strings "`~^~^~`" and/or "`~&~&~`" in any text field as they are reserved.
- When SQL rule(s) with **Apply on creation** rule event are triggered to override default value of a table component column, the default value may still display for a very short period of time (say 0.5 second) due to server speed.

UI Rules: Examples

Following are some examples of well-designed UI rules.

Example 1

If the user sets the **Priority** field on a request of this type to "Critical," you want the field background to change to red to make it highly visible to users who view the request. To do this, you must configure four rules:

Rule 1:

The positive case, in which the **Priority** field value changes to Critical based on some other value.

Event: Apply on field change

Dependency: Priority = "Critical"

Rule logic: `setFieldStyle("redBackground")` Result Fields: Priority

where "redBackground" is a CSS class defined in the `RequestRulesUserCss.css` file in the `<PPM_Home>/server/<PPM_Server_Name>/deploy/itg.war/web/knta/crt/css` directory on the PPM Server, as follows:

```
.redBackground { background:red; }
```

Rule 2:

The negative case, in which the **Priority** field changes from "Critical" to some other value.

Event: Apply on field change

Dependency: Priority != "Critical"

Rule logic: setFieldStyle("whiteBackground")Result Fields: Priority

where "whiteBackground" is a CSS class defined in the RequestRulesUserCss.css file in the <PPM_Home>/server/<PPM_Server_Name>/deploy/itg.war/web/knta/crt/css directory on the PPM Server, as follows:

```
.whiteBackground { background:white; }
```

Rule 3:

Once the request detail page is reloaded, rules #1 and #2 no longer apply (since the **Priority** field is not changing). So, you must define the same rules for the "Apply on page load" event.

Event: Apply on page load

Dependency: Priority = "Critical"

Rule logic: setFieldStyle("redBackground")Result Fields: Priority

where "redBackground" is a CSS class defined in the RequestRulesUserCss.css file in the <PPM_Home>/server/<PPM_Server_Name>/deploy/itg.war/web/knta/crt/css directory on the PPM Server, as follows:

```
.redBackground { background:red; }
```

Rule 4:

The negative case for rule 3.

Event: Apply on page load

Dependency: Priority != "Critical"

Rule logic: setFieldStyle("whiteBackground")Result Fields: Priority

where "whiteBackground" is a CSS class defined in the RequestRulesUserCss.css file in the <PPM_Home>/server/<PPM_Server_Name>/deploy/itg.war/web/knta/crt/css directory on the PPM Server, as follows:

```
.whiteBackground { background:white; }
```

Note that the UI rules you configure apply only to fields. You cannot, for instance, set the style for an entire page. Any logic you add to the **Logic** text box applies to the fields listed in the **Results Fields** table. (The only exceptions are showing an alert and adding a resource, which do not apply to any field in particular.)

Example 2

You want to prevent users from saving a value in a date field if the value falls after tomorrow's date. Display an alert if the rule is violated.

To accomplish this, do the following:

1. Create a hidden field for "tomorrow's" date on the request type.
2. Set a SQL defaulting "Apply on page load" rule to set the value of the hidden field based on the following SQL statement:

```
select trunc(sysdate+1),  
trunc(sysdate+1)  
from dual
```

3. Create an "Apply before save" rule with a dependency that checks to determine whether the value the user provided in the date field is later than the value in the "tomorrow" field, and then, if rule fires, shows the alert message and stops the event.

Example 3

A request type has **Start date** and **Finish date** fields. You want to prevent users from specifying a **Finish date** value that is earlier than the **Start date** value provided on a request of this type, and highlight these fields whenever the rule is violated.

To accomplish this, do the following:

1. Create an advanced "Apply on field change" rule with two dependencies: start is after finish, and finish is before start.

```
Rule logic: showMessage('Finish date cannot be earlier than the Start date. Please  
fix before saving.', false)
```

A change in either the start or the finish date will trigger this rule.

2. Create an "Apply on field change" rule that sets the border of the date fields to red.

```
Rule logic: setFieldStyle("redBorder")
```


where the RequestRulesUserCSS.css file contains:

```
.redBorder { border:2px solid red }
```

3. Create an advanced "Apply before save" rule with one dependency: start is after finish.

Rule logic: showMessage('Finish date cannot be earlier than the Start date. Please fix before saving.', false)

Note: This is the same as the field-change rule from [step 1](#).

4. To keep the red border on the date fields whenever this request displays information that violates the date rule, create an "Apply on page load" rule to set the border of the date fields to setFieldStyle("redBorder").

The borders will be shown every time you view the request, if the condition is met.

Example 4

You want to highlight critical-priority requests to emphasize their urgency.

To accomplish this, do the following:

1. Create an advanced "Apply on field change" rule with the dependency Priority!= "Critical".

Rule logic: setFieldStyle("redBackground")Result Fields: Priority

where the RequestRulesUserCSS.css file contains:

```
.redBackground { background-color:red; font-weight:bold; color:white }
```

This rule changes the style of the field whenever the value in the **Priority** field changes to "critical." However, because the styling will be lost if the page is reloaded, you would create a similar additional rule to trigger on page load.

2. Create an advanced "Apply on page load" rule with the dependency Priority != "Critical".

Rule logic: setFieldStyle("redBackground")Result Fields: Priority

where the RequestRulesUserCSS.css file contains:

```
.redBackground { background-color:red; font-weight:bold; color:white }
```

Example 5

You have a request form that contains three custom fields:

- **Cost** (a currency field)
- **Approver** (an auto-complete field)
- **Comment** (a text field)

If a user provides a value greater than \$500 in the **Cost** field, then you want the **Approver** and **Comment** fields to be required before the request can be saved.

To accomplish this, do the following:

1. Create an advanced "Apply on field change" rule with a dependency on when the **Cost** field value is greater than "500."

Rule logic: `setFieldRequired(true)`Result Fields: Approver, Comment

2. Create the inverse behavior (if the value in the **Cost** field is not greater than \$500, then keep the **Approver** and **Comment** fields optional), create an advanced "Apply on field change" rule with a dependency on when the **Cost** field value is less than or equal to "500."

Rule logic: `setFieldRequired(false)`Result Fields: Approver, Comment

3. Changes to the **Cost** field value trigger the first two rules. But when the request is reloaded (or viewed by someone else in a different session), the results of the rules do not apply. So, you must also create two additional rules that are applied when the request is loaded, as follows:

- a. Create an advanced "Apply on page load" rule with a dependency on when the **Cost** field value is greater than "500."

Rule logic: `setFieldRequired(true)`Result Fields: Approver, Comment

- b. Create an advanced "Apply on page load" rule with a dependency on when the **Cost** field value is less than or equal to "500."

Rule logic: `setFieldRequired(false)`Result Fields: Approver, Comment

Example 6

You have a changed request type that contains the following custom fields:

- **Resolution** (a drop-down list, that includes the value **Automatic**, to capture the resolution of the request)
- **Team Manager** (an auto-complete)
- **Get feedback at completion** (**Yes** and **No** options)

When **Automatic** is selected in the **Resolution** list, you want to hide the **Team Manager** field and display the **Get feedback at completion** option.

To accomplish this, do the following:

1. Create an advanced "Apply on field change" rule with the dependency `Resolution = "Automatic"`.

Rule logic: `setFieldVisible(true)`Result Fields: Team Manager

2. Create an advanced "Apply on field change" rule with the dependency `Resolution = "Automatic"`.

Rule logic: `setFieldVisible(false)`Result Fields: Get feedback at completion

The first two rules are triggered when **Resolution** is set to **Automatic**.

3. Create two additional rules to determine field behavior when the **Resolution** field is set to any value other than automatic, as follows:

- a. Create an advanced "Apply on field change" rule with the dependency `Resolution != "Automatic"`.

Rule logic: `setFieldVisible(false)`Result Fields: Team Manager

- b. Create an advanced "Apply on field change" rule with the dependency `Resolution != "Automatic"`.

Rule logic: `setFieldVisible(true)`Result Fields: Get feedback at completion

4. For the appropriate fields to be visible or hidden when the request is reloaded, or when someone else views the request in a different session, create four "Apply on page load" rules using the same rule logic as you used in the first four rules.

To accomplish this example use case, you would need a total of eight UI rules.

Example 7

You have a request form that contains three numerical fields that represent percentages. For a user to save the request form, the sum of the values in the three fields must total 100%.

If the user clicks **Save**, and the sum is not 100%, you want to:

- Display a pop-up alert that informs the user that he has specified invalid data
- Set the background color of the three numerical fields to red
- Abort the save operation

To make this work, you need an additional field to hold the sum, an "Apply on field change" rule to calculate the sum, and an "Apply before save" rule to correctly set the style of the three number fields.

To accomplish this, do the following:

1. Add three numerical fields to your request type. For each field, specify Percentage Text Field as the validation to use and V_1, V_2, and V_3 as the tokens.
2. To hold the sum, add a numerical field labeled **Sum**, and specify Percentage Text Field as the validation to use and SUM as the token.

Note: Because this field is used only to keep track of the sum, you can hide the field.

3. For each numerical field you created in [step 1](#), define an advanced "Apply on field change" rule with three dependencies on when the field contains any value.

Specify the **Sum** field in the Results table.

4. Specify SQL-defaulting logic to calculate the sum of the three number fields, as follows:

```
select  nv1('[REQD.P.V_1]',0) + nv1('[REQD.P.V_2]',0) +
        nv1('[REQD.P.V_3]',0),
        nv1('[REQD.P.V_1]',0) + nv1('[REQD.P.V_2]',0) + nv1('[REQD.P.V_3]',0)
from dual
```

5. Create an advanced "Apply before save" rule with a dependency on when the **Sum** field is greater than 100.

```
UI Rule logic: showMessage('The sum of values 1, 2, & 3 cannot be greater than 100.
Currently they sum to [REQD.VP.SUM]. Please fix before taking WF action.',
false); setFieldStyle("redBackground")
```

6. Specify the three numerical fields in the Results table so that the background color style is applied to each of them.

Example 8

If you configure the **Description** field with the setting **Editable = No** from the **Status Dependency** tab in the Request Type window, you want the **Description** field to become editable when the **Priority** field is set to "Critical".

To do this, you must configure the following rule:

Event: Apply on field change

Dependency: Priority = "Critical"

Rule logic: setFieldEditable(true)Result Fields: Description

Creating Advanced Request Type Rules

To create an advanced request type rule:

1. Log on to PPM Center.
2. On the **Open** menu, click **Administration > Open Workbench**.
The PPM Workbench opens.
3. On the shortcut bar, click **Demand Mgmt > Request Types**.
The Request Type Workbench opens.
4. Open a request type for which you want to create an advanced rule.
5. In the Request Type window, click the **Rules** tab.
6. Click **New**.
The Request Type Rules window opens.
7. In the **Rule Name** field, type a name for the rule.
8. In the **Description** field, you can type a short description of the rule.
9. To implement this rule, for **Enabled?**, leave **Yes** selected.
10. To specify the event to trigger the rule, from the **Rule Event** list, select an event. For a description of selectable events, see ["Rule Event Behavior on the Request Details Page"](#) on page 142.
11. From the **Rule Type** list, select **Advanced**.
A warning is displayed.
Click **Yes** to continue. The Request Type Rules window switches to advanced mode.
12. If the results of the new rule might trigger rules defined for the same event (same dependency) that occur later in the rules sequence, and you want to prevent that from occurring, for the **Processing cascading rules?** option, select **No**.
13. To set up a dependency:
 - a. In the **Dependencies** section, click **New**.
The Dependencies window opens.
 - b. Use the **Field** list to select a field to trigger the rule.

Tip: After the Validate window opens, expand it to the right so that you can view all displayed columns.

Note: You cannot configure request default rules to trigger from a multiple select auto-complete. Do not select a multi-select auto-complete field.

After you select a field, the following read-only fields are populated:

- **Field Type.** This field displays the type of field you selected.
- **Validation Name.** This field displays the type of validation (such as CRT - Assigned To - Enabled) for the field you selected.
- **Visible Token.** This field displays the name of the visible token (such as REQ.ASSIGNED_TO_NAME) for the selected field.
- **Token.** This field displays the name of the token, (such as REQ.ASSIGNED_TO_USER_ID) for the selected field.

The field you select determines which items are available in the **Condition** list.

- c. From the **Condition** list, select one of the following:
 - A condition such as **is null**, or **contains any value**.
 - A condition to use to compare the value in the selected field with a constant, which you must then specify.
 - A condition to use to compare the value in the selected field with a value in a different field, which you must then specify.
- d. If you selected a condition other than **is null** or **contains any value**, then from the list displayed to the right, either leave **constant value** selected or select **another field value**.

Note: If you specify a numeric field, make sure that you always use.

- e. Do one of the following:
 - If you selected **constant value**, then in the **Value** field, type the value to compare to the field value. (Depending on the field, the **Value** field might be a list, an auto-complete, or text box.)

Caution: If your PPM Center instance supports multiple languages, and you specify a numeric field as a dependency, make sure that you always use English format to specify field values.

Example: Suppose a request type includes a numeric field, and you want to create a rule that triggers an event if the field is set to the constant value 1234.56. In this case, make sure that you use the English variant without a group separator.

- If you selected **another field value**, then use the **Field** auto-complete to select the field for value comparison.

Caution: If your PPM Center instance supports multiple languages, and the other field value you specify is a numeric field, make sure the values for the other field use English format.

- f. Click **OK**.

The dependency you created is listed in the **Dependencies** table.

14. In the **Results** section of the Rules window, click **New**.

The Results window opens.

15. Use the **Field** auto-complete to select a field that the rule is to either automatically populate or for which it is to change an attribute such as read-only or hidden.
16. To close the window, click **OK**. Alternatively, if you want to specify another field, click **Add**, and then repeat [step 15](#).

The **Result Fields** table lists the fields you selected.

17. To specify a rule that controls the behavior of the field(s) listed in the **Result Fields** table, do one of the following:
 - To specify an SQL-based rule that populates the fields:
 - i. In the **Logic** list, leave **SQL Default** selected.
 - ii. In the **Logic** field, type the SQL statement that is to load values into the field(s) you added to the **Result Fields** table section.

Each SELECT value is loaded into its corresponding column in the **Results** table in order.

The system validates the SQL statement to ensure that it contains the correct tokens:

[SYS] tokens, [AS] tokens, or tokens of fields present in the **Dependencies** section. If the SQL statement is invalid, an error message is displayed.

- To specify a JavaScript-based rule that changes field behavior (for example, showMessage, setFieldRequired, or setFieldVisible) but does not populate fields with values:
 - i. In the **Logic** list, select **UI Rules**.
 - ii. In the **Logic** field, type the JavaScript function.

Caution: For the purposes of validation, you cannot just type any JavaScript in the **Logic** text box (in the Rules window). You must use a function defined in the RequestRulesSystemLibrary.js file or in the RequestRulesUserLibrary.js file.

Tip: To view a list of the available JavaScript functions, their descriptions, syntax, and usage, to the right of the **Logic** list, click **?**. This list automatically includes all functions defined in both the RequestRulesSystemLibrary.js and RequestRulesUserLibrary.js files.

18. Click **OK**.
19. Click **Save**.

SQL Rules: Using Functions of KNTA_USER_UTIL Package

You can use KNTA_USER_UTIL functions to enable the following features:

- ["Copying Date Field Value" on page 162](#)
- ["Converting Request Parameter or Request Detail Parameter to Number" on page 165](#)

KNTA_USER_UTIL Functions

The KNTA_USER_UTIL package includes the following functions:

- KNTA_USER_UTIL.to_char(user, option, dateObject)

This function converts a date object to a String according to the specified user's regional settings.

The return type is string.

This function takes the three parameters listed in the following table:

Parameter	Type	Description
user	String	Current user. For example, [SYS.USERNAME]
option	String	Date format option. Valid values: LONG, MEDIUM, SHORT

Parameter	Type	Description
dateObject	Date (Oracle date object)	Result of KNTA_USER_UTIL.to_date and KNTA_USER_UTIL.next_date, or result of Oracle built-in functions like sysdate, to_date(), add_months(), and so on.

- KNTA_USER_UTIL.to_date(user, option, dateString)

This function converts a date string to an Oracle date object according to the specified user's regional setting.

The return type is Oracle date object.

This function takes the three parameters listed in the following table:

Parameter	Type	Description
user	String	Current user. For example, [SYS.USERNAME]
option	String	Date format option. Valid values: LONG, MEDIUM, SHORT
dateString	String (of a specific date format)	Date string of the specified user's default date format. Use [REQD.VP.FIELDNAME] of a date field for this parameter. Note: Date Time and Time validation fields are not supported.

- KNTA_USER_UTIL.next_date(user, option, dateString, dateGap)

This function converts a date string to an Oracle date object according to the specified user's regional setting. Users can change the date to another using a date gap.

The return type is Oracle date object.

This function takes the four parameters listed in the following table:

Parameter	Type	Description
user	String	Current user. For example, [SYS.USERNAME]
option	String	Date format option. Valid values: LONG, MEDIUM, SHORT
dateString	String (of a specific date format)	Date string of the specified user's default date format. Use [REQD.VP.FIELDNAME] of a date field for this parameter. Note: Date Time and Time validation fields are not supported.
dateGap	Integer	Set the date to several days after if the dateGap value is a

Parameter	Type	Description
		positive integer, or several days before if the dateGap value is a negative integer.

- `KNTA_USER_UTIL.tonumber(string)`

This function fetches and converts a String to a Number in English format (with comma `,` as thousandths separator and dot `.` as decimal separator). For example, 12,123,123.12.

Apart from not taking session NLS into account, this function is similar to the system function `to_number`.

The return type is number.

This function takes one parameter as described below:

Parameter	Type	Description
string	String (of a value)	A string of value.

Copying Date Field Value

To enable formatting dates based on users' current regional settings when copying date field values, you can use `KNTA_USER_UTIL` functions in request type SQL defaulting rules:

- `KNTA_USER_UTIL.to_char(user, option, dateObject)`
- `KNTA_USER_UTIL.to_date(user, option, dateString)`
- `KNTA_USER_UTIL.next_date(user, option, dateString, dateGap)`

Following are examples of SQL defaulting rules with SQL-default logic using `KNTA_USER_UTIL` functions:

- To copy a value from **DATE1** field to **DATE2** field.

DATE1: Date (Long) validation

DATE2: Date (Long) validation

Specify SQL defaulting rule as follows:

```
Select KNTA_USER_UTIL.to_date('[SYS.USERNAME]','LONG','[REQD.VP.DATE1]'), '[REQD.VP.DATE1]' from dual
```

- To copy a value from **DATE1** field to **DATE2** field.

DATE1: Date (Medium) validation

DATE2: Date (Long) validation

Specify SQL defaulting rule as follows:

```
Select KNTA_USER_UTIL.to_date('[SYS.USERNAME]','MEDIUM','[REQD.VP.DATE1]'), '[REQD.VP.DATE1]' from dual
```

Where

- KNTA_USER_UTIL.to_date('[SYS.USERNAME]','MEDIUM','[REQD.VP.DATE1]') is used to parse a "medium" date string to a date object
- The VP value in '[REQD.VP.DATE1]' is parsed by the validation. Therefore, there is no need to parse it with an additional SQL statement.

Another example:

DATE1: Date (Long) validation

DATE2: Date (Short) validation

Specify SQL defaulting rule as follows:

```
Select KNTA_USER_UTIL.to_date('[SYS.USERNAME]','LONG','[REQD.VP.DATE1]'), '[REQD.VP.DATE1]' from dual
```

Where

- KNTA_USER_UTIL.to_date('[SYS.USERNAME]','LONG','[REQD.VP.DATE1]') is used to parse a "long" date string to a date object
 - The VP value in '[REQD.VP.DATE1]' is parsed by the validation, and there is no need to parse it with an additional SQL statement.
- To copy a value from **DATE1** field to **TEXT1** field

DATE1: Date validation

TEXT1: Text Field

Specify SQL defaulting rule as follows:

```
Select '[REQD.VP.DATE1]','[REQD.VP.DATE1]' from dual
```

- To copy a value from **TEXT1** field to **DATE1** field

DATE1: Date validation

TEXT1: Text Field

Depending on the regional settings, the **TEXT1** value must be of one of the following user default date formats: LONG, SHORT, or MEDIUM.

In this case, the LONG format option is used.

Specify SQL defaulting rule as follows:

```
Select KNTA_USER_UTIL.to_date('[SYS.USERNAME]','LONG','[REQD.VP.TEXT1]'), '[REQD.VP.TEXT1]' from dual
```

- To copy a value from **DATE1** field (with dateGap of +10) to **DATE2** field

DATE1: Date (Long) validation

DATE2: Date (Long) validation

Specify SQL defaulting rule as follows:

```
Select KNTA_USER_UTIL.next_date('[SYS.USERNAME]','LONG','[REQD.VP.DATE1]','10'),KNTA_USER_UTIL.to_char('[SYS.USERNAME]','LONG',KNTA_USER_UTIL.next_date('[SYS.USERNAME]','LONG','[REQD.VP.DATE1]','10'),) from dual
```

- To copy a value from **DATE1** field (with dateGap of -10) to **DATE2** field

DATE1: Date Format (Long) validation

DATE2: Date Format (Long) validation

Specify SQL defaulting rule as follows:

```
Select KNTA_USER_UTIL.next_date('[SYS.USERNAME]','LONG','[REQD.VP.DATE1]','-10'),KNTA_USER_UTIL.to_char('[SYS.USERNAME]','LONG',KNTA_USER_UTIL.next_date('[SYS.USERNAME]','LONG','[REQD.VP.DATE1]','-10'),) from dual
```

Limitations

This feature is subject to the following limitations:

- It applies to **Date Format** only. Make sure to set **Time Format** to **None**.
- HPE strongly recommends using the VP value instead of the P value in SQL defaulting rules. Because the token engine replaces P object value with its toString() results, the P date object becomes a string similar to 2011-07-12 00:00:00 (JVM determines the format). This unexpected date string from P date object may cause errors.
- The SQL defaulting rules using KNTA_USER_UTIL functions do not support the following regional settings:

- English (India)
- Chinese (Singapore)

Converting Request Parameter or Request Detail Parameter to Number

To enable converting request parameter or request detail parameter to number, you can use the new `KNTA_USER_UTIL` function in request type SQL defaulting rules:

```
KNTA_USER_UTIL.tonumber(string)
```

Example 1

PPM Center changes Oracle session parameter according to the regional settings and saves values for fields in `CODE` parameter (`PARAMETER COLUMN`), not in `MEANING` (`VISIBLE_PARAMETER` column). If you have users coming from countries that use non-English format numbers, for example, using comma as decimal separator, they may encounter invalid number error when they perform portlet queries that involve system function `to_number` and `VISIBLE_PARAMETER` column values.

In this case, as administrators, you can write SQL queries similar to the follows for them using the `KNTA_USER_UTIL.tonumber` function, which always fetches `CODE` parameter values:

```
SELECT sum(KNTA_USER_UTIL.tonumber(NVL(krv.parameter25, '0'))) FROM kcrt_requests_v
krv
WHERE 1=1
```

Example 2

If you want to make value of `Field3` equal to the result of `Field1` multiplying `Field2`, use the following SQL rule:

```
Select KNTA_USER_UTIL.tonumber('[REQD.P.Field1] ')*KNTA_USER_UTIL.tonumber('
[REQD.P.Field2] '),KNTA_USER_UTIL.tonumber('[REQD.P.Field1] ')*KNTA_USER_
UTIL.tonumber('[REQD.P.Field2] ') from dual
```

Configuring Commands for Request Types

Request types can have many commands, and each command can have many command steps. A command can be viewed as a particular function for a request. Copying a file can be one command, and

checking that file into version control can be another. To perform these functions, a series of events must take place. These events are defined in the command steps.

An additional level of flexibility is introduced when some commands must only be executed in certain cases. This is powered by the condition field of the commands and is discussed in "[Command Conditions](#)" on page 168.

Adding Commands to Request Types

To add commands to request types:

1. On the PPM Workbench shortcut bar, click **Demand Mgmt > Request Types**.

The Request Type Workbench opens.

2. Open a request type.
3. In the Request Type window, click the **Commands** tab.
4. Click **New Cmd**.

The New Command window opens.

5. Provide the following information:
 - a. In the **Command** field, type a name for the command.
 - b. In the **Condition** field, you can type a condition that determines whether the command steps are executed. (For more information, see "[Command Conditions](#)" on page 168).
 - c. In the **Description** field, you can type a short description of the command.
 - d. In the **Timeout(s)** field, to change the amount of time the command can run before its process is terminated, replace the default (90 seconds) with a different value (in seconds).

Note: The **Timeout(s)** value is used to abort commands that hang or take too long to run.

- e. To disable this command, for the **Enabled** option, select **No**. Otherwise, leave **Yes** selected.
- f. In the **Steps** field, type the command code.

To help compose the command, you can do the following:

- Click **Tokens** to open the Token Builder and select tokens to add to the command.
- Click **Special Cmd** to open the Special Command Builder and select preconfigured commands to add.

- Click **Show Desc** to display the Description field, in which you can type a description of the command.
6. Click **OK**.
The **Commands** tab lists the new command.
 7. Click **OK**.

Editing Commands of Request Types

To edit a command on a request type:

1. On the PPM Workbench shortcut bar, click **Demand Mgmt > Request Types**.
The Request Type Workbench opens.
2. Open a request type.
3. In the Request Type window, click the **Commands** tab.
4. Click **Edit Cmd**.
The Edit Command window opens.
5. Select the command to edit.
6. Complete the fields described in the following table.

Field Name	Description
Command	Simple name for the command.
Condition	Condition that determines whether the steps for the command are executed or not. (See " Command Conditions " on the next page for more information).
Description	Description of the command.
Timeout(s)	Amount of time the command can run before its process is terminated. This setting is used to abort commands that are hanging or taking too long to run.
Enabled?	Indicates whether the command is enabled for execution.

7. Click **OK**.
The **Commands** tab now lists the edited command.
8. Click **OK**.

Copying Commands in Request Types

To copy a command in a request types:

1. On the PPM Workbench shortcut bar, click **Demand Mgmt > Request Types**.
The Request Type Workbench opens.
2. Open a request type.
3. In the Request Type window, click the **Commands** tab.
4. Select the command to copy.
5. Click **Copy Cmd**.
6. Click **OK**.

Deleting Commands in Request Types

To delete a command in a request types:

1. On the PPM Workbench shortcut bar, click **Demand Mgmt > Request Types**.
The Request Type Workbench opens.
2. Open a request type.
3. In the Request Type window, click the **Commands** tab.
4. Select the command to delete.
5. Click **Remove**.
6. Click **OK**.

Command Conditions

In some cases, you might have to run a different set of commands depending on the context of execution. You can use *conditional commands* to achieve this flexibility. The **Condition** field for a command is used to define the situation under which the associated command steps execute.

Conditions are evaluated as boolean expressions. If the expression evaluates to true, the command is executed. If false, the command is skipped and the next command is evaluated. If no condition is specified, the command is always executed. The syntax of a condition is identical to the WHERE clause of a SQL statement, which allows enormous flexibility when evaluating scenarios. "Table 5-7. Example conditions" below lists some example conditions.

Note: Be sure to place single quotes around string literals or tokens used to evaluate strings.

Table 5-7. Example conditions

Condition	Evaluates to
BLANK	Command is executed in all situations.
{P.P_VERSION_LABEL} IS NOT NULL	Command is executed if the parameter with the token P_VERSION_LABEL in the package line is not null.
{DEST_ENV.ENVIRONMENT_NAME} = 'Archive'	Command is executed when the destination environment is named "Archive."
{AS.SERVER_TYPE_CODE} = 'UNIX'	Command is executed if the application server is installed on a UNIX® machine.

The condition can include tokens. For detailed information about using tokens, see the *Commands, Tokens, and Validations Guide and Reference*.

Configuring Sub-Types for Request Types

To classify a request type further, you can use *sub-types*. For example, a request type for software bugs might list each of the software applications supported by the IT organization as sub-types.

Adding Sub-Types to Request Types

To add sub-types to the request type:

1. On the PPM Workbench shortcut bar, click **Demand Mgmt > Request Types**.
The Request Type Workbench opens.
2. Open a request type.
The Request Type window opens.

3. In the Request Type window, click the **Sub-Types** tab.
4. Click **New**.

The Request Sub-Type window opens.

5. Complete the fields described in the following table.

Field Name	Description
Sub-Type Name	The name of the sub-type.
Description	A description of the sub-type.
Enabled	Select to make the sub-type available to the system. Select Yes to make the sub-type available to the system.

6. Click **OK**.
7. From the **Sub-Types** tab, click **OK**.

Editing Sub-Types for Request Types

To edit a sub-type:

1. On the PPM Workbench shortcut bar, click **Demand Mgmt > Request Types**.

The Request Type Workbench opens.

2. Open a request type.

The Request Type window opens.

3. Click the **Sub-Types** tab.
4. Select a subtype, and then click **Edit**.

The Request Sub-Type window opens.

5. Complete the fields described in the following table.

Field Name	Description
Sub-Type Name	Name of the sub-type.
Description	Description of the sub-type.
Enabled	Select Yes to make the sub-type available to the system.

6. Click **OK**.
7. On the **Sub-Types** tab, click **OK**.

Deleting Subtypes from Request Types

To delete subtypes from a request type:

1. On the PPM Workbench shortcut bar, click **Demand Mgmt > Request Types**.
The Request Type Workbench opens.
2. Open a request type.
3. In the Request Type window, click the **Sub-Types** tab.
4. Select the sub-type to delete, and then click **Remove**.
5. Click **OK**.

Configuring Request Types to Work with Workflows

You can set up request types to work with all workflows, or with only selected workflows.

Adding Workflows to Request Types

To add workflows to the request type:

1. On the PPM Workbench shortcut bar, click **Demand Mgmt > Request Types**.
The Request Type Workbench opens.
2. Open a request type.
3. In the Request Type window, click the **Workflows** tab.
4. Do one of the following:

- To let all workflows use this request type, select the **All Workflows are allowed for the Request Type** option.
 - To specify the workflows that can use the request,
 - i. Clear the All **Workflows are allowed for the Request Type** option.
 - ii. Click **New**.
The Workflow window opens.
 - iii. In the **Workflow** field, select a workflow.
 - iv. Click **OK**.
5. From the **Workflow** tab, click **OK**.

Deleting Workflows from Request Types

To delete workflows from the request type:

1. On the PPM Workbench shortcut bar, click **Demand Mgmt > Request Types**.
The Request Type Workbench opens.
2. Open a request type.
3. In the Request Type window, click the **Workflows** tab.
4. Select a workflow to delete, and then click **Remove**.
5. Click **OK**.

Configuring Participants for Requests

This section provides instructions on how to give users different levels of access to requests of a given type.

Adding Request Participants to a Request Type

To add request participants to a request type:

1. Log on to PPM Center.
2. On the **Open** menu, click **Administration > Open Workbench**.
The PPM Workbench opens.
3. On the shortcut bar, click **Demand Mgmt > Request Types**.
The Request Type Workbench opens.
4. Open a request type.
5. In the Request Type window, click the **User Access** tab.
6. Click **New**.
The Participant Security window opens.
7. To specify the request type participants, do one of the following:
 - To specify one or more security groups to act on the workflow step:
 - i. From the list at the top of the window, select **Enter a Security Group Name**.
 - ii. Use the **Security Group** auto-complete to select one or more security group names.
 - iii. Click **Add**.
 - To authorize one or more individual users to act on the workflow step:
 - i. From the list at the top of the window, select **Enter a User Name**.
 - ii. Use the **User Name** auto-complete to select one or more user names.
 - iii. Click **Add**.
 - To authorize user or security groups to act on the workflow step using a standard token (that resolves to a list users or security groups):
 - i. From the list at the top of the window, select **Enter a Standard Token**.
 - ii. Use the **Standard Token** auto-complete to select a standard token that returns the resources you want to act on the workflow step.
 - iii. Click **Add**.

The value displayed in the **Security Type** field is based on the token you selected. To add another token, repeat [step i](#) through [step iii](#).
 - To specify a user-defined token that resolves to a list users or security groups:

Note: For information about standard tokens and how to use them, see the *Commands, Tokens, and Validations Guide and Reference*.

- i. From the list at the top of the window, select **Enter a User Defined Token**.
- ii. If the token has already been defined, then in the **User Defined Token** field, type the token name. Otherwise, to open the Token Builder and define a new token that returns the resource(s) you want to act on the workflow step, click **Tokens**.

Note: For information about how to use the Token Builder to create user-defined tokens, see the *Commands, Tokens, and Validations Guide and Reference*.

- iii. In the **Security Type** list, select the security type to which the token resolves.
- iv. Click **Add**.
- v. To add another user-defined token, repeat [step ii](#) through [step iv](#).
- vi. Click **OK**.

The **User Access** tab lists the selected participants.

8. Add the attributes for the participant.

Attributes are attached to a participant by clicking **Create**, **View**, **Edit**, **Cancel**, or **Delete**.

9. On the **User Access** tab, click **OK**.

Editing Participants on Request Types

To edit participants of a request type:

1. On the PPM Workbench shortcut bar, click **Demand Mgmt > Request Types**.

The Request Type Workbench opens.

2. Open a request type.
3. In the Request Type window, click the **User Access** tab.

4. Select a participant to edit, and then click **Edit**.

The Participant Security window opens.

5. Edit the attributes for the participant.

Attributes are attached to a participant by clicking **Create**, **View**, **Edit**, **Cancel**, or **Delete**.

6. On the **User Access** tab, click **OK**.

The changes to the request type are saved.

Deleting Participants from Request Types

To delete participants from the request type:

1. On the PPM Workbench shortcut bar, click **Demand Mgmt > Request Types**.
The Request Type Workbench opens.
2. Open a request type.
3. In the Request Type window, click the **User Access** tab.
4. Select a participant to delete, and then click **Remove**.
5. Click **OK**.

Configuring View and Edit Access Grants for Request Creators

The View and Edit access grants for the creator of a request are configurable. You can restrict the creator's permissions on a request by configuring these access grants.

This is typically used in scenarios like the following:

- The request creators are assigned a new role that should not have enough permission to view or edit the requests they created.
- You delegate the creation of a request to someone who should not be able to view or edit the request after it is created.

To configure the View and Edit access grants for the creator of a request,

1. Open PPM Workbench, and then select **Demand Mgmt > Request Types**.
The Request Type Workbench opens.
2. Open the corresponding request type.
The Request Type window opens.
3. Click the **User Access** tab. Note that the View and Edit access grants for the Created By

participant are configurable.

4. Select or clear the checkboxes for these access grants according to your business needs.

By default, the **View** and **Edit** access grants are selected.

If you uncheck these two access grants, the request creator cannot view or edit the request unless other security settings provide the creator with the View and Edit access grants to the request.

Note: Currently, this enhancement is only available for PPM Center version 8.03, version 9.10 and later. Request types migrated from other versions do not support this feature.

Configuring Import and Export of Requests

This section provides instructions on how to enable users to import requests from XML files as well as export requests to XML files.

Configuring an XML Importable Request Template

An XSLT template is necessary for importing XML files to create new requests or update existing requests.

XSLT Template is a standard XSL file that is used to convert a XML file of user-defined format to an XML data file of PPM format as PPM Center can accept PPM format XML data file only.

Developing an XSL File

If you want to import a request from an XML file with your own format, you need to develop an XSL file to convert the user-defined format file to a PPM format XML data file.

You may refer to the following XSD file to develop your own XSL file:

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="requests">
  <xs:complexType>
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="request">
        <xs:complexType>
```



```

    <xs:sequence>
      <xs:element minOccurs="1" maxOccurs="1" name="requestType"
type="xs:string"/>
      <xs:element minOccurs="0" maxOccurs="unbounded" name="field">
        <xs:complexType>
          <xs:sequence>
            <xs:element minOccurs="0" name="token" type="xs:string"/>
            <xs:element minOccurs="0" name="tableValue">
              <xs:complexType>
                <xs:sequence>
                  <xs:element minOccurs="0" maxOccurs="unbounded"
name="tableColumn">
                    <xs:complexType>
                      <xs:sequence>
                        <xs:element minOccurs="0" name="columnToken"
type="xs:string"/>
                        <xs:element minOccurs="0" maxOccurs="unbounded"
name="cellValue" type="xs:string"/>
                      </xs:sequence>
                    </xs:complexType>
                  </xs:element>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
            <xs:element minOccurs="0" name="value" type="xs:string"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:schema>

```

Note: This XSD file defines the XML format accepted by PPM Center's generic request import operation. When you develop an XSL template, make ensure that the XML file generated using the XSL template conforms to the format defined by the XSD file. Do not use this file as a template directly.

If you import a request from an XML file without providing any XSL template, you need to ensure that your XML file conform with the format of PPM XML data file.

Configuring an XSLT Template

When exporting a request, you can define an XSLT template to convert the raw data XML file to the desired format, such as XML, HTML, Text, and CSV. The XSLT template is a standard XSL file, it can convert the raw XML data file in PPM format.

For more technical details about the format of PPM XML data file, see the XSD file described in ["Developing an XSL File" on page 176](#).

To configure an XSLT template,

1. Log on to PPM Center.
2. On the **Open** menu, click **Administration > Open Workbench**.
The PPM Workbench opens.
3. On the shortcut bar, click **Configuration > Validations**.
The Validation Workbench opens.
4. Click **List**, then locate and open the **XML Importable Request Template** validation.
The Validation: XML Importable Request Template window opens.
5. Click **New**.
The Add Validation Value dialogue box opens to Value Information tab.
6. On the Value Information tab, provide values for required **Code** and **Meaning** fields and optional field **Desc** as necessary.
The value of **Meaning** field will be displayed in the drop-down list for **XSLT Template** field on the Import Request from XML page.
7. To use this new template, for the **Enabled?** field, leave it selected.
8. If you want to set this new template default, select the checkbox for **Default** field. Otherwise, leave it empty.
For more information about the fields in the Add Validation Value dialogue box, see the *Configuring Static List Validations* section of the *Commands, Tokens, and Validations Guide and Reference*.
9. Click the **User Data** tab.
10. Click **Add**.

The Add Document dialog box opens.

11. Do the following:
 - a. Click **Browse** to locate and select the XSLT file you want to add. Normally it is the file you developed in ["Developing an XSL File" on page 176](#).
 - b. In the **Description** field, you can type a short description for the XSLT file.
 - c. click **Add**.
12. To add more values and keep the Add Validation Value window open, click **Add**.
13. To save your changes and close the window, click **OK**.

Exporting Requests to XML Files

Exporting a request to an XML file of a specific format involves the following tasks,

1. ["Enabling Report Type \(REFERENCE\) Export Request Report" below](#)
2. ["Configuring an XML Exportable Request Template" on the next page](#)

Enabling Report Type (REFERENCE) Export Request Report

To facilitate exporting a request to XML of a specific format, a new report type (**REFERENCE**) **Export Request Report** is added. This report type enables you to export a request report to a specific format.

To enable the report type,

1. Log on to PPM Center and launch the PPM Workbench.
2. On the shortcut bar, click **Configuration > Report Types**.

The Report Type Workbench opens.

3. Click **List**, then locate and select report type (**REFERENCE**) **Export Request Report**.
4. Click **Copy**.

The Copy Report Type window opens.

5. Type a **Report Type Name**. For example, `Sample Report`.
6. Click **Copy** to continue.
7. When prompted, click **Yes**.

The Report Type: *<Report Type Name>* window opens.

8. For the **Enabled** field, select **Yes**.
9. Configure other fields or options as necessary.
10. Click **OK**.

Configuring an XML Exportable Request Template

To configure a specific format template for exporting a request to,

1. From the Validations Workbench, select validation **XML Exportable Request Template**.
2. Click **New**.

The Add Validation Value dialogue box opens to Value Information tab.

3. On the Value Information tab, provide values for required **Code** and **Meaning** fields and optional field **Desc** as necessary.

The value of **Meaning** field will be displayed in the drop-down list for **XSLT Template** field on the Import Request from XML page.

4. To use this new template, for the **Enabled?** field, leave it selected.
5. If you want to set this new template default, select the checkbox for **Default** field. Otherwise, leave it empty.

For more information about the fields in the Add Validation Value dialogue box, see the *Configuring Static List Validations* section of the *Commands, Tokens, and Validations Guide and Reference*.

6. Click the **User Data** tab.
7. Do the following:
 - a. In the **Request Type ID** field, select a request type ID using the auto-complete icon.
 - b. In the **Content Type** field, you can type a short description of the request type.
 - c. For **XSLT File** field, click **Add**.

The Add Document dialog box opens.

- i. Click **Browse** to locate and select the XSLT file you want to add.
 - ii. In the **Description** field, you can type a short description for the XSLT file.
 - iii. click **Add**.
8. To add more values and keep the Add Validation Value window open, click **Add**.
 9. To save your changes and close the window, click **OK**.

Exporting Requests by Running the Special Command

In general, a request is exported via the PPM Center report. A new special command (`ksc_export_request_as_xml`) is introduced to generate the exported file for the report. Setting `true` indicates report mode.

Report mode example:

```
ksc_export_request_as_xml [P.TARGET_REQUEST] "[P.TEMPLATE]" true [RP.REPORT_
SUBMISSION_ID]
```

where,

[P.TARGET_REQUEST] – ID of the request to be exported.

[P.TEMPLATE] – The template code.

`true` – Report mode, indicates that the result is exported to report.

[RP.REPORT_SUBMISSION_ID] – Report submission ID.

You can also use this special command in request mode by integrating it into your own workflows. In this case, the special command generates the exported file and pastes the URL of the exported file to a corresponding field on a specified request. A URL field is necessary for this mode because it shows more readable file name while referring to the full address of the exported file.

Request mode example:

```
ksc_export_request_as_xml [P.TARGET_REQUEST] "[P.TEMPLATE]" false [REQ.REQUEST_ID]
[REQD.P.RESULT]
```

where,

[P.TARGET_REQUEST] – ID of the request to be exported.

[P.TEMPLATE] – The template code.

false – Non-report mode, indicates that the URL of the exported file is updated to an appropriate field of the request.

[REQ.REQUEST_ID] – ID of the request that contains URL of the exported file.

[REQD.P.RESULT] – The field token that contains the URL of the exported file.

Note: When you use the special command in a workflow to export a request, make sure to set **Workflow Scope to Requests** in the corresponding execution step. Otherwise you will not get an exported file successfully.

Configuring Quick Edit and Mass Update

To disable both the quick edit and the mass update features, you can set the server configuration parameter `DISABLE_QUICK_EDIT_MASS_UPDATE` to `true`. The default value is `false`.

Configuring Resource Tracking

As a request is processed through its workflow, it is typically assigned to several different users who are responsible for it at different process stages. For some types of requests, it is important to keep track of all the users who were assigned on a request, and optionally, how much time they were expected to spend, and how much time they actually spent, working on the request. For this purpose, you can enable resource tracking for a request type.

Note: For information on how to assign resources by configuring workflow steps, see "[Configuring Workflows](#)" on page 38.

For information on how to use Time Management to track time spent on requests, see "[Configuring Request Types for Use with Time Management](#)" on page 185.

Tracking Resources Assigned to Requests

In Demand Management, you can assign a resource to a request. A request is typically assigned to different resources at different phases of its workflow.

You can configure request types to keep track which resource has been assigned to a request, what the scheduled effort is for the resource, how much work they actually contributed to the request and when by enabling resource tracking. After you enable resource tracking, you can configure security on the request type to determine who can edit the list of resources, allocate new resources, and so on.

The resource tracking information that you enable becomes available to authorized users based on the access you set for the request type. These authorized users can maximize the **Resources** section on the request detail page to view the information. (Without management capabilities, individual resources can only update their own actuals on the request details page.)

Note: The **Resources** section is also controlled by Time Management in that, if resources are logging time sheets for the request, then the actual effort, actual start, and actual finish come from the time sheet.

In this case, resources are not allowed to manually change the actuals data directly on the request; they must go to the respective time sheet(s) to update their actuals. For more details on how to use Time Management in conjunction with requests, see "[Configuring Request Types for Use with Time Management](#)" on page 185.

To configure resource tracking on requests:

1. On the PPM Workbench shortcut bar, click **Demand Mgmt > Request Types**.

The Request Type Workbench opens.

2. Open a request type.
3. In the Request Type window, click the **Resources** tab.
4. Select the **Track resources** checkbox.
5. Use the **Resource: Validation** auto-complete to select the validation to use for the **Add Resource** button in the **Resources** section of the request details page.

Note: If you do not specify a validation, then the validation used for the request type's **Assigned To** field is used.

6. To select the fields to track for requests of this type, select any or all of the following checkboxes:
 - o **Scheduled Effort**

Note: A resource who manages the request can use this **Scheduled Effort** field in the **Resources** section of the request detail page to allocate resources. If the request type also includes the Work Item field group, then this allocation will appear as workload for the resource in his Gantt view and in other visualizations of assignment workload used by

project managers or resource managers. If, in addition, Time Management is used to track actuals, then these allocations will show up in the **Expected Hours** field on the time sheet for the allocated resource.

- **Actual Effort**
- **Actual Start and Actual Finish**

The **Resources** section on the request details page will display a column for the resource names, and for each of the fields you selected.

The **Resource Security** section lets you specify who can manage the **Resources** section on the request details page. (Individual resources can update their own actuals, if they have access to the request.)

7. To allow all users who can edit the request to create, edit, or remove resources on the request, select the **All users who can edit request** checkbox.
 - To specify security groups that can manage request resources:
 - i. Click **New**.

The Participant Security window opens.
 - ii. To specify one or more security groups to manage request resources:
 - A. From the list at the top of the window, select **Enter a Security Group Name**.
 - B. Use the **Security Group** auto-complete to select one or more security group names.
 - C. Click **Add**.
 - To authorize one or more individual users to manage request resources:
 - i. From the list at the top of the window, select **Enter a User Name**.
 - ii. Use the **User Name** auto-complete to select one or more user names.
 - iii. Click **Add**.
 - To authorize users or security groups to manage request resources using a standard token (that resolves to a list of users or security groups):
 - i. From the list at the top of the window, select **Enter a Standard Token**.
 - ii. Use the **Standard Token** auto-complete to select a standard token that returns the resources.
 - iii. Click **Add**.
 - iv. The value displayed in the **Security Type** field is based on the token you selected. To

add another token, repeat [step i](#) through [step iii](#).

Note: For information about standard tokens and how to use them, see the *Commands, Tokens, and Validations Guide and Reference*.

- o To specify a user-defined token that resolves to a list users or security groups:
 - i. From the list at the top of the window, select **Enter a User Defined Token**.
 - ii. If the token has already been defined, then in the **User Defined Token** field, type the token name. Otherwise, to open the Token Builder and define a new token that returns the resource(s) you want to act on the workflow step, click **Tokens**.

Note: For information about how to use the Token Builder to create user-defined tokens, see the *Commands, Tokens, and Validations Guide and Reference*.

- iii. In the **Security Type** list, select the security type to which the token resolves.
 - iv. Click **Add**.
 - v. To add another user-defined token, repeat [step ii](#) through [step iv](#).
 - vi. Click **OK**.

The **User or Group** field on the **Resources** tab lists your selection(s).

Configuring Request Types for Use with Time Management

In PPM Center, *actuals* represent the number of hours a resource has worked on an activity or request, and the dates on which that work was done. You can enable tracking of this data in Time Management from the **Resources** tab. Once you enable the request type for Time Management, you can specify at a more granular level who can log time against a request of that type.

To configure tracking of actuals in Time Management:

1. On the PPM Workbench shortcut bar, click **Demand Mgmt > Request Types**.

The Request Type Workbench opens.

2. Open a request type.
3. In the Request Type window, click the **Resources** tab.
4. Expand the Request Type window so that you can view the entire **Resources** tab.

5. Select the **Use Time Management to track actuals** checkbox.
6. To specify who can log time against a request of this type, do the following:
 - a. To specify resources who are directly assigned to requests of this type during the current time sheet time period, select the **Resources assigned to the request during the current time period** checkbox.

To successfully enable assigned resources to log time against a request of this type, you must also perform the following steps:

- i. From the PPM Workbench, open the request header type used.
 - ii. On the **Attributes** tab of the **Assigned To** field, set the **Transaction History** attribute to **Yes**.
- b. If you enabled resource assignment tracking, and you want to select all of the resources listed in the new resources tracking section, select the **All resources listed on the request** checkbox.
 - c. To select users who belong to the assigned security group, select the **All users in the assigned group** checkbox.

To successfully enable assigned security group members to log time against a request of this type, you must also perform the following steps:

- i. From the PPM Workbench, open the request header type used.
 - ii. On the **Attributes** tab of the **Assigned Group** field, set the **Transaction History** attribute to **Yes**.
- d. To allow all request participants (as specified through workflow security and the participant model) to log time against a request of this type, leave the **All request participants** checkbox selected. Otherwise, clear this checkbox.
 - e. If this is an Asset request type, for which you use staffing profiles to allocate resources for an asset, you can select the **All resources allocated in the staffing profile** checkbox.

7. Click **Save**.

Note: Any time the **Assigned To** field value changes in the request details page (any time the request is assigned to a different user), that new value is automatically added to the **Resources** section of the page, and is a resource on the request.

A resource who is managing the request can use the **Scheduled Effort** field in the **Resources** section to allocate resources. These allocations will be displayed in the **Expected Hours** field on the time sheet for that resource.

Configuring Notifications for Request Types

You can configure a request type to send notifications based on field contents. You can send notifications to different recipients at different times, at different intervals, and based on different events.

Adding Notifications

To add a notification:

1. On the PPM Workbench shortcut bar, click **Demand Mgmt > Request Types**.

The Request Type Workbench opens.

2. Open a request type.
3. In the Request Type window, click the **Notifications** tab.
4. Click **New**.

The Add Notification for Step window opens.

5. Configure the **Setup** tab.

For information about how to configure the **Setup** tab, see ["Configuring the Setup Tab" below](#).

6. Configure the **Message** tab.

For information about how to configure the **Message** tab, see ["Configuring Message Tab" on page 190](#).

7. Click **OK**.

The **Notifications** tab lists the notifications added.

8. Click **OK**.

Configuring the Setup Tab

To configure the **Setup** tab:

1. On the PPM Workbench shortcut bar, click **Demand Mgmt > Request Types**.

The Request Type Workbench opens.

2. Open a request type.
3. In the Request Type window, click the **Notifications** tab.
4. Click **New**.

The Add Notification for Step window opens to the **Setup** tab.

5. In the **Options** section, complete the fields described in the following table.

Field Name	Description
Description	Brief description of the notification.
Event	Type of event that triggers notification transmission. Field Changes is the default and cannot be edited.
Interval	<p>A notification can be sent at different intervals. For example, you might choose to send a notification of a final approval step at midnight so that it is ready for approval in the morning.</p> <p>Note also that multiple notifications to a single recipient can be brought together in a batch and sent together. Selecting an interval other than Immediate allows this batch and send to occur.</p> <p>The available interval options are:</p> <ul style="list-style-type: none"> ○ 8:00AM Daily M-F ○ Hourly Daily M-F ○ Immediate
Field	Selects the request type field that triggers the notification from the list. If a change occurs in the selected field, the notification is sent.
Specific Value	Send the notification when the selected field is the specified value. Selecting Specific Value enabled the text field. Type the value in the text field. Selecting Specific Value clears Any Value and No Value .
Any Value	Send the notification when the selected field is changes to any value. Selecting Any Value clears Specific Value and No Value .
No Value	Send the notification when the selected field is empty. Selecting No Value clears Specific Value and Any Value .
Send on Request Submission	Send the notification when the request is first submitted.

Field Name	Description
Enabled	Make the notification available to the system. Selecting Yes makes the notification available to the system.
Don't send if obsolete	Do not send the notification if the trigger values are no longer true. For repeating messages: <ul style="list-style-type: none"> ○ 8:00AM Daily M-F ○ Hourly Daily M-F For example, if a notification is sent hourly when the field is empty, the notification will automatically stop when the field has a value.

6. To configure the **Recipients** section:

- a. In the **Recipients** section, click **New**.

The Add New Recipient window opens.

- b. Click **To**, **Cc**, or **Bcc**.

- c. To specify the recipient:

- i. **Enter a Username.** Select a user as the recipient of the notification. Selecting a user changes the name of the auto-complete to **Username**. The security type dynamically changes to **Username**.
- ii. **Enter an Email Address.** Select an email address as the recipient of the notification. Selecting an email address changes the name of the auto-complete to **Email Address**. The security type is dynamically changed to **Email Address**.
- iii. **Enter a Security Group.** Select a security group as the recipient of the notification. Selecting a security group changes the name of the auto-complete to **Security Group**. The security type is dynamically changed to **Security Group**.
- iv. **Enter a Standard Token.** Select a standard token to act upon the workflow step. Selecting a standard token changes the name of the auto-complete to **Standard Token**. The security type is left undefined. Select a standard token from the auto-complete. The **Security Type** field is defined based on the standard token chosen.
- v. **Enter a User Defined Token.** Select a user defined token to act upon the workflow step. Selecting a user defined token changes the name of the auto-complete to **User Defined Token**. The security type is dynamically changed to a list. The **Tokens** button is enabled. Click **Tokens** to open the Token Builder window and select a token. Select one of the following from the list:

- **Username.** The selected token resolves to a username.
 - **User ID.** The selected token resolves to a user ID.
 - **Security Group Name.** The selected token resolves to a security group.
 - **Security Group ID.** The selected token resolves to a security group ID.
- d. Click **OK**.
7. On the **Setup** tab, click **OK**.

Configuring Message Tab

You can construct the notification's message to ensure that it contains the correct information for the recipient. For example, if a notification is sent to instruct you that a request requires your approval, the message should instruct you to log onto PPM Center and update the request status. Additionally, the notification should include a link (URL) to the referenced request.

The following features to make notifications simpler to configure and use:

- Select from a number of preconfigured notification templates to more quickly construct the body of your message.
- The body of the notification can be plain text or HTML.
- Multiple tokens can be included in the notification. These tokens will resolve to information relevant to the recipient. For example, you can include tokens for the URL to the request approval page, information on request status and priority, and emergency contacts.

To configure the **Message** tab:

1. On the PPM Workbench shortcut bar, click **Demand Mgmt > Request Types**.
The Request Type Workbench opens.
2. Open a request type.
3. In the Request Type window, click the **Notifications** tab.
4. Click **New**.
The Add Notification for Step window opens to the **Setup** tab.
5. Click the **Message** tab.
6. In the **Notification Template** field, select a template.

This updates the contents in the **Body** section with the information defined in the selected template.

7. In the **Notification Format** field, select the message format.

The HTML format provides more flexibility in creating the look and feel of the notification. You can write and test the HTML code in any HTML editor, and then paste the content into the Body window.

8. Provide values for the **From** and **Reply to** fields, as follows:

- a. To the right of the **From** or **Reply to** field, click **Choose**.

The Email Header Field window opens.

- b. Select the notification recipient(s), as follows:

- **Enter a Username.** Select a user as the recipient of the notification. Selecting a user changes the name of the auto-complete to **Username**. The security type is dynamically changed to **Username**.
- **Enter an Email Address.** Select an email address as the recipient of the notification. Selecting an email address changes the name of the auto-complete to **Email Address**. The security type is dynamically changed to **Email Address**.
- **Enter a Standard Token.** Select a standard token to act upon the workflow step. Selecting a standard token changes the name of the auto-complete to **Standard Token**. The security type is left undefined. Select a standard token from the auto-complete.
- **Enter a User Defined Token.** Select a user defined token to act upon the workflow step. Selecting a user defined token changes the name of the auto-complete to **User Defined Token**. The **Tokens** button is enabled. Click **Tokens** to open the Token Builder window and select a token.

Select one of the following from the list:

- **Username.** The selected token resolves to a username.
- **User ID.** The selected token resolves to a user ID.
- **Security Group Name.** The selected token resolves to a security group.
- **Security Group ID.** The selected token resolves to a security group ID.
- **Email Address.** The selected token resolves to an email address.

- c. Click **OK**.

The **Message** tab lists the selected recipients.

9. Construct the body of the message.

When constructing the body, consider using the following:

- Token for the URL to the Request Detail page.
- Token for the URL to the package (PPM Workbench or standard interface).
- Tokens in the body of the message:

Click **Tokens** to access the Token Builder window where tokens can be added to the message body.

- Tokens related to specific package lines:

Add tokens to the **Linked Token** field to include tokens that resolve information related to the individual package line.

10. Click **OK**.
11. From the **Notifications** tab, click **OK**.

Editing Notifications

To edit a notification:

1. On the PPM Workbench shortcut bar, click **Demand Mgmt > Request Types**.

The Request Type Workbench opens.

2. Open a request type.
3. In the Request Type window, click the **Notifications** tab.
4. Select a notification that you want to change, and then click **Edit**.

The Add Notification for Step window opens to the **Setup** tab.

5. Edit the **Setup** tab (see "[Configuring the Setup Tab](#)" on page 187).
6. Edit the **Message** tab (see "[Configuring Message Tab](#)" on page 190).
7. Click **OK**.

Copying Notifications

To copy a notification:

1. On the PPM Workbench shortcut bar, click **Demand Mgmt > Request Types**.

The Request Type Workbench opens.

2. Open a request type.
3. In the Request Type window, click the **Notifications** tab.
4. Select the notification you want to copy, and then click **Copy**.

The Add Notification for Step window opens to the **Setup** tab.

Note: For information about how to edit the **Setup** tab, see ["Configuring the Setup Tab" on page 187](#). For information about how to edit the **Message** tab, see ["Configuring Message Tab" on page 190](#).

5. On the **Notifications** tab, click **OK**.

The changes to the request type are saved.

Deleting Notifications

To delete a notification:

1. On the PPM Workbench shortcut bar, click **Demand Mgmt > Request Types**.

The Request Type Workbench opens.

2. Open a request type.
3. In the Request Type window, click the **Notifications** tab.
4. Select a notification that you want to remove, and then click **Delete**.
5. Click **OK**.

Configuring Ownerships of Request Types

To configure request type ownership groups, you add security groups to the **Ownership** tab. If no ownership groups are associated with the entity, the entity is treated as global, and any user who can edit request types can edit, copy, or delete the entity. For more information about access grants, see the *Security Model Guide and Reference*.

If a security group is disabled or loses its ability to edit a request type, that group can no longer edit the entity.

Adding Ownerships to Request Types

To add an ownership:

1. On the PPM Workbench shortcut bar, click **Demand Mgmt > Request Types**.

The Request Type Workbench opens.

2. Open a request type.
3. In the Request Type window, click the **Ownership** tab.
4. Select the ownership option.

The **All users with the Edit Request Type access grant** option gives all users who can edit request types access to the request type. The **Only groups listed below that have the Edit Request Type access grant** option requires selected groups to be added to the ownership of the request type.

If you select, **Only groups listed below that have the Edit Request Type**, complete the following:

- a. On the **Ownership** tab, click **Add**.

The Add Security Groups window opens.

- b. In the **Security Groups** field, select the security groups.

The Validate window opens.

- c. Select one or more security groups, and then click **OK**.

The Add Security Groups window lists the selected security groups.

- d. Click **OK**.

From the **Ownership** tab, the **Security Group** column lists the selected security groups.

5. Click **OK**.

Deleting Ownerships from Request Types

To delete an ownership:

1. On the PPM Workbench shortcut bar, click **Demand Mgmt > Request Types**.

The Request Type Workbench opens.

2. Open a request type.
3. In the Request Type window, click the **Ownership** tab.
4. Select an ownership.

The **All users with the Edit Request Type access grant** option gives all users who can edit request type access to the request type. The **Only groups listed below that have the Edit Request Type access grant** option requires selected groups to be added to the ownership of the request type.

5. Click **Remove**.
6. Click **OK**.

Configuring Help Contents for Request Types

You can provide accessible online information to users who are processing the requests. Configure the request type to display additional, custom information about the request, sections or fields.

To add help to the request type:

1. On the PPM Workbench shortcut bar, click **Demand Mgmt > Request Types**.

The Request Type Workbench opens.

2. Open a request type.
3. In the Request Type window, click the **Help Content** tab.

4. In the **Sections/Fields** section, select the item to which content is to be added.
5. In the **Help Content for Request Type** section, type the help content for the selected item.
Type plain text or HTML-formatted text.
6. To preview the help display, click **Preview**.
7. Provide additional, optional information to further define help content for those items.
8. From the **Display Help Icons at the:** field, specify how the help icons are to be displayed in the standard interface.
 - **Request, Section and Field Level.** Display a help icon (question mark) beside each request, section and field that has associated help content.
 - **Request and Section Level Only.** Does not display the help icon at the individual field level. Any help content defined for the fields can be accessed from the section level help.
9. From the **Help Content** tab, click **Save**.

Configuring Request Header Types

Request header types define the collection of fields that appear in the header region of the requests. Request header types typically include more general information that is tracked between multiple types of requests. This can include such information as who logged the request, its priority, and a description of the issue.

Every request type must include a request header type. A single request header type can be used for multiple request types.

"[Table 5-8. Request header types](#)" below lists the HPE-supplied request header types.

Table 5-8. Request header types

System Header Type (REFERENCE)	Description
Default	Default request header type. Includes a percentage complete (% Complete) field.
Comprehensive	Displays all information. Consistent with previous versions of PPM Center.
Simple	Displays only the most essential information.

Table 5-8. Request header types, continued

System Header Type (REFERENCE)	Description
Departmental	Example request header type for simple cross-departmental requests.
Application	Example request header type for simple cross-application requests.
Help Desk	Example request header type for help desk requests, including contact and assignment information.

Overview of Request Header Types

Request header types contain a set of standard predefined fields that can be enabled or disabled. Request header types can also contain custom fields. Request header types are created and configured in the Request Header Type window ("Figure 5-7. Request Header Type window" below).

Figure 5-7. Request Header Type window

Request Header Type : Default

Request Header Type Name: Default

Reference Code: _DEFAULT

Description: Default Request Header Type

Extension: Enabled: Yes No

Fields | Layout | Filter | Ownership | User Data | References

Prompt	Displ...	Display Only	Transaction Hist.	Notes Hist.	On Search/Filter Pag
[-] Summary					
Request No.:	Y	Y	N	N	N
Request Type:	Y	N	N	N	Y
Created By:	Y	Y	N	N	Y
Department:	Y	N	N	N	Y
Sub-Type:	Y	N	N	N	Y
Created On:	Y	Y	N	N	Y

+ All - All New Edit Remove Field Groups

OK Save Cancel

Ready

The main components of a request header type are as follows:

- **General information.** General information includes basic information concerning the request type, such as the request type name and the request type category. See ["Configuring General Information for Request Header Types" on page 201](#).
- **Fields.** Every request header type has a set of predefined fields. The **Fields** tab is used to create additional fields for the request header type. Creating fields for a request header type is identical to creating fields for request types. See ["Creating and Configuring Request Type Fields" on page 118](#).

Note: PPM Center stores Request Header Type (RHT) fields in the KCRT_REQ_HEADER_DETAILS table, which has only 50 columns available.

You cannot create a Table Component type field for a request header type.

- **Layout.** The layout of fields can be configured using the **Layout** tab. Laying out fields for request header types is identical to laying out fields for request types. See ["Configuring Layouts for Request Types" on page 127](#).
- **Filter.** Several fields on request header types can be filtered to display specific information in a request. See ["Configuring Filters for Request Header Types " on page 202](#).
- **Ownership.** Configure who can edit the request header type. Configuring who can edit the request header type is identical to configuring who can edit a request type. See ["Configuring Ownerships of Request Types" on page 194](#).
- **User Data.** Product entities such as packages, workflows, requests and projects include a set of standard fields that provide information about those entities. While these fields are normally sufficient for day to day processing, user data fields provide the ability to capture additional information specific to each organization. User data is defined under the **User Data** tab. If there are no user data fields, the **User Data** tab is disabled.
- **References.** Displays reference information for the request header type.
- **Field Groups.** Request header type field groups are a way for PPM Center to distribute a collection of fields required for certain functionality. For more information, see ["Request Header Type Field Groups " below](#).

Request Header Type Field Groups

Request header type field groups are a way for PPM Center to distribute a collection of fields required for certain functionality. For example, Demand Management distributes a collection of fields for service level agreements in the Demand Management SLA Fields field group.

Field group fields behave just as normal fields do, with the restrictions that you cannot remove them except by removing the entire field group and you might not be able to modify some of the field properties. "Table 5-9. Request header type field groups" below lists the request header type field groups that are delivered with various PPM Center products.

You can add field groups to request header types by clicking **Field Groups** in the Request Header Type window.

Each request header type field group has a custom token prefix that allows the user to access the data of that field by using the format:

```
REQ.P.<Field_Group_Token_Starting_With_KNTA_>
```

When field groups are associated with existing request types (through the request header type definition), PPM Center database tables are updated to handle this new configuration. Because of the scope of database changes, the Database Statistics should be rerun on your database. Instructions for this are included in the *Installation and Administration Guide*. Contact the application administrator for help with this procedure.

Table 5-9. Request header type field groups

Field Group	Description
CMQC Application Project	CMQC Application Project (Also include PFM Project and CMQC QC/ALM Instance field groups)
CMQC Project Status	CMQC Project Status field group
CMQC QC/ALM Administration	CMQC QC/ALM Administration field group
CMQC QC/ALM Instance	CMQC QC/ALM Instance field group
CMQC Test Status	CMQC Test Status field group
CMQC Testing Project	CMQC Testing Project (Also include PFM Project and CMQC QC/ALM Instance field groups)
Demand Management SLA Fields	This Field Group contains the fields necessary to manage requests with SLA.
Demand Management Scheduling Fields	This Field Group allows a request to be scheduled with the Demand Management solution.
PFM Asset	Allows Requests to be considered as Assets in a Portfolio.
PFM Project	Required for any request type that will represent a project lifecycle (regardless of whether the project will be considered in the portfolio) If the PPM Center system does not include the PFM - Project request

Table 5-9. Request header type field groups, continued

Field Group	Description
	type, you can create a new request type, add this field group to it, and then use this request type for projects. This field group is required for any request type that models the project lifecycle, regardless of whether the project is part of a portfolio.
PFM Proposal	Allows requests to be considered as Proposals in the Portfolio Management process.
Program Issue	Allows Requests to be considered as Issues in a Program. Note: After you enable field group Program Issue: Allows Requests to be considered as Issues In a Program for a request header type, no Program Issue section is added to the Fields tab of the request header type. This behavior is by design and it has no functional impact. Request types using this request header type can still be used as program issue type and it works fine for program issue tracking.
Program Reference	Contains a field that allows a user to add a Program reference to a request.
Program Risk	Allows Requests to be considered as Risks in a Program.
Project Issue	Allows Requests to be considered as Issues in a Project.
Project Reference	Contains a field that allows a user to add a Project reference to a Request.
Project Risk	Allows Requests to be considered as Risks in a Project.
Project Scope Change	Allows Requests to be considered as Scope Changes in a Project.
QC/ALM Defect Information	This field group contains fields for PPM-QC/ALM Defect integration
QC/ALM Info	Allows Requests to use the special integration with QC/ALM
QC/ALM Release Information	This field group contains fields for QC/ALM Metrics integration
Service	Allows selection to identify the service associated with the request
Universal CMDB Impact Analysis	Enables integration with Universal CMDB
Work Item Fields	These fields allow requests to be tracked as load in Resource Management visualizations.

Opening the Request Header Type Workbench

To open the Request Header Type Workbench:

- On the PPM Workbench shortcut bar, click **Demand Mgmt > Request Header Types**.

The Request Header Type Workbench opens.

Configuring General Information for Request Header Types

To configure the general information of a request header type:

1. On the PPM Workbench shortcut bar, click **Demand Mgmt > Request Header Types**.

The Request Header Type Workbench opens.

2. Open a request header type.

The Request Header Type window opens.

3. Provide the information specified in the following table.

Field Name	Description
Request Header Type Name	The name of the request header type.
Reference Code	Code used to reference this request header type.
Description	A useful description of how the request header type is used.
Extension	For request header types created for an Deployment Management extension, select the extension from the list.
Enabled	Indicates whether or not the request header type is available to PPM Center.

4. Do one of the following:

- To save the changes and close the window, click **OK**.
- To save the changes and leave the window open, click **Save**.

Configuring Filters for Request Header Types

To configure filters for a request header type,

1. On the PPM Workbench shortcut bar, click **Demand Mgmt > Request Header Types**.
The Request Header Type Workbench opens.
2. Open a request header type.
3. In the Request Header Type window, click the **Filter** tab.

4. Provide the information specified in the following table.

Field Name	Description
This section of the Contact Name field is limited by:	<ul style="list-style-type: none"> ○ All Contacts. Limit the number of contact names displayed in the Contact Name field when creating or updating a request header type by selecting one of the contact name options available in the Filter tab. Selecting this option will display all users with no restrictions on the list of contact names. ○ The Company field of the Request. Users can limit the number of contact names shown in the Contact Name field when creating or updating a request header type by selecting one of the contact name options available in the Filter tab. Selecting this option will restrict the displayed list of contact names shown to those found in the Company field of the request. ○ Use Validation defined in the Fields tab. Selecting this option will restrict the displayed list of contact names shown to those found in the Contact Name field of the request.
This section of the Assigned Group Field is limited by:	<ul style="list-style-type: none"> ○ Only Security Groups with the Request option enabled. Users can limit the number of group names shown when creating or updating a request header type by selecting one of two Assigned Group options available on the Filter tab. Selecting this option will restrict the displayed list of group names shown to only those security groups where the request option is enabled. ○ Participants only. Users can limit the number of group names they would see when creating or updating a request header type by selecting one of two Assigned Group options available on the Filter tab. Selecting this option will restrict the displayed list of group names shown to participants in the request. ○ Use Validation defined in the Fields tab. Selecting this option will restrict the displayed list of contact names shown to those found in the Contact Name field of the request.
This section of the Assigned To field is limited by:	<ul style="list-style-type: none"> ○ Only users who are in Security Groups with the Request option enabled. Limit the number of user names displayed in the Assigned To field when creating or updating a request header type by selecting one of two Assigned To options available in the Filter tab. Selecting this option restricts the displayed list of user names the user would see to only those security groups where the request option is enabled. ○ Participants only. Users can limit the number of user names shown in the Assigned To field when creating or updating a request header type by selecting one of two Assigned To options available in the Filter tab. Selecting this option restricts the displayed list of user

Field Name	Description
	<p>names shown to participants of the request. In this instance, participants are defined as the assigned user, the creator of the request, members of the assigned group, or members of the workflow.</p> <ul style="list-style-type: none">○ Use Validation defined in the Fields tab. Selecting this option will restrict the displayed list of contact names shown to those found in the Contact Name field of the request.

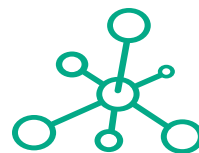
5. Click **OK**.



Chapter 6: Enabling Service for Requests

Project and Portfolio Management Center Service Portfolio Management is designed to add value to your organization. If your organization has adopted the ITIL definitions of services and service lifecycle, you may use this feature to track, categorize, and analyze business services and labor cost related IT project initiatives and requests. This offers possibility and flexibility to your organization in governing and maximizing your investments in business services and managing them for value.

Enabling Service for



Hewlett Packard
Enterprise

Requests

With administrator privileges, you can enable a service field on the request creation page in PPM Center, so that users can associate their requests to a related service when they create and submit the requests.

To enable the **Service** field in the request creation page, perform the following tasks:

1. ["Enable Service Field Group for Request Header Type" below](#)
2. ["Enable Service Field Group for Request Type" on the next page](#)

Then users can specify a service when they create and submit their requests. For information about specifying a service, see the *Demand Management User's Guide*.

Enable Service Field Group for Request Header Type

Before you can enable service field for requests, you need to create a request type for the requests.

To create a request type with service enabled, you need to enable the newly introduced Service field group for its Request Header Type.

You can define a new request header type or modify an existing Request Header Type to meet your business needs.

To define a new request header type:

1. On the PPM Workbench shortcut bar, select **Demand Mgmt > Request Header Types**.
The Request Header Type Workbench opens.
2. Click **New Request Header Type**.
The Request Header Type window opens.
3. Type a name in the Request Header Type Name field, for example, `RHT_service`.
The **Reference Code** field is populated automatically with the value you just provided for Request Header Type Name field.
4. Click **Field Groups**.
The Field Groups window opens.

5. Select field group **Service: Allows selection to identify the service associated with the request.**
6. Click **OK**.

The Service field group is added. It contains only one field: Service.

7. Double-click the newly added **Service** field.

The Field window opens.

Note: If you need to switch to Service List uCMDB validation, HPE strongly recommends that you modify the server configuration parameter value to keep server configuration consistency. For more information, see the *Solution Integrations Guide*.

Changing validation value here in the Field workbench window is not recommended, as the change will apply to the current request header type only.

8. Review the default settings provided. Make sure that you change default settings only when necessary. Click **OK**.

The new request header type is created.

Enable Service Field Group for Request Type

Create a new request type and enable the Service field group for it. For more information, see ["Configuring Request Types and Request Header Types" on page 108](#).

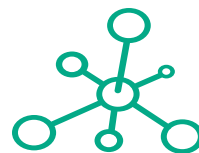
Note: To leverage the new functionality, HPE recommends that you create a new request type. Modifying existing request types might cause problems for existing requests that are based on those request types.



Chapter 7: Configuring Workflow Components

- "Overview of Workflow Step Sources " on the next page
- "Creating Workflow Step Sources " on page 211
- "Creating Decision Workflow Step Sources" on page 212
- "Creating Execution Workflow Step Sources" on page 215

- "Creating Subworkflow Workflow Step Sources " on



Hewlett Packard
Enterprise

[page 226](#)

- ["Using Workflow Parameters" on page 228](#)
- ["Modifying Workflows Already In Use" on page 230](#)

Overview of Workflow Step Sources

This section covers information about Demand Management workflows.

PPM Center includes a number of standard workflow step sources that you can add to a workflow. These sources are preconfigured with standard validations (transition values), workflow events, and workflow scope. These steps specify the following common attributes, which are expected to remain consistent across all workflows that use that step source:

- Validation associated with the step (and, thus, the list of valid transition values out of the step)
- Voting requirements of the step
- Default timeout value for the step. (You can configure a unique timeout value for each step.)
- Icon used for the step in the graphical layout.

Browse through all of the workflow step sources using the Available Workflow Steps window in the Workflow Workbench. If a step source that meets the process requirements is not available, one needs to be created.

If PPM Center has a workflow step source that meets the process requirements, you can copy and rename it. This can save configuration effort and avoid user processing errors. For example, if you need a step to route a request based on whether it needs more analysis, you could copy and use the preconfigured Request Analysis workflow step source.

Copy the step source so that it can be used uniquely for the processes. This allows you to control who can edit the step source, ensuring that the process will not be inadvertently altered by another user.

Create a new step source when the step requires any of the following:

- A unique validation (transition values) leaving the step
- A unique execution in the step: PL/SQL function, token, SQL function, or workflow step commands
- A different processing type: immediate versus manual
- A specific workflow scope
- A unique combination of these settings

Restrictions on Configuring and Using Workflow Step Source

The following restrictions apply to workflow step sources:

- You cannot delete a step source that is in use in a workflow.
- You cannot change a validation for a step source that is in use. If you must change the validation, copy the associated step source, and then configure a new validation.
- You must enable the workflow step source before you can add it to a workflow.
- Only add step sources to a workflow if the workflow has a matching workflow scope, or the step source scope is set to All.
- You cannot delete a workflow step in a workflow that has processed a request, package line, or release. Deleting the step would compromise data integrity. Instead, remove all transitions to and from the workflow step, and then disable the step.
- If Mobility Access is enabled in your PPM Center system, make sure that the name of any validations you create for custom decision workflow step sources contain no single quote (°Æ) characters. Otherwise, the Mobility Access feature cannot work correctly. For information about Mobility Access, see ["PPM Center Mobility Access" on page 94](#).

Opening the Workflow Workbench

To open the Workflow Workbench:

1. Log on to PPM Center.
2. On the **Open** menu, click **Administration > Open Workbench**.
The PPM Workbench opens.
3. On the shortcut bar, click **Configuration > Workflows**.
The Workflow Workbench opens.

For information about how to search and select, copy, or delete a workflow, see the *Getting Started* guide.

Creating Workflow Step Sources

You can create decision and execution workflow step sources from the Workflow Step Sources window. You cannot add to, delete, or modify condition steps.

Note: Subworkflow workflow steps are created by configuring a standard workflow as a subworkflow (see ["Creating Subworkflow Workflow Step Sources" on page 226](#)). You cannot add to, delete, or modify condition steps.

To create a workflow step source:

1. On the PPM Workbench shortcut bar, click **Configuration > Workflows**.
2. From the Workflow Workbench, open a workflow.
The Workflow window opens.
3. Select the Workflow Step Sources window.
4. In the first **Filter by** field, select **Requests, Packages, or Release Distributions**, depending on the type of workflow.
5. In the second **Filter by** field, select **Only items I can edit**.
6. Under Workflow Step Sources, select **Decisions** or **Executions**.
7. Click **New**.
A window that corresponds to the selected workflow step source type opens.
8. Provide the required information and any optional information to define the workflow step.
For information about how to configure a specific workflow step source, see ["Creating Decision Workflow Step Sources" on the next page](#) or ["Creating Execution Workflow Step Sources" on page 215](#).
9. Configure the ownership of the workflow step source.
For information about configuring the ownership of a workflow step source, see ["Configuring Ownership of Workflow Step Sources" on the next page](#).
10. For **Enabled**, select **Yes**.
11. Click **OK**.

The new workflow step source is now included in the Workflow Step Sources window. You can use it in any new or existing workflow with the corresponding workflow scope.

Configuring Ownership of Workflow Step Sources

As you Configure a workflow step source, you can specify who can edit the workflow step source.

To configure ownership of a new workflow step source:

1. On the PPM Workbench shortcut bar, click **Configuration > Workflows**.

2. Open a workflow.

The Workflow window opens.

3. Open a decision or execution workflow step source window.

A window that corresponds to the selected workflow step source type opens.

4. Click the **Ownership** tab.

Note: You use the **Ownership** tab to select the security groups that can edit this workflow step. The default is to allow all security groups who can edit workflows to edit a workflow step source.

5. Select **Only groups listed below that have the Edit Workflows Access Grant**.

6. Click **Add**.

The Add Security Group window opens.

7. Select a security group.

8. Click **OK**.

Only users who belong to a listed security group that can edit workflows can now edit this workflow step source.

9. From the **Ownership** tab, click **OK**.

The new workflow step source is now listed in the Workflow Step Sources window. You can use it in any new or existing workflow with the corresponding workflow scope.

Creating Decision Workflow Step Sources

This section provides instructions for creating a decision workflow step source. Before you perform these steps, HPE recommends that you first collect the information required to correctly configure the

decision workflow step source. You can use the "[Decision Workflow Step Worksheets](#)" on page 267 to gather this information.

To create a new decision workflow step source:

1. On the PPM Workbench shortcut bar, click **Configuration > Workflows**.
2. Open a workflow.
 The Workflow window opens.
3. In the first **Filter by** field, select **Requests**, **Packages**, or **Release Distributions**, depending on the type of workflow.
4. Select the **Workflow Step Sources**.
5. Under Workflow Step Sources, select **Decisions**.
6. Click **New**.
 The Decision window opens.
7. On the **Decision** tab, provide the information described in the following table.

Field Name	Description
Name	The name that describes the workflow step source. The step can be renamed when added to the workflow.
Workflow Scope	Describes the type of workflow that will be using this step source. Use the list to select a workflow scope. The following lists the possible values: <ul style="list-style-type: none"> ○ ALL. For all workflow types. ○ Requests. For Demand Management request workflows. ○ Packages. For Deployment Management package workflows. ○ Release Distributions. For Deployment Management release workflows.
Description	Description of the workflow step source.
Validation	Validations determine the transition values for the workflow step. Use the list to select a validation. Note: If Mobility Access is enabled in your PPM Center system, make sure that the name of any validations you create for custom decision workflow step sources contain no single quote (') characters. Otherwise, mobile access feature cannot work correctly on workflow step notifications. For information about Mobility Access, see " PPM Center "

Field Name	Description
	"Mobility Access" on page 94.
Decisions Required	<p>Defines the number of decisions required for the workflow step. Use the list to select a value. The following lists the possible values:</p> <ul style="list-style-type: none"> ○ One. If selected, the workflow step can progress if any one user who is eligible to act on this step makes a decision. ○ At Least One. If selected, the workflow step waits for the voters to vote on this step for a predefined amount of time, designated as the timeout. If all voters mark their decisions before the timeout period, it takes the cumulative decision as the decision for the step and proceeds forward. If any of the voting results differ before the timeout period, the step will immediately result in a No consensus outcome. A timeout period must be defined to use this choice. You can define Specific Errors in workflow steps such as Timeout and No consensus as either Success or Failure in the Define Transition window. If all voters decide on Approve, the final decision is Approve. If all voters decide on Not Approved, the final decision is Not Approved. If some voters decide on Approved and one voter decides on Not Approved, the result is No consensus. If at the end of the timeout, only a few voters (or only one voter) have cast their vote, the cumulative decision of the voters that voted will be used. If at the end of the Timeout no one has voted, the step will result in a Timeout. ○ All. If selected, the workflow step waits for all of the voters to vote. This workflow step is used along with a specified timeout period. Selecting All makes it mandatory for all voters to vote on the workflow step. The workflow step waits until the timeout period for the voters to vote. If all voters vote, the cumulative decision is considered. If some or none of the voters voted, the step remains open or closes due to a timeout, depending on the configuration. <p>When using All or At Least One, all users must unanimously approve or not approve one of the validation's selections. Otherwise, the result is No Consensus.</p>
Timeout	<p>A timeout specifies the amount of time that a step can stay eligible for completion before completing with an error (if Decisions Required is All, One, or At Least One). Timeouts can be by minute, hour, weekday or week. Timeout parameters for executions and decisions are a combination of a numerical timeout value and a timeout unit (such as weekdays).</p> <p>If this workflow step remains eligible for the value provided in the timeout value, the request, package, or release can be configured to send an appropriate notification. This field is often used in conjunction with the At</p>

Field Name	Description
	Least One and All settings for Decisions Required . Timeouts can be uniquely configured for each workflow step in the Layout tab. The timeout value specified in the workflow step source acts as the default timeout value for the step. When adding a workflow step to the workflow using this workflow step source, you can specify a different timeout value for the workflow step.
Icon	A different graphic can be specified to represent steps of this source for use on the workflow Layout tab. The graphic needs to exist in the icons subdirectory. All icons are in gif format.
Enabled	The workflow step source must be enabled in order to add the workflow step to the workflow layout.

8. Click the **Ownership** tab, and then specify the security groups that can edit this workflow step.

For detailed information about how to configure the **Ownership** tab, see ["Configuring Ownership of Workflow Step Sources" on page 212](#).

9. Click the **User Data** tab.

Product entities such as packages, workflows, requests and projects include a set of standard fields that provide information about those entities. While these fields are normally sufficient for day to day processing, user data fields provide the ability to capture additional information specific to each organization. User data is defined under the **User Data** tab. If there are no user data fields, the **User Data** tab is disabled.

10. Click the **Used By** tab. The **Used By** tab displays reference information concerning the workflow step.

11. Click **OK**.

The new workflow step source is now included in the Workflow Step Sources window. It can be used in any new or existing workflow with the corresponding workflow scope.

Creating Execution Workflow Step Sources

This section provides instructions on how to create an execution workflow step source. Before you start to perform these steps described in this section, HPE recommends that you use the Execution Step Worksheets (see ["Execution Workflow Step Worksheets" on page 266](#)) to gather the information you will need to successfully create an execution workflow step source.

To create a new execution workflow step source:

1. On the PPM Workbench shortcut bar, click **Configuration > Workflows**.
2. Open a workflow.
 The Workflow window opens.
3. Select the Workflow Step Sources window.
4. In **Filter by** field, select **Requests, Packages, or Release Distributions**, depending on the type of workflow.
5. Select the **Executions** folder.
6. Click **New**.
 The Execution window opens.
7. Provide the information described in the following table.

Field Name	Description
Name	The name of the workflow step source. The step can be renamed when added to the workflow.
Workflow Scope	Describes the type of workflow that will be using this step source. Use the list to select a workflow scope. The following lists the possible values: <ul style="list-style-type: none"> ○ ALL. For all workflow types. ○ Requests. For Demand Management request workflows. ○ Packages. For Deployment Management package workflows. ○ Release Distributions. For Deployment Management release workflows.
Reference Code	Code to refer to this execution workflow step source. PPM Center automatically populates this box based on the value you type in the Name box. You can accept the default value or enter a different code.
Description	Description of the step source.
Execution Type	Used to select the type of execution to be performed. Use the list to select an execution type. The following lists the possible values: <ul style="list-style-type: none"> ○ Built-in Workflow Event. Executes a predefined command and returns its result as the result of the step. ○ SQL Statement. Executes a SQL statement and returns its result as the result for the workflow step.

Field Name	Description
	<ul style="list-style-type: none"> ○ PL/SQL Function. Runs a PL/SQL function and returns its result as the result for the workflow step. ○ Token. Calculates the value of a token and returns its value as the result for the workflow step. ○ Workflow Step Commands. Executes a set of commands, independent of an object, at a workflow step.
Workflow Event	<p>For Execution Type Built-in Workflow Event, the specific event to perform must be selected. The available choices in the list depend on the workflow scope selected. The choices include:</p> <ul style="list-style-type: none"> ○ execute_object_commands. Executes the object type commands for a package line. ○ execute_request_commands. Executes the request type commands for a request. ○ create_package. Generates an Deployment Management package. ○ create_package_and_wait. Generates an Deployment Management package. The create workflow step that generates the package holds it until the package is closed. ○ create_request. Generates another request. ○ wf_close_success. Sets the request or package line as closed with an end status of Success. ○ wf_close_failure. Sets the request or package line as closed with an end status of Failed. ○ wf_jump. (Deployment Management and Demand Management) Instructs the workflow to proceed to a corresponding Receive Workflow Step in another workflow. ○ wf_receive. (Deployment Management and Demand Management) Instructs the workflow to receive a Jump Workflow Step and continue processing a request or package line initiated in another workflow. ○ wf_return. (Deployment Management and Demand Management) Used to route a subworkflow process back to its parent workflow.
PL/SQL Function	<p>For Execution Type PL/SQL Function, the actual function to run. The results of the function determine the outcome of the step. The results must be a subset of the validation values for that workflow step.</p>
Token	<p>For Execution Type Token, the token that will be resolved. The results of the token resolution determine the outcome of the workflow step.</p>

Field Name	Description
SQL Statement	<p>For Execution Type SQL Statement, the actual query to run. The results of the query will determine the outcome of the workflow step.</p> <p>The results of the query must be a subset of the validation values for that step.</p>
Workflow step commands	<p>For Execution Type Workflow Step Commands, the actual commands to run. The commands will result with a Succeeded or Failed value. Use a validation with those values to enable transitioning out of the step based on the execution results.</p>
Processing Type	<p>Defines when the execution is performed. Use the list to select a processing type. The following lists the possible values:</p> <ul style="list-style-type: none"> ○ Immediate. Executes the workflow step when the workflow step becomes eligible. ○ Manual. Executes the workflow step manually by a user. <p>Note: If the previous step is an execution step and the Processing Type is set to Immediate, the status dependencies, such as Clear, will not be triggered in the current step. It requires user interaction for these types of status dependencies.</p>
Validation	<p>Validations determine the transition values for the workflow step. Use the list to select a validation.</p>
Timeout	<p>Amount of time that a step is eligible before completing with an error. Timeouts can expressed in minutes, hours, days, or weeks. Timeout parameters for executions are a combination of a numerical timeout value and a timeout unit, such as days.</p> <p>If this workflow step remains eligible for the value provided in the timeout value, you can configure the request, package line, or release to send an appropriate notification.</p> <p>Timeouts can be uniquely configured for each workflow step on the Layout tab. The timeout value specified in the workflow step source acts as the default timeout value for the step. When adding a workflow step to the workflow using this workflow step source, you can specify a different timeout value for the workflow step.</p> <p>For executions, timeouts can also be uniquely configured for the amount of time that an execution is allowed to run before completing with an error. This applies to the workflow step commands and object type commands only. Command-level timeouts are set in the Command window of an object type.</p>
Icon	<p>You can select a different graphic to represent this steps of this workflow</p>

Field Name	Description
	step source. This graphic needs to exist in the icons subdirectory. All icons are in gif format.
Enabled	The workflow step source must be enabled in order to add it to the workflow layout.

8. Click the **Ownership** tab.

Use the **Ownership** tab to specify the security groups that can edit this workflow step. The default is to allow all security groups who can edit workflows to edit a workflow step source. For complete instructions on how to configure workflow step security, see "[Configuring Ownership of Workflow Step Sources](#)" on page 212.

9. Click the **User Data** tab.

Product entities such as packages, workflows, requests and projects include a set of standard fields that provide information about those entities. While these fields are sufficient for day-to-day processing, user data fields provide the ability to capture additional information specific to your organization. (User data is defined from the **User Data** tab. If there are no user data fields, the **User Data** tab is disabled.)

10. Click the **Used By** tab.

The **Used By** tab displays reference information about the workflow step.

11. Click **OK**.

The new workflow step source is now included in the Workflow Step Sources window. It can be used in any new or existing workflow with the corresponding workflow scope.

Setting Up Execution Steps

When setting up execution workflow steps, be sure to include workflow events (transitions) for both success and failure. If a workflow step has failed and users cannot select Failure as one of the workflow events, the workflow cannot continue to drive the request.

Defining Executions Types

Execution workflow steps are used to perform specific actions. Demand Management provides a number of built-in workflow events for processing common execution events, such as running request type commands, object type commands, and closing a request. You can create custom executions based on SQL, PL/SQL, token resolution, and custom commands.

Executing Request Type Commands

Certain process steps require that specific commands be executed. Commands can be added to each request type and the workflow can be configured to execute request type commands at a specific step in the process. Each step runs its own commands to ensure the correct execution for that request type.

The execution workflow step source Execute Request Commands performs this task. Use this step source unless it does not meet the required specifications, such as validation or processing type.

To create the execution step source, make a copy of execution workflow step source Execute Request Commands and change the field values as shown in "[Table 7-1. Execution window values to execute request type commands](#)" below.

Table 7-1. Execution window values to execute request type commands

Field Name	Description
Name	Type a descriptive name for the step source.
Workflow Scope	Requests
Execution Type	Built-in Workflow Event
Workflow Event	execute_request_commands
Processing Type	Manual or Immediate
Validation	WF - Standard Execution Results This is the default selection. You can select another existing or create a new validation.
Enabled	Yes
Processing Type	Manual
Page Response	<ul style="list-style-type: none">• Finish execution before displaying the request type page to user: if

Table 7-1. Execution window values to execute request type commands, continued

Field Name	Description
	<p>you select this option, the step will complete the execution before reloading the request page for the user (enabling them to make further changes).</p> <p>Note: This option does not work for request creation. It only works when end users click a workflow action button.</p> <ul style="list-style-type: none"> • Display the request page immediately while execution is still running: if you select this option, the request page will reload immediately while the execution is still in progress. <p>Note: The request status on the request page may be incorrect if you select this option.</p>

Closing Requests as Success

You can create an execution step that closes a request and marks the request as successful. Each request workflow should must with a closed request. All the requests that were closed successfully can then be included in reports.

The execution workflow step sources Close (Immediate success) and Close (Manual success) perform this task. Use one of these step sources unless they do not meet the required specifications, such as validation or processing type.

To create the execution step source, make a copy of execution workflow step source Close (Immediate success) or Close (Manual success) and change the field values as defined in "[Table 7-2. Execution window values to close requests as success](#)" below.

Table 7-2. Execution window values to close requests as success

Field Name	Description
Name	Type a descriptive name for the step source.
Workflow Scope	Requests
Reference Code	Accept the default value, or type a different reference code
Execution Type	Built-in Workflow Event
Workflow Event	wf_close_success
Processing Type	Manual or Immediate

Table 7-2. Execution window values to close requests as success, continued

Field Name	Description
Validation	WF - Standard Execution Results This is the default selection. You can select another validation or create a new one.
Enabled	Yes

Closing Requests as Failed

You can create an execution step that closes a request and marks the request as Failed. Each request workflow must resolve with a closed request. The execution workflow step source Close (Immediate failure) performs this task. Use this step source unless it does not meet the required specifications, such as validation or processing type.

To create the execution step source, make a copy of execution workflow step source Close (Immediate failure) and changes the field values as defined in "[Table 7-3. Execution window values to close requests as failed](#)" below.

Table 7-3. Execution window values to close requests as failed

Field Name	Description
Name	Type a descriptive name for the step source.
Workflow Scope	Requests
Reference Code	Accept the default value, or type a different reference code
Execution Type	Built-in Workflow Event
Workflow Event	wf_close_failure
Processing Type	Manual or Immediate
Validation	WF - Standard Execution Results (This is the default selection. You can select another existing or create a new validation.)
Enabled	Yes

Executing PL/SQL Functions and Creating Transitions Based on the Results

PL/SQL function execution workflow steps are used when a workflow must be routed based on the results of the PL/SQL function. A PL/SQL function execution workflow step runs a PL/SQL function and returns its results as the result of that workflow step.

Create a new execution step source with the field values as defined in "[Table 7-4. Execution window values for executing PL/SQL functions](#)" below.

Table 7-4. Execution window values for executing PL/SQL functions

Field Name	Description
Name	Type a descriptive name for the step source.
Workflow Scope	Requests
Reference Code	Accept the default value, or type a different reference code
Execution Type	PL/SQL Function
Processing Type	Manual or Immediate
Validation	Selects or creates a validation that includes all of the possible values of the SQL query. You can also create a validation validated by SQL. Use the same SQL from the execution minus the WHERE clause.
Execution	Type the PL/SQL function.
Enabled	Yes

Executing SQL Statements and Creating Transitions Based on the Results

SQL statement execution workflow steps are used when a workflow must be routed based on the result of a query. An SQL statement execution workflow step runs a SQL query and returns its results as the result of that workflow step.

The following rules apply to creating an SQL statement:

- Use only SELECT statements.
- You can use tokens within the WHERE clause.
- A query must return only one value.

Create a new execution step source with the field values as defined in "[Table 7-5. Execution window values for executing SQL statements](#)" below.

Table 7-5. Execution window values for executing SQL statements

Field Name	Description
Name	Type a descriptive name for the step source.
Workflow Scope	Requests
Reference Code	Accept the default value, or type a different reference code
Execution Type	SQL Statement
Processing Type	Manual or Immediate
Validation	Selects or creates a validation that includes all of the possible values of the SQL query. Tip: You can create a validation validated by SQL. Use the same SQL defined for the execution minus the WHERE clause.
Execution	Type the SQL query.
Enabled	Yes

Evaluating Tokens and Creating Transitions Based on the Results

Demand Management includes workflow execution steps that you can use to set up data-dependent rules for routing workflow processes. Token execution workflow steps enable a workflow to be routed based on the value of any field within a particular entity.

A token execution workflow step references the value of a given token and uses that value as the result of the workflow step. A transition can be made based on the value stored in the product by using tokens in the execution workflow step.

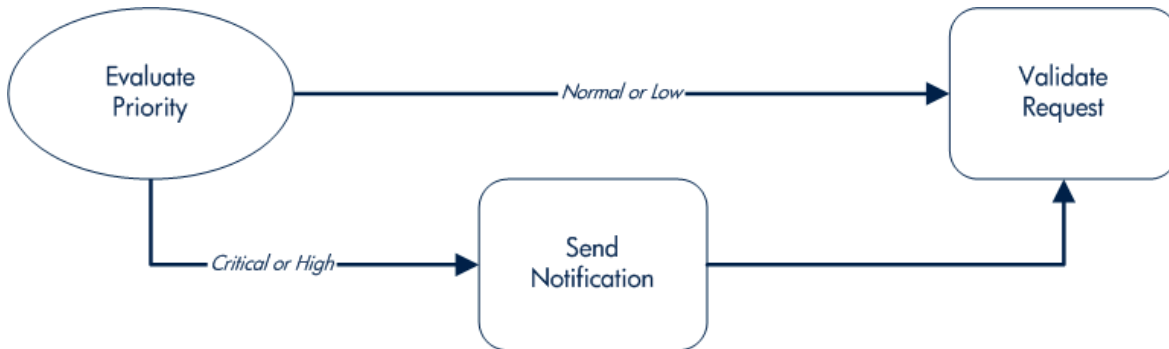
Create a new execution step source with the field values as defined in "[Table 7-6. Execution window values for evaluating tokens](#)" on the next page.

Table 7-6. Execution window values for evaluating tokens

Field Name	Description
Name	Type a descriptive name for the workflow step source.
Workflow Scope	Requests
Reference Code	Accept the default value, or type a different reference code
Execution Type	Token
Processing Type	Manual or Immediate
Validation	Selects or creates a validation that includes all of the possible values of the resolved token. For example, if the token is for the Priority field, use the validation for the Priority field here as well.
Execution	Type the token for the value on which the transition is to be based.
Enabled	Yes

For example, IT needs to send an email notification to the Validate and Approve Requests group if the request priority is High or Critical.

Figure 7-1. Transitioning based on a token



IT decides to use an execution workflow step to automatically evaluate the request priority and route it accordingly. If the request priority is High or Critical, it gets sent to an immediate execution workflow step that then sends a notification to the Validate and Approve Requests group before continuing along the workflow. To accomplish this, an execution workflow step source, Evaluate Priority, is configured with the parameters listed in ["Table 7-7. Example of execution window values for evaluating tokens" below](#).

Table 7-7. Example of execution window values for evaluating tokens

Field Name	Description
Name	Evaluate Priority

Table 7-7. Example of execution window values for evaluating tokens, continued

Field Name	Description
Workflow Scope	Requests
Reference Code	Accept the default value, or type a different reference code
Execution Type	Token
Processing Type	Immediate
Validation	CRT - Priority - Enabled
Execution	[REQ.PRIORITY_CODE]
Enabled	Yes

Executing Multiple System-Level Commands

System-level commands can be run for execution steps of the following execution types:

- Built-in workflow event (execute_request_commands)
- Workflow step commands

When either the workflow or the request type commands execute at this step, the commands either succeed or fail. It may be preferable to retain the option of resetting failed execution steps, rather than immediately transitioning along a failed path. This is often helpful when troubleshooting the execution.

Creating Subworkflow Workflow Step Sources

This section provides instructions on how to create a subworkflow workflow step source. Before you start to perform these steps described in this section, HPE recommends that you use the Subworkflow Step Worksheet (see "[Subworkflow Workflow Step Worksheets](#)" on page 269) to gather the information you will need to successfully create a subworkflow workflow step source.

A subworkflow is a workflow that is referenced from within another workflow. Subworkflows enable you to model complex business processes into logical, more manageable, and reusable subprocesses.

You can drag a subworkflow from the Workflow Step Sources window and drop it onto the Layout tab. When the package, request, or release reaches the subworkflow step, it follows the path defined in that subworkflow. The subworkflow either closes within that workflow or returns to the parent workflow.

You define subworkflows from the PPM Workbench using the same process as you use to configure a workflow. To create a subworkflow, you must:

- Set the **Sub-workflow** option to **Yes**.
- Ensure that the validation for the step leaving the subworkflow layout matches the subworkflow step in the parent workflow.

Subworkflows Returning to Demand Management Workflows

You can set up an execution workflow step so that it automatically returns from a subworkflow to its parent Demand Management workflow.

For a request to transition back to the parent workflow, the subworkflow must contain a return step. Transitions leading into the return step must match the validation established for the subworkflow step.

You must verify that the validation defined for the subworkflow step is synchronized with the transitions entering the return step. Demand Management includes the execution workflow step source Return from Subworkflow that performs this task. Use this step source unless it does not meet the required specifications, such as validation or processing type.

To create the execution step source, make a copy of execution workflow step source Return from Subworkflow and change the field values as defined in "[Table 7-8. Execution window values for subworkflows](#)" below.

Table 7-8. Execution window values for subworkflows

Field Name	Description
Name	Type a descriptive name for the workflow step source.
Workflow Scope	Requests
Reference Code	Accept the default value, or type a different reference code
Execution Type	Built-in Workflow Event
Workflow Event	wf_return
Processing Type	Manual or Immediate
Validation	WF-Standard Execution Results (This is the default selection. You can select another existing or create a new validation.)
Enabled	Yes

Using Workflow Parameters

Use workflow parameters to store the results of a workflow step. This value can then be used later to define a transition. The following lists the rules concerning workflow parameters:

- You can use the `WF.I.P` token prefix to reference workflow parameters.
- You can use workflow parameters in PL/SQL and SQL workflow step executions.

Creating Workflow Parameters

To create a workflow parameter:

1. Log on to PPM Center.
2. On the **Open** menu, click **Administration > Open Workbench**.
The PPM Workbench opens.
3. On the shortcut bar, click **Configuration > Workflows**.
4. Open a workflow.
5. In the **Parameters** section of the **Workflow** tab, click **Add**.
The Workflow Parameter window opens.
6. Provide the following information:
 - a. In the **Prompt** field, type the name of the workflow parameter.
 - b. In the **Token** field, type the token name (for example, `LOOP_COUNTER`).
 - c. In the **Description** field, you can type a short parameter description.
 - d. In the **Default Value** field, you can specify the initial parameter value.
7. In the **Parameters** section of the **Workflow** tab, click **Add**.
8. Click **OK**.
9. From the **Workflow** tab, click **OK**.

Example: Using Workflow Parameters to Build a Loop Counter

You can use a workflow parameter that generates a counter to keep track of the number of times a workflow step enters a state.

To build a loop counter:

1. On the PPM Workbench shortcut bar, click **Configuration > Workflows**.
2. Open a workflow.
3. In the **Parameters** section **Workflow** tab, click **Add**.
4. In the Workflow Parameter dialog box, complete the following fields:
 - a. In the **Prompt** field, type `Loop Counter`.
 - b. In the **Token** field, type the token name (for example, `LOOP_COUNTER`).
 - c. In the **Description** field, you can type a short parameter description.
 - d. In the **Default Value** field, you can specify a default parameter value.
5. In the Workflow Parameter dialog box, click **Add**.
6. Click **OK**.
7. From the **Workflow** tab, click **OK**.
8. Create a new immediate SQL execution workflow step.

For details on how to create an SQL execution workflow step, see "[Creating Execution Workflow Step Sources](#)" on page 215.

There are two key concepts to note about the new step definition.

- The result of the SQL execution workflow step returns the result `LOOP_COUNTER + 1`. This return value is linked back into the parameter when the workflow step is generated on a workflow.
- A validation for a numeric text field is used. This allows you to use `<=`, `<`, `>=`, and `>` comparisons in transitions off this step.

The following shows the Execution window for the SQL execution workflow step.

9. Add the workflow step to a workflow and choose the new workflow parameter `Loop Counter`.

By choosing Loop Count, the workflow engine is told to assign the result of "select loop counter val + 1" from dual back into the loop counter parameter.

You can now add transitions to and from the new loop counter step. For example, you add the loop counter each time an execution fails. If the execution fails three times, a notification is sent to the user. If the execution fails five times, management is notified.

Modifying Workflows Already In Use

Workflows can be modified while they are going through their workflow steps after a package or request has been initiated. These modifications include adding new workflow steps, as well as changing the transitions, security assignments and notifications from within the workflow.

You can make changes to workflows that are in use by using the same procedures that you used to define the workflows from the Workflow Workbench.

Keep in mind that, when you modify workflows that are in use, specific limitations apply to which entities you can add, change, delete, or rename. These limitations are described in "[Table 7-9. Rules for modifying production workflows](#)" below.

Table 7-9. Rules for modifying production workflows

Entity	Procedure
Transitions Security Notifications Workflow steps Workflow parameters	You can change any of these entities or add them to a workflow that is in use.
Transitions Security Notifications Workflow parameters	You can delete any of these entities from a workflow in use.
Workflow steps	You cannot delete this entity from a workflow in use, but you can rename it. You can delete transitions coming into or going out of a workflow step to effectively remove it from the workflow.

If a workflow that is in use is changed and saved, the changes take effect immediately. Any changes made to workflow steps are applied to all open package lines, requests, releases, and distributions.

Changes to a workflow can have undesirable effects on requests or packages that are in progress and are using that workflow.

Modifying a workflow that is in use can disrupt the normal flow in and out of the workflow and prevent it from reaching completion. For example, removing a transition from a workflow step may result in the requests or package lines getting stuck in that workflow step.

Performance Considerations

Updating workflow step security with a specific configuration can affect system performance. When adding dynamic security to a step, such as based on a standard or user-defined token, in the Workflow Step window on the **Layout** tab, product database tables are updated to handle the new configuration.

Migrating a workflow with these types of changes into an instance of the PPM Center can also affect system performance. Product database tables must be updated to handle the new workflow.

Because of the scope of these database changes, HPE recommends that you rerun statistics on your PPM Center database schema. For information about how to collect database statistics on your database, see the *Installation and Administration Guide*. For help with this procedure, contact your database administrator.

Copying and Testing Trial Versions of Workflows

Before you modify a workflow that is in use, do the following:

1. Make a copy of the original workflow.
2. Modify the copied version of the workflow with the changed workflow steps.
3. Test the modified version of the workflow to ensure that it works correctly.
4. Determine if the workflow step is in use. To determine which steps are eligible, remove the incoming transition to the step that will be deleted and run the following reports:
 - To determine when the requests have flowed out of a workflow step, run the Workflow Detail Report. This report indicates if the step to delete is eligible for user action or has been completed.
 - To determine if any package lines are eligible for user action in a workflow, run the Packages Pending Report.

You can now to make the same changes to the original workflow.

Modifying Production Workflows

The final step in modifying workflows already in use is to modify the production workflow. The following sections offer guidance on how to modify the production workflow.

Disabling Workflow Steps

As mentioned in "[Table 7-9. Rules for modifying production workflows](#)" on page 230, you cannot delete a step from a workflow that is in use. You can only disable it. However, you may want to change the process. Any changes to the process must be reflected in the workflow. This may require disabling existing steps and adding new steps.

To disable a step and add a new one:

1. Remove transitions to the workflow step you no longer want to use.
2. Add a new step to the workflow.
3. Redirect the transitions to the new workflow step.

Redirecting Workflows

If you disable a workflow step that is eligible for user action, the requests or package lines in that step become stuck. Because the step is disabled, the user cannot take action on it and so cannot proceed to the next step in the workflow.

The outgoing transition to be deleted is still intact, so the eligible package lines and requests will eventually be acted upon and flow out of the workflow step.

Add a new workflow step to the workflow and redirect the transitions to that new workflow step so that the movement of package lines and requests avoids the disabled step and is not interrupted.

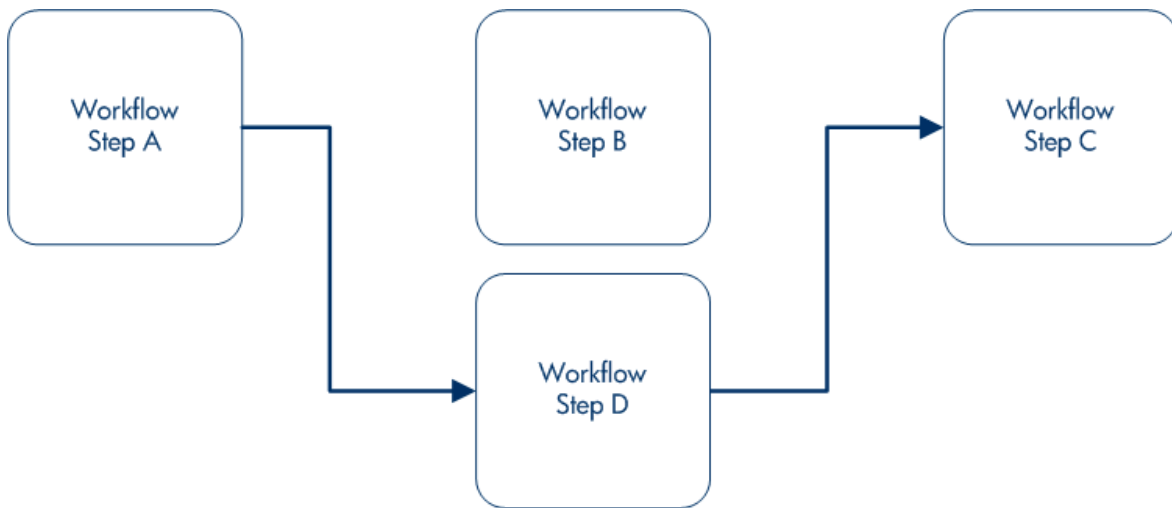
For example, consider the workflow sequence shown in "[Figure 7-2. Redirecting the workflow, step 1](#)" below, in which you want to disable step B.

Figure 7-2. Redirecting the workflow, step 1



After you remove the incoming and outgoing transitions to B, add a new workflow step D, which connects steps A and C and allow the workflow to continue to process requests or package lines (see "Figure 7-3. Redirecting the workflow, step 2" below).

Figure 7-3. Redirecting the workflow, step 2



Run the report(s) again to ensure there are no entities eligible for action by the user in the disabled step.

Moving Requests or Packages Out of Steps

If the requests or packages are stuck in a step after a transition is removed from a workflow in use, add the deleted transition back to the workflow. After the requests or packages have flowed out of the step, delete the transition again.



Chapter 8: Configuring Contacts

- "Overview of Contacts " below
- "Opening the Contact Workbench" on the next page
- "Creating Contacts" on the next page

Overview of Contacts



Hewlett Packard
Enterprise

Contacts are resources used as a point of reference or information. Contacts must have a valid PPM Center username and the company they work for must be included in the validation, CRT - Company Validation. Contact information can be added for users in PPM Center as well as external users.

Contacts are created in the Contact window. The Contact window consists of a general information section and a large section reserved for potential user data fields.

Opening the Contact Workbench

To open the Contact Workbench:

1. Log on to PPM Center.
2. On the **Open** menu, click **Administration > Open Workbench**.

The PPM Workbench opens.

3. On the shortcut bar, click **Demand Mgmt > Contacts**.

The Contact Workbench opens.

Creating Contacts

To create a new contact:

1. On the PPM Workbench shortcut bar, click **Demand Mgmt > Contacts**.

The Contact Workbench opens.

2. Click **New Contact**.

The Contact window opens.

3. Provide the information described in the following table.

Field Name	Description
First Name	First name of the contact.
Last Name	Last name of the contact.
User	PPM Center username for the contact. This field is populated from the KNTA - User Id - All Validation auto-complete and cannot be edited.

Field Name	Description
Phone Number	Phone number of the contact.
Email Address	Email address of the contact.
Company	Company that employs the contact. This field is populated from CRT - Company Validation auto-complete and cannot be edited.
Enabled	Select Yes to make the notification available to the system.

4. In the Contact window, click **OK**.

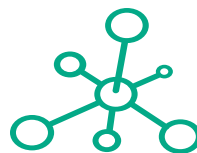
The changes to the notification template are saved.



Chapter 9: Configuring Notification Templates

- "Overview of Notification Templates " on the next page
- "Opening the Notification Template Workbench " on the next page
- "Creating Notification Templates" on page 239
- "Configuring Notification Intervals" on page 242

- "Checking the Usage of Notification Templates " on



Hewlett Packard
Enterprise

Overview of Notification Templates

Notification templates are preconfigured notifications that you can use to quickly construct the body of a message. You can use notification templates with the following PPM Center entities:

- Tasks
- Projects
- Requests
- Packages
- Releases
- Workflows
- Reports

Opening the Notification Template Workbench

To create a notification template, you use the Notification Template Workbench.

To open the Notification Template Workbench:

1. Log on to PPM Center.
2. On the **Open** menu, click **Administration > Open Workbench**.
The PPM Workbench opens.
3. On the shortcut bar, click **Configuration > Notification Templates**.
The Notification Template Workbench opens.

For information about how to search and select and copy an existing notification template, see the *Getting Started* guide.

Deleting Notification Templates

You cannot delete notification templates that are referenced from an existing notification. To delete such a notification template, you must first remove these references. Referenced notification templates can be disabled. For information about how to determine whether a notification template is referenced, see "[Checking the Usage of Notification Templates](#) " on page 243. For information about how to delete a notification template type, see the *Getting Started* guide.

Creating Notification Templates

To create a notification template:

1. On the PPM Workbench shortcut bar, click **Configuration > Notification Templates**.
The Notification Template Workbench opens.
2. Click **New Notification Template**.
The Notification Template window opens.
3. In the **Template Name** field, type a name for the template.
4. To indicate the range of use for this new notification, from the **Notification Scope** list, select a PPM Center product area.
Note: The default notification scope is **Packages**. Selecting a different scope changes the notification template format.
5. To specify a message format, from the **Notification Format** list, select **Plain Text** or **HTML**.
6. To make this template available in PPM Center, for the **Enabled** option, leave **Yes** selected.
7. To make this template the default notification template for PPM Center, for the **Default** option, select **Yes**.
8. To specify "From" address:
 - a. In the **From** row, click **Choose**.
The Email Header Field window opens.
 - b. From the list at the top of the window, select the sender category.

The context-sensitive required field is dynamically updated to gather the necessary information for that category. For instance, if you select **Enter an Email Address** from the list, then it is necessary to specify an email address. If you select **User Defined Token**, click **Tokens** to bring up a list of available tokens or type in a specific token.

- c. Complete the required field.
 - d. If you select **User Defined Token**, select the token type that corresponds to the evaluated token value.
 - e. In the Email Header Field window, click **OK**.
9. In the Notification Template window, provide a reply-to address, as follows:
- a. Next to **From**, click **Choose**.
The Email Header Field window opens.
 - b. Select the recipient category.
The context-sensitive required field is dynamically updated to gather the necessary information for that category. For instance, if **Enter an Email Address** is selected, then it is necessary to provide an email address. If **User Defined Token** is selected, click **Tokens** to bring up a complete list of available tokens or type in a specific token.
 - c. Provide the information in the required fields.
 - d. If **User Defined Token** is specified, select the token type that corresponds with the evaluated token value.
 - e. In the Email Header Field window, click **OK**.
10. In the **Body** field, type the notification content.
Ensure that the message body format is the same as that specified in **Notification Format**.
11. In the **Body** field, add tokens to the body of the text, as follows:
- a. Click **Tokens**.
The Token Builder window opens.
 - b. Select a token.
 - c. In the **Token** field, copy the token name and then paste it in the **Body** field.
 - d. Click **Close**.
12. Configure the ownership of the notification template.

For detailed information about how to configure the ownership of the notification template, see "[Configuring Ownership of Notification Templates](#)" below.

13. Click **OK**.

Configuring Ownership of Notification Templates

Ownership groups are defined by adding security groups to the Ownership window. If no ownership groups are associated with a particular entity, the entity is considered global and any user who has the edit access grant for the entity can edit, copy, or delete it. For detailed information about access grants, see the *Security Model Guide and Reference*.

If a security group is disabled or loses the edit access grant, members of that group can no longer edit the entity.

To configure the ownership of a custom notification template:

Note: You can only configure ownership for custom notification templates, and not for the preconfigured templates.

1. On the PPM Workbench shortcut bar, click **Configuration > Notification Templates**.
The Notification Template Workbench opens.
2. Open a custom notification template.
The Notification Template window opens.
3. At the bottom of the window, click **Ownership**.
The Ownership window opens.
4. Select one of the following ownership options:
 - **All users with the Edit Notification Template Access Grant**
 - **Only groups listed below that have the Edit Notification Template Access Grant**
5. If you selected **Only groups listed below that have the Edit Notification Template Access Grant**:
 - a. Click **Add**.
The Add Security Groups window opens.
 - b. Use the Security Groups auto-complete to select one or more security groups.

- c. Click **OK**.

The **Ownership** tab lists the selected security groups.

6. Click **OK**.

The changes to the notification template are saved.

Deleting Ownerships from Notification Templates

To delete an ownership:

1. On the PPM Workbench shortcut bar, click **Configuration > Notification Templates**.

The Notification Template Workbench opens.

2. Open a notification template.
3. Click **Ownership**.
4. In the Ownership window, select an ownership to remove.
5. Click **Remove**.
6. Click **OK**.

Configuring Notification Intervals

To create a new notification template:

1. On the PPM Workbench shortcut bar, click **Configuration > Notification Templates**.

The Notification Template Workbench opens.

2. On the PPM Workbench menu, click **Notification Templates > Intervals**.

The Notification Intervals window opens.

3. Click **New**.

The Notification Interval: New window opens to the **Interval** tab.

4. Provide the information described in the following table.

Field Name (*Required)	Description
*Interval Name	Name assigned to the interval.
Description	(Optional) Description of the interval.
*Interval Type	For internal use. This is always set to Periodic , unless Immediate Interval is used.
*Start Time	Time to start sending out notifications and to start counting down the time interval until the next batch.
*End Time	Time to stop sending out notifications.
*Time Interval (Hours)	Number of hours to wait after the start time or the last batch sent, before sending out the next batch of notifications.
*Days	Used to select which days on which this interval is to execute.
*Enabled	If set to Yes , this interval is selectable. If set to No , this interval is unavailable.

5. Click **OK**.
6. Click **Close**.

The new notification interval can now be used in any workflow step notification.

If notifications are sent at an hourly or daily interval, there are sometimes several notifications pending for a particular user. In this case, all notifications are grouped together in one email message. The subject of each notification is displayed in a **Summary** section at the top of the message.

Checking the Usage of Notification Templates

To check the usage of a notification template:

1. On the PPM Workbench shortcut bar, click **Configuration > Notification Templates**.
The Notification Template Workbench opens.
2. Open the notification template.
3. Click **Used By**.

The Used By window opens and lists all references to the notification template.

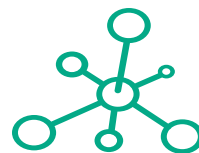
4. Click **OK**.
5. In the Notification Template window, click **OK**.



Chapter 10: Configuring User Data

- "Overview of User Data " on the next page
- "Opening the User Data Workbench " on page 248
- "Viewing General Information for User Data Types" on page 249
- "Creating a User Data Context" on page 250

- "Configuring User Data Fields" on page 251



**Hewlett Packard
Enterprise**

- ["Configuring User Data Layouts " on page 258](#)
- ["Configuring User Data for Resources" on page 261](#)

Overview of User Data

Product entities such as packages, workflows, requests, and projects include a set of standard fields that provide information about the entities. While these fields are normally sufficient for day-to-day processing, you can create *user data fields* to capture additional information specific to your organization. For example, if you want to include an additional field on every package, you can open the **Validation Value User Data** user data type (with global scope) and define the extra field, which is then displayed on the **User Data** tab for a validation.

You configure user data types from the User Data Context window in the User Data Workbench. In ["Figure 10-1. User data types" below](#), the User Data Workbench **Results** tab lists some of the preconfigured user data types available.

Figure 10-1. User data types

Query	User Data Type	Scope	Context Field	Context Value	Enabled
Results	Security Group User Data	Global			Y
	Service Item User Data	Global			Y
	Staff Prof Position User Data	Global			Y
	Staffing Profile User Data	Global			Y
	Task User Data	Global			Y
	Time Sheet Line User Data	Global			Y
	User User Data	Global			Y
	Validation Value User Data	Global	Validation Name		Y
	Validation Value User Data	Context	Validation Name	CONNECTION_PR...	Y
	Validation Value User Data	Context	Validation Name	DATA_MASK	Y
	Validation Value User Data	Context	Validation Name	Default Type	Y
	Validation Value User Data	Context	Validation Name	IT Request Copying ...	Y
	Validation Value User Data	Context	Validation Name	CRT - Platform	Y
	Validation Value User Data	Context	Validation Name	TRANSFER PROT	Y

40 Record(s) loaded.

The following four columns in the User Data Workbench define the components that fully define a user data type:

- **User Data Type.** This column displays the user data type names, which are predefined and uneditable in PPM Center.

Although you cannot create new user data types, you can create new user data *contexts* (based on the Validation Value User Data, the Package User Data, or the Environment User Data types) and define user data fields for them.

- **Scope.** This column displays the scope of the user data type field. The two possible scope values are:
 - **Global.** If the user data type field has a global scope, the **User Data** tab for every designated entity contains the defined user data field.
 - **Context.** If the user data type field has a context scope, then the defined user data field is added only to the **User Data** tab for entities that have specific **Context Field** and **Context Value** definitions.
- **Context Field.** This column displays the context-sensitive fields. It applies only to user data type fields with context scope. Because each user data type only has one available context field value, the cells in this column are populated automatically.
- **Context Value.** This column lists the value (context) for the context-sensitive field. It applies only to user data type fields with a context scope. You cannot create a new context value. You can only assign an existing one.

You can define up to 20 user data type fields for display on the **User Data** tab of a defined entity. You can configure the major attributes of each field, including its graphical presentation, the validation method, and whether it is required.

Referencing User Data

Once you have a user data field, you can refer to it from other parts of the product (in notifications and command executions) by using its token name, preceded by the entity abbreviation and the user data (UD) qualifier. For example, Validation Value User Data might have the field "Class Name" with the token value `CLASS_NAME`, and the user data qualifier `USER_DATA1`.

Migrating User Data

For any configuration entity that has user data type fields, the data in the user data type fields is migrated with the entity.

- If two instances have identical user data configurations, then the user data is migrated correctly.
- If two instances do not have identical user data configurations, then the user data is mapped to the data model according to the storage configuration in the source instance. Verify that the two instances have the same user data fields. Otherwise, you must correct the user data after migration.
- If the user data is context-sensitive, then a corresponding context-sensitive configuration must exist in the destination instance, or the migration fails.
- User data fields that have hidden and visible values can cause problems. If the hidden value of a user data field refers to a primary key (such as Security Group ID) that is different in the source and destination instances, the migrator does not correct the hidden value. In this case, you must correct the user data manually, after migration.

User Data Configuration Tasks

The following sections provide instructions for configuring user data, which involves the following tasks:

- Open the User Data Workbench ("[Opening the User Data Workbench](#) " below)
- Open a user data type and view general information ("[Viewing General Information for User Data Types](#)" on the next page)
- Create user data fields ("[Configuring User Data Fields](#)" on page 251)
- Configure user data field layout ("[Configuring User Data Layouts](#) " on page 258)

Opening the User Data Workbench

To open the User Data Workbench:

1. Log on to PPM Center.
2. On the menu bar, click **Administration > Open Workbench**.
The PPM Workbench opens.
3. On the shortcut bar, click **Configuration > User Data**.

For information about how to search for and select existing user data, copy user data, and delete user data, see the *Getting Started* guide.

Viewing General Information for User Data Types

To view general information for a user data type in the User Data Context window:

- From the User Data Workbench, open a user data type, or create a new user data context.

"Table 10-1. Fields in the User Data Context window" below lists descriptions of the fields in the User Data Context window.

Table 10-1. Fields in the User Data Context window

Field Name (*Required)	Description
*User Data Type	User data type name. For global user data types, this field is automatically populated. If you are creating a context-sensitive user data context, you select the type from the list.
*Context Field	For user data types and user data contexts that have context scope, this field is automatically populated with the name of the context-sensitive field. The Context Field auto-complete is only enabled for the Environment User Data and Package User Data user data types.
*Context Value	For context-sensitive user data types, this field displays the value for the context field. This field is disabled for user data types with global scope. You can only define one context value for the context field. For example, you cannot have two context-sensitive user data types with the same context field and context value (such as a field labeled Priority with a value of "Critical").
Enabled	Use this option to enable (default) or disable the user data type in PPM Center.
Scope	Scope of the user data type. This field is automatically populated based on the user data type. The possible scopes for a user data type are: <ul style="list-style-type: none"> Global. Standard user data type scope. If the scope is global, the User Data tab for every designated entity displays the defined field(s). Context. Indicates that this is a context-sensitive user data type. If the user data type has the context scope, the User Data tab displays the

Table 10-1. Fields in the User Data Context window, continued

Field Name (*Required)	Description
	defined field(s) only if the designated entities have the correct Context Field and Context Value definitions.
*Meta Layer View	Meta layer views relate information specific to PPM Center. For example, the reporting meta layer view MREQ_OPENED_CLOSED_BY_TYPE_D provides summary information for request submission and completion activity, broken down by request type and by calendar day.

Creating a User Data Context

Although you cannot create a new user data type in PPM Center, you can create a user data context that is based on one of the following user data types:

- **Validation Value User Data.** Create user data fields for a named drop-down validation. Typically, you create this new user data context in order to associate more data with values available for users to select.

Example: Your PPM Center system has a **US States** drop-down list validation that has 50 validation values. Somewhere else in the system, you need to get the capital of the state that a user has selected. So, you create a new user data context for the **US States** list and add the **Capital** field to it.

You next open the **US States** drop-down list validation, and for each validation value (state), you complete the **Capital** field. Now, the system can detect which state a user has selected and pick up the capital.

- Environment User Data
- Package User Data

To create a new user data context for a drop-down list validation:

1. Open the User Data Workbench.
2. On the **Query** tab, click **New User Data Context**.

The User Data Context Window opens.

3. Click **User Data Type**.

The **User Data Type** field displays the value **Validation Value User Data**, the **Context Field** field displays the value **Validation Name**, and the **Scope** field displays **Context**.

4. Use the **Context Value** auto-complete to select a validation value for the **Validation Name** context field.
5. Create one or more user data fields.

For information about how to create a user data field, see ["Configuring User Data Fields" below](#).

6. Click **OK**.

Configuring User Data Fields

This section provides instructions on how to configure a user data field to capture information specific to your organization.

Note: Not all user data field types have **Dependency** and **Security** tabs.

To create a user data field:

1. Open the User Data Workbench (for instructions, see ["Opening the User Data Workbench" on page 248](#)).
2. Open a user data type, or create a new user data context (for instructions, see ["Creating a User Data Context" on the previous page](#)).

The User Data Context window opens to the **Fields** tab.

3. Click **New**.

The Field: New window opens.

4. Provide information for the fields described in the following table.

Field Name	Description
Field Prompt	Label displayed for the user data field in the request.
Token	Uppercase text string used to identify the token. The token name must be unique to the specific user data. An example of a token name is ASSIGNED_TO_USER_ID.
Description	Type a description of the user data field in this field.

Field Name	Description
Enabled	To disable the field in PPM Center, select No . (The user data field is enabled by default.)
Validation	Use the Validation auto-complete to specify the logic to use to determine the valid values for this field. This could be a list of user-defined values, a rule that the result must be a number, and so on. After you select the validation logic, the Component Type field displays the type of component (for example, drop-down list, text field, auto-complete list) used in the validation.
Multiselect	If the validation uses an auto-complete list component type, and you want users to be able to specify multiple values, select Yes .

5. If the **Attributes** tab is displayed, provide the information listed in the following table.

Field Name	Description
User Data Col	Indicates the internal column in which the field value is to be stored. These values are then be stored in the corresponding column in the table for the given entity (such as KNTA_USERS for the users entity). User data provides the ability to store information in up to 20 columns, thus allowing for up to 20 fields. No two fields in user data can use the same column.
Display Only	Indicates whether the field is read-only. Select Use Dependency Rules to use the logic defined on the Dependencies tab.
Display	Indicates whether the user can view this field on the User Data tab.
Required	Indicates whether the user must specify a value for this field. Select Use Dependency Rules to use the logic defined on the Dependencies tab.

6. If the **Defaults** tab is displayed, click it, and then provide the information listed in the following table.

Field Name	Description
Default Type	Specifies whether the field is to have default value, and if it is, whether the default value is a constant value or a parameter value. Note: If the new user data field has a default value, the field is added to all existing requests, but has a NULL value for those requests. This affects request searches that use this field because users cannot specify NULL as a field value in the search criteria.

Field Name	Description
Visible Value	If you select the Constant default type, specify the constant value here.
Depends On	To default from another field, choose the token name for that field. When using this user data, every time a value is provided or updated in the source field, it will automatically be provided or updated in this destination field.

7. If the **Dependencies** tab is displayed, click it, and then provide the information listed in the following table.

Field Name	Description
Clear When the Following Changes	Indicates whether the field is to be cleared when the specified field changes.
Display Only When	Indicates that the field is to be editable only if certain logical criteria are satisfied. The field functions with two adjacent fields, a list that contains logical qualifiers, and a text field. To enable this functionality, on the Attributes tab, from the Display Only list, select Use Dependency Rules .
Required When	Indicates that the field is to be required only if certain logical criteria are satisfied. The field functions with two adjacent fields, a list of logical qualifiers, and a text field. To enable this functionality, on the Attributes tab, from the Required list, select Use Dependency Rules .

8. If the **Security** tab is visible, to specify the users who can view and edit this field, do the following:

- a. Click the **Security** tab.

By default, the new user data field is visible to and editable by all users.



- b. Click **Edit**.

The Edit Field Security window opens.

- c. Configure the security-related fields described in the following table.

Field or Option	Description
Visible to all users	To make the field visible only to specific users or security group members, start by clearing this checkbox. This clears the Editable by all users checkbox and enables fields in the Select User/ Security Group that can view this field section.
Editable by all users	To make the field editable only by specific users or security group members, start by clearing this checkbox. (If you cleared the Visible to all users checkbox, this checkbox is already cleared.) This

Field or Option	Description
	enables fields in the Select User/ Security Group that can edit this field section.
Select Users/ Security Groups that can view this field (list)	<p>To indicate how you want to specify field visibility or editability permission, select one of the following from the list:</p> <ul style="list-style-type: none"> • Security Group. Enables you to specify one or more security groups whose members can view and/or edit the field. This selection dynamically updates the auto-complete displayed under the list to provide security group names. • Username. Enables you to specify one or more user who can view and/or edit the field. (The user must have an email address.) This selection dynamically updates the auto-complete displayed under the list to provide user names. • Standard Token. Enables you to select a standard token that resolves to a security group (based on group name or ID) or user name, or user (based on username or ID) who can view and/or edit the field. • User Defined Token. Enables you to define a token that resolves to a security group (based on group name or ID) or user (based on username or ID) who can view and/or edit the field.
Security Group Username Standard Token or User Defined Token (auto- complete)	<p>The label for this field changes dynamically based on the value you select from the Select Users/Security Groups that can view this field list.</p> <p>Do one of the following:</p> <ul style="list-style-type: none"> • Use the Security Group auto-complete to specify one or more security groups to view and/or edit the field. • Use the Username auto-complete to select the names of one or more users who can view and/or edit the field. • Use the Standard Token auto-complete to select a standard token that resolves to a security group (based on group name or ID) or user name, or user (based on username or ID) who can view and/or edit the field. • Use the User Defined Token auto-complete to define a token that resolves to a security group (based on group name or ID) or user (based on username or ID) who can view and/or edit the field. <p>You can use the Tokens button to access the Token Builder). For information on how to use the Token Builder to create user-defined tokens, see the <i>Commands, Tokens, and Validations Guide and Reference</i>.</p>

Field or Option	Description
Provide Editing Rights	<p>To give the security groups and users you selected (using the auto-complete) permission to edit the user data field, leave this checkbox selected (the default), and then click the Add button .</p> <p>To prevent the security groups and users you selected (using the auto-complete) from editing the user data field, clear this checkbox, and then click the Add button .</p>

9. Click **OK**.

The Field window displays the new field.

10. Click **OK**.

Copying a Field Definition

You can streamline the process of configuring a new field by copying the definition of an existing field.

To copy a field definition:

1. On the PPM Workbench shortcut bar, click **Configuration > User Data**.

The User Data Workbench opens.

2. Open an existing user data type or create a new user data type.

The User Data Context window opens to the **Fields** tab.

3. Click **New**.

The Field: New window opens.

4. Click **Copy From**.

The Field Selection window opens.

5. To search for the field to copy, complete one or more of the fields, and then click **List**.

The Field Selection window lists the fields that match your search criteria.

Note: You can query fields using several criteria, including the token name or field prompt. You can also perform more complex queries. For example, you can list all fields that reference a specific validation or all fields that a specific entity uses.

6. Select the field to copy, and then click **Copy**.

The Field: New window displays the attributes of the copied field.

7. Make any necessary changes, and then click **OK**.

Editing User Data Fields

To edit a user data field:

1. On the PPM Workbench shortcut bar, click **Configuration > User Data**.

The User Data Workbench opens.

2. Open a user data type.

The User Data Context window opens to the **Fields** tab.

3. Select the field to edit, and then click **Edit**.

The Field window opens.

4. Make the required changes, and then click **OK**.

Make sure that you include the **Attributes**, **Default**, and **Dependencies** tabs. For information about these tabs, see ["Configuring User Data Fields" on page 251](#).

5. In the User Data Context window, click **OK**.

Configuring User Data Field Dependencies

Field behavior and properties can be linked to the value of other fields defined for that entity. A **Report Type** field can become required when the value in another field in that report type is **Critical**.

You can configure a field to:

- Clear after the value in another field changes.
- Become read-only after another field meets a logical condition defined in ["Table 10-2. Field dependencies" on the next page](#).
- Become required after another field meets a logical condition defined in ["Table 10-2. Field dependencies" on the next page](#).

Table 10-2. Field dependencies

Logical Qualifier	Description
like	Looks for close matches of the value to the contents of the selected field.
not like	Looks for contents in the selected field that are not close matches to the specified value.
is equal to	Looks for an exact match of the value to the contents of the selected field.
is not equal to	Is true when no results exactly match the value specified in the field.
is null	Is true when the selected field is blank.
is not null	Is true when the selected field is populated.
is greater than	Looks for a numerical value greater than the value specified.
is less than	Looks for a numerical value less than the value specified in the field.
is less than equal to	Looks for a numerical value less than or equal to the value specified.
is greater than equal to	Looks for a numerical value greater than or equal to the value specified.

To configure a user data field dependency:

1. On the PPM Workbench shortcut bar, click **Configuration > User Data**.
The User Data Workbench opens.
2. Open a user data type.
The User Data Context window opens to the **Fields** tab.
3. Select the field, and then click **Edit**.
The Field window opens.
4. Click the **Dependencies** tab.
5. Use the following fields to set the field dependencies:
 - To clear the current field whenever the value in another field changes, from the **Clear When The Following Changes** list, select the field to trigger the clearing of the current field.
 - To make the field become read-only after another field satisfies a logical criterion, from the **Display Only When** list, select the field which, when changed, is to make the current field read-only.

The **Display Only When** list functions with the two lists to the right. One is a list of logical qualifiers (described in ["Table 10-2. Field dependencies" on the previous page](#)), and the other, a list that dynamically changes to a date, list, or text field, depending on the validation specified for the current field.

- To make the current field become required after a selected (trigger) field meets the condition selected from the list of logical qualifiers (described in ["Table 10-2. Field dependencies" on the previous page](#)), from the **Required When** field, select a trigger field

The **Required When** list functions with the two lists to its right. One is a list of logical qualifiers (described in ["Table 10-2. Field dependencies" on the previous page](#)), and the other, a list that dynamically changes to a date, list, or text field, depending on the validation specified for the current field.

6. Click **OK**.
7. In the Field window, click **OK**.
8. In the User Data Context window, click **OK**.

Removing Fields

To permanently remove a field from a user data type:

1. on the PPM Workbench shortcut bar, click **Configuration > User Data**.

The User Data Workbench opens.

2. Select an existing user data type or create a new user data type.

The User Data Context window opens to the **Fields** tab.

3. Select the field to remove, and then click **Remove**.
4. Click **OK**.

Configuring User Data Layouts

The layout of user data fields can be changed in the **Layout** tab of the User Data Context window.

Changing Column Widths

To change the column width of a field:

1. On the PPM Workbench shortcut bar, click **Configuration > User Data**.
The User Data Workbench opens.
2. Select an existing user data type or create a new user data type.
The User Data Context window opens to the **Fields** tab.
3. Click the **Layout** tab.
4. Select the field.
5. From the **Field Width** list, select **1** or **2** (inches).

Note: You cannot make changes that conflict with another field in the layout. For example, you cannot change the width of a field from 1 to 2 if another field exists in column two on the same row.

For fields of component type Text Area, you can determine the number of lines the text area is to display. Select the Text Area type field and change the value in the **Component Lines** attribute. If the selected field is not of type Text Area, this field is not enabled.

6. Click **OK**.

Moving Fields

To move a field or a set of fields:

1. On the PPM Workbench shortcut bar, click **Configuration > User Data**.
The User Data Workbench opens.
2. Select an existing user data type or create a new user data type.
The User Data Context window opens to the **Fields** tab.
3. Click the **Layout** tab.
4. Select the field.

To select more than one field, press **Shift**, and then select the first and last fields in a set. (You can only select adjacent fields.)

Note: You cannot move a field to a position occupied by another field.

5. Use the directional arrows to move the fields in the layout builder.
6. Click **OK**.

Swapping Positions of Two Fields

To swap the positions of two fields:

1. On the PPM Workbench shortcut bar, click **Configuration > User Data**.
The User Data Workbench opens.
2. Select an existing user data type or create a new user data type.
The User Data Context window opens to the **Fields** tab.
3. Click the **Layout** tab.
4. Select the field.
5. Select the **Swap Mode** option.
An **S** is displayed in the option section of the selected field.
6. Double-click the field to swap positions with the selected field.
7. Click **OK**.

Previewing the Layout

You can preview the field layout as it will be displayed to users.

To preview field layout:

- In the User Data Content window, click the **Layout** tab, and then click **Preview**.

The Field Layout Preview window opens and displays the user data fields as they are to be displayed.

If all fields have a width of one column, all displayed columns automatically span the entire available section when an entity of the given user data is viewed or generated.

Configuring User Data for Resources

Pages in PPM Center display a set of standard fields for collecting and displaying information. User data is the set of custom fields that can be defined by the system administrator.

If you want to include additional fields in a resource's page, you can define them in the User Data Workbench.

Resource user data are customizable resource attributes that are added to the **Details** tab in a resource's page. The system administrator configures the resource attributes that are displayed in a resource's page and the resource can modify the values of these resource attributes from the **Details** tab.

Resource user data is not available in the Search Resources page nor can they be modified for multiple resources simultaneously.

To configure user data,

1. Log on to PPM Center.
2. Select **Open > Administration > Open Workbench**.
The PPM Workbench opens.
3. From the shortcut bar, select **Configuration > User Data**.
The User Data Workbench opens.
4. Click **List**.
The Results tab opens with the available user data types.
5. To configure resource user data, select **Resource User Data** and click **Open**.
The User Data Context window opens.
6. To add a field, click **New**.
The Field: New window opens.
7. From the Field: New window, do the following:

- a. Configure up to 100 fields. Enter the following information:
 - i. In the **Field Prompt** box, type the label to display for the new field.
 - ii. In the **Token** box, type an uppercase text string to use to identify this field.

The token name must be unique to the specific user data. An example token name is `ASSIGNED_TO_USER_ID`.
 - iii. In the **Description** box, you can enter text that describes what the field captures and how it is to be used.
 - iv. To enable the new field, leave **Enabled** selected.
 - v. In the **Validation** box, enter or select the validation logic to use to determine the valid values for the field.

This can be a list of user-defined values, a rule that the result must be a number, and so on.
 - vi. The **Component Type** field indicates the field type (list, free-form text field, and so on). This read-only field is derived from the validation you selected.
 - vii. If the field lists selectable items, and you want users to be able to select more than one of these, select **Multiselect**.

If you select **Multiselect**, the PPM Workbench displays a dialog box that lists limitations imposed on multiselect user fields.

If you selected **Multiselect**, make a note of the limitations, and then click **Yes**.
- b. On the **Attributes** tab, enter the following information:
 - i. In the **User Data Col** list, select the internal column in which the field value is to be stored.

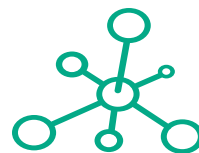
You can store information in up to 100 columns, which means that you can create up to 100 custom fields for resources. No two fields in user data can use the same column.
 - ii. To make the new field read-only at all times, in the **Display Only** list, select **Always**. To make the field editable at all times, select **Never**.
 - iii. To make the field visible to users, next to **Display**, leave **Yes** selected. To hide the field, select **No**.
 - iv. To make the field required (the user must specify a value) at all times, in the **Required** list, select **Always**. To make the field optional at all times, select **Never**.
- c. To configure a default value for the user data field, open the **Default** tab and enter the following information:



Appendix A: Worksheets

- "Configuration Workflow Worksheet" on the next page
- "Execution Workflow Step Worksheets" on page 266
- "Decision Workflow Step Worksheets" on page 267
- "Subworkflow Workflow Step Worksheets" on page 269

- "Request Type Configuration Sheets" on page 270



Hewlett Packard
Enterprise

Configuration Workflow Worksheet

Table A-1. Workflow skeleton

#	Step Name	Description	Type ^a	Transition Values
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				
19				
20				
a. Type = Workflow Step Type: Decision (D), Execution (E), Condition (C), Subworkflow (S)				

Execution Workflow Step Worksheets

Table A-2. Workflow step [execution], step number ____

Workflow Step	Value
Step Name	
Goal/Result of Step	
Validation	See "Table A-3. Validation Information" on the next page
Execution Type	See "Table A-4. Workflow step [execution], step number ____ execution type" on the next page
Processing Type	
Timeout (Days)	
Source Environment (Group)	
Dest Environment (Group)	
Security (who can act on step): <ul style="list-style-type: none"> User Name Standard Token User Defined Token 	
Include Notification (Yes/No)	
Notification Event	
Notification Recipient: <ul style="list-style-type: none"> Username Email Address Security Group Standard Token User Defined Token 	
Notification Message	
Request Status at Step	
Request % Complete at Step	

Table A-2. Workflow step [execution], step number ____, continued

Workflow Step	Value
Authentication Required (Y/N)	
Authentication Type (if Y)	

Table A-3. Validation Information

Validation Information	Value
Existing Validation?	
New Validation?	
Validation Type: (text field, auto-complete, list, and so on.)	
Validation Definition (list of values or SQL)	

Table A-4. Workflow step [execution], step number ____ execution type

Execution Type	Value
Built-in Workflow Event: <ul style="list-style-type: none"> • Execute Commands • Close • Jump/Receive • Ready for Release • Return from Subworkflow 	
PL/SQL Function	
Token	
SQL Statement	
Workflow step commands	

Decision Workflow Step Worksheets

Table A-5. Workflow step [decision], step number ____

Workflow Step	Value
Step Name	
Goal/Result of Step	

Table A-5. Workflow step [decision], step number ____, continued

Workflow Step	Value
Validation	
Decisions Required (Vote on Step's outcome?)	<ul style="list-style-type: none"> • One • At Least One • All
Timeout (Days)	
Security (who can act on step): <ul style="list-style-type: none"> • Security Group • User Name • Standard Token • User Defined Token 	
Include Notification (Yes/No)	
Notification Event	
Notification Recipient: <ul style="list-style-type: none"> • Username • Email Address • Security Group • Standard Token • User Defined Token 	
Notification Message	
Request Status at Step	
Request % Complete at Step	
Parent Assigned To User	
Authentication Required (Y/N)	
Authentication Type (if Y)	

Table A-6. Workflow step [decision], step number ____, validation

Validation Information*	Value
Existing Validation?	

Table A-6. Workflow step [decision], step number ____ validation, continued

Validation Information*	Value
New Validation?	
Validation Type: (text field, auto-complete, list, and so on.)	
Validation Definition (list of values or SQL)	

Subworkflow Workflow Step Worksheets

Table A-7. Workflow step [subworkflow], step number ____

Workflow Step	Value
Step Name	
Goal/Result of Step	
Validation*	
Vote on Step outcome?	
Timeout (Days)	
Source Environment (Group)	
Dest Environment (Group)	
Security (who can act on step): <ul style="list-style-type: none"> • Security Group • User Name • Standard Token • User Defined Token 	
Include Notification (Yes/No)	
Notification Event	
Notification Recipient: <ul style="list-style-type: none"> • Username • Email Address • Security Group • Standard Token • User Defined Token 	

Table A-7. Workflow step [subworkflow], step number ____, continued

Workflow Step	Value
Notification Message	
Request Status at Step	
Request % Complete at Step	
Authentication Required (Y/N)	
Authentication Type (if Y)	

Table A-8. Workflow step [subworkflow], step number ____ validation

Validation Information*	Value
Existing Validation?	
New Validation?	
Validation Type: (text field, auto-complete, list, and so on)	
Validation Definition (list of values or SQL)	

Request Type Configuration Sheets

Table A-9. Request type information

Information	Value
Request Type Name	
Associated Request Header Type	
Description	

Table A-10. Request type field information

#	Field Name	Description
1		
2		
3		
4		
5		
6		

Table A-10. Request type field information, continued

#	Field Name	Description
7		
8		
9		
10		
11		
12		
13		
14		
15		

Table A-11. Request type commands

Command	Value
Goal of Commands	
Command Steps	
Conditions (When to execute)	

Table A-12. Request type status values

Status	Corresponds to Workflow Step

Table A-12. Request type status values, continued

Status	Corresponds to Workflow Step

Table A-13. Request type attributes

Information	Value
Field Name	
Validation	
Field Behavior	
Attributes (select one):	<ul style="list-style-type: none"> • Display • Editable • Display Only • Required
Default Value	
Users/Security Groups allowed to View Field	
Users/Security Groups allowed to Edit Field	
Status Dependencies	
Clear field when Status = ?	
Display only when Status = ?	
Reconfirm only when Status = ?	
Required when Status = ?	
Auto-Population Behavior	
Auto-Population triggered by (Depends on) Field:	
Value used to populate Field:	

Table A-14. Field validation information

Validation Information	Value
Existing Validation?	
New Validation?	
Validation Type: (text field, auto-complete, list, and so on.)	
Validation Definition (list of values or SQL)	
Notes on Validation (data masks, auto-complete behavior, and so on.)	

Table A-15. Request header type information

Request Header Type	Value
Request Header Type Name	
Associated Request Type(s)	
Description	
Associated Field Group(s)	

Table A-16. Existing request header type field information

Prompt	Display	Display Only	Transaction History	Notes History	Search Filter Page
Request No					
Request Type					
Created By					
Department					
Sub-Type					
Created On					
Workflow					
Request Status					
Priority					
Application					
Contact Name					
Assigned To					
Assigned Group					

Table A-16. Existing request header type field information, continued

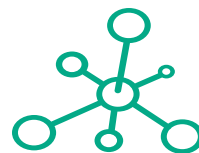
Prompt	Display	Display Only	Transaction History	Notes History	Search Filter Page
Contact Phone					
Request Group					
Contact Email					
Description					
Company					
% Complete					



Appendix B: Examples

- "A Simple PL/SQL Function for Execution Steps and Field Population" below
- "Executing PL/SQL Functions With A Parameter and Creating Transitions Based on the Results" on page 277
- "Examples of Using Advanced Rule with SQL-default Logic" on page 279

A Simple PL/SQL



Hewlett Packard
Enterprise

Function for Execution Steps and Field Population

In this example, you create a simple PL/SQL function to return a boolean value. Then you create PL/SQL function execution workflow steps to run the PL/SQL function and return its results as the results of the PL/SQL function execution workflow steps. You will configure the workflow transitions and run the workflow to populate a request type field.

Follow the steps below:

1. Create a simple function.

```
CREATE OR REPLACE FUNCTION yes_or_no
  RETURN VARCHAR2
IS
  tmpvar  VARCHAR2 (10);
BEGIN
  tmpvar := 'YES';
  RETURN tmpvar;
EXCEPTION
  WHEN NO_DATA_FOUND
  THEN
    NULL;
  WHEN OTHERS
  THEN
    RAISE;
END yes_or_no;
```

2. Log on to PPM Center, and open PPM Workbench. On the shortcut bar, select **Configuration > Validations**, click **New Validation** and create a new validation.

Make sure to select **Enabled** and **Use in Workflow** checkboxes.

Save the validation.

3. On the PPM Workbench shortcut bar, select **Configuration > Workflows**. Click **List**, locate a workflow of your choice and open it.
 - a. From the Workflow Step Sources window, click **New**. The Execution window opens.
 - b. For the **Execution Type** field, select **PL/SQL Function** from the drop-down list; For the **Validation** field, select the validation you just created in [step 2](#); Type the function you created in [step 1](#) in the **Execution** text area.

Note: Other option for the **Execution Type** field: **SQL Statement**.

- c. Click **Verify**.
- d. Save the execution step.
4. Go to **Layout** tab of the workflow, drag necessary workflow steps from the Workflow Step Sources window and drop to the **Layout** tab area, configure the transitions, and save your changes.
5. Create a new Request Type with a new field.
 - a. On the PPM Workbench shortcut bar, click **Demand Mgmt > Request Types**. The Request Type Workbench opens.
 - b. Click **New Request Type**. Provide values as necessary.
 - c. Go to the **Fields** tab, click **New** to add a new field.
 - d. Go to **Commands** tab, create a command to populate the new field.
 - e. Go to **Rules** tab, add a rule for the request type.
 - f. Save the changes.
6. Go back to the Workflow workbench. Create a workflow Execution step, later you will use which to execute the command you just created in [step d](#).
7. Add a step to the workflow to populate the field using the command you just created.

Note: Make sure to set **Source Environment**.

8. In PPM Center, create a request using the Request Type you created in [step 5](#) and submit it.
9. Open the request you just created, click **Execute Now** to trigger the PL/SQL function.
10. Click **View Full Status Below > Graphical View** to view the execution status.

Executing PL/SQL Functions With A Parameter and Creating Transitions Based on the Results

In this example, you create a PL/SQL function with a parameter that can be passed. Then you create PL/SQL function execution workflow steps to run the function in order to route a PPM workflow based on the results the function returns. You will configure the workflow transitions and execute the PL/SQL function execution workflow steps to pass the parameter in order to populate request type fields.

Perform the following steps.

1. Create a function with a parameter that can be passed.

```
CREATE OR REPLACE FUNCTION sample_param (p_num IN NUMBER)
RETURN NUMBER IS
tmpVar NUMBER;
BEGIN
    tmpVar := p_num;
    RETURN tmpVar;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        NULL;
    WHEN OTHERS THEN
        -- Consider logging the error and then re-raise
        RAISE;
END sample_param;
/
```

2. Create a validation to match the values to be returned by the function.

Make sure to select **Enabled** and **Use in Workflow** checkboxes.

Save the validation.

3. Create a new workflow and a workflow Execution step.

- a. On the PPM Workbench shortcut bar, select **Configuration > Workflows**. Click **New Workflow**. In the workflow window that opens, provide necessary values.
- b. From the Workflow Step Sources window, click **New**. The Execution window opens.
- c. For the **Execution Type** field, select **PL/SQL Function** from the drop-down list; For the **Validation** field, select the validation you just created in [step 2](#); Type the function you created in [step 1](#) in the **Execution** text area.
- d. Click **Verify**.
- e. Save the execution step.

4. Create a new Request Type with two new fields.

- a. On the PPM Workbench shortcut bar, select **Demand Mgmt > Request Types**. The Request Type Workbench opens.
- b. Click **New Request Type**. Provide values as necessary for the new request type.
- c. On the **Fields** tab, add two new fields using the validation you created in [step 2](#).
- d. Go to **Commands** tab, create a command to populate the new fields.

- e. Go to **Rules** tab, add a rule for the request type using the workflow you created earlier.
 - f. Save the changes.
5. Go back to the Workflow workbench. Create a workflow Execution step to execute the command you just created in [step d](#).
 6. Go to **Layout** tab of the workflow, add the execution step to the workflow to populate the new fields using the command you just created.
 - From the Workflow Step Sources window, drag the workflow step of your choice and drop it to the **Layout** tab area.
 - In the Workflow Step window that opens, set **Source Environment**.
 - Go to **Security** tab, click **New**. In the Workflow Step Security dialog box, add yourself to make sure you have the step execution security.
 - Configure the transitions, and save your changes.

Note: **Other Results** are based on the validation.

7. In PPM Center, create a request using the Request Type you created earlier and submit it.
8. Open the request you just created, set a value for the first field that will be used in PL/SQL Function.
9. The toolbar displays available actions you can take for the first execution step you configured earlier. Click **Execute Now**.

The execution step with PL/SQL function determines which transition to take next based on the value you set for the first field. In this example, it is Step 4 (as shown below).

From the toolbar, click **Execute Now**.

10. The second request type field is successfully populated by the execution steps.
Click **Graphical View** to view the execution status.

Examples of Using Advanced Rule with SQL-default Logic

- ["PL/SQL Function Example" on the next page](#)
- ["Example A" on page 281](#)

PL/SQL Function Example

In this example, you create a PL/SQL function, and a SQL statement that returns a single row with two values. You will create an advanced rule with SQL-default logic to set a new value in any fields in the request, based on the SQL statement.

1. Create a function with a parameter that can be passed.

```
CREATE OR REPLACE FUNCTION sample_param (p_num IN NUMBER)
RETURN NUMBER IS
tmpVar NUMBER;
BEGIN
    tmpVar := p_num;
    RETURN tmpVar;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        NULL;
    WHEN OTHERS THEN
        -- Consider logging the error and then re-raise
        RAISE;
END sample_param;
/
```

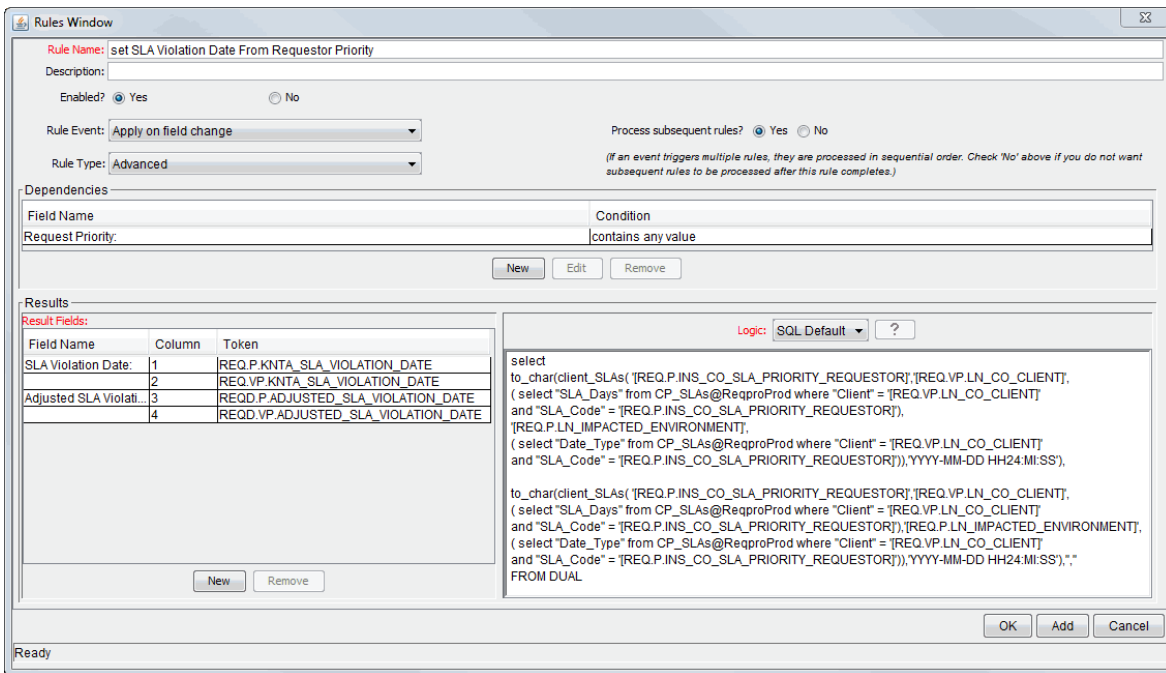
2. Create a new Request Type and add two new fields.
 - a. On the PPM Workbench shortcut bar, select **Demand Mgmt > Request Types**. The Request Type Workbench opens.
 - b. Click **New Request Type**. Provide values as necessary for the new request type.
 - c. On the **Fields** tab, add two new fields.
 - d. Go to the **Rules** tab, create an advanced rule for the request type that uses the PL/SQL function.
 - e. Save the changes.
3. In PPM Center, create a request using the Request Type you just created and submit it.
4. Open the request you just created, set a value for the first field based on the SQL Statement to trigger the PL/SQL function. The second field is populated automatically.

Syntax Examples

Example A

The following example shows a query with `select` statement in relation to PPM when using tokens. The syntax includes single quotes when a nested token is of `VARCHAR2` data type.

Note: No need to include single quotes when a nested token is of `Number` data type.



Syntax:

```
-- Set SLA Violation Date From Requestor Priority
select
  to_char(client_SLAs( '[REQ.P.INS_CO_SLA_PRIORITY_REQUESTOR]',
    '[REQ.VP.LN_CO_CLIENT]',
    ( select "SLA_Days" from CP_SLAs@ReqproProd
      where "Client" = '[REQ.VP.LN_CO_CLIENT]'
        and "SLA_Code" = '[REQ.P.INS_CO_SLA_PRIORITY_
REQUESTOR]'
    ),
    '[REQ.P.LN_IMPACTED_ENVIRONMENT]',
    ( select "Date_Type" from CP_SLAs@ReqproProd
      where "Client" = '[REQ.VP.LN_CO_CLIENT]'
```

```

                                and "SLA_Code" = '[REQ.P.INS_CO_SLA_PRIORITY_
REQUESTOR]')
                                ),
                                'YYYY-MM-DD HH24:MI:SS'
                                ),
to_char(client_SLAs( '[REQ.P.INS_CO_SLA_PRIORITY_REQUESTOR]',
                    '[REQ.VP.LN_CO_CLIENT]',
                    ( select "SLA_Days" from CP_SLAs@ReqproProd
                      where "Client" = '[REQ.VP.LN_CO_CLIENT]'
                        and "SLA_Code" = '[REQ.P.INS_CO_SLA_PRIORITY_
REQUESTOR]'
                    ),
                    '[REQ.P.LN_IMPACTED_ENVIRONMENT]',
                    ( select "Date_Type" from CP_SLAs@ReqproProd
                      where "Client" = '[REQ.VP.LN_CO_CLIENT]'
                        and "SLA_Code" = '[REQ.P.INS_CO_SLA_PRIORITY_
REQUESTOR]'
                    ),
                                ),
                                'YYYY-MM-DD HH24:MI:SS'
                                ),
                                ',
                                ',
                                ',
FROM DUAL

```

Example B

Here is a syntax example of PL/SQL function:

```
select LN_CUSTOM.ValidateCompanions('[REQ.P.LNCOMPANIONREQNO]') from dual
```

This syntax launches a select statement and returns the result back into LN_WARNING.

Rules Window

Rule Name:

Description:

Enabled? Yes No

Rule Event: Process subsequent rules? Yes No

Rule Type: (If an event triggers multiple rules, they are processed in sequential order. Check 'No' above if you do not want subsequent rules to be processed after this rule completes.)

Dependencies

Field Name	Condition
Companion Request No:	contains any value

Results

Result Fields:

Field Name	Column	Token
Warning Message:	1	REQD.P.WARNING_MESSAGE
	2	REQD.VP.WARNING_MESSAGE

Logic:

```
select LN_CUSTOM.ValidateCompanions(['REQ.P.LNCOMPANIONREQNO']) from dual
```

Ready

Appendix C: Configuring A Single Email Notification for Multiple Recipients with Different Locales

When sending email notifications of a workflow step or request field change, PPM Center behaves as follows by default:

- For PPM Center users who have already logged on to PPM Center at least once (or valid PPM users), the columns LANGUAGE_FORMAT_CODE and REGION_FORMAT_CODE in the table KNTA_USER_REGIONAL_SETTINGS indicate their user locales. PPM send email notifications according to their user locales.
- For non-PPM users and PPM Center users who have never logged on to PPM Center (or invalid PPM users), the columns LANGUAGE_FORMAT_CODE and REGION_FORMAT_CODE for them are null. PPM sends freestanding emails to them.

As a result, email notifications are sent to recipients in several batches. To avoid split email issue, do the following:

1. Set the USE_SERVER_LOCALE_FOR_NOTIFICATIONS server configuration parameter value to true. This ensures that PPM sends an email notification to all recipients in one go.

The USE_SERVER_LOCALE_FOR_NOTIFICATIONS parameter flags whether or not to check notification recipient regional settings. Setting the parameter value to true ignores recipients' regional settings, and uses the values of server configuration parameters SERVER_LOCALE_COUNTRY_CODE and SERVER_LOCALE_LANGUAGE_CODE instead. This ensures that recipients are not split into different groups according to their regional settings.

2. Set a value for the SERVER_LOCALE_COUNTRY_CODE server configuration parameter in the server.conf file.

Valid values are any two-letter abbreviation of a country in uppercase. For example, if you want to set the regional settings to United States, set the SERVER_LOCALE_COUNTRY_CODE server configuration parameter to US.

3. Set a value for the SERVER_LOCALE_LANGUAGE_CODE server configuration parameter in the server.conf file.

Valid values are any two-letter abbreviation of a language in lowercase. For example, if you want to set the regional settings to United States, set the `SERVER_LOCALE_LANGUAGE_CODE` server configuration parameter to `en`.

Note: These parameters have no impact to the notifications from the Resource Management module.

When you set the `USE_SERVER_LOCALE_FOR_NOTIFICATIONS` to `false`, the server locale parameters `SERVER_LOCALE_COUNTRY_CODE` and `SERVER_LOCALE_LANGUAGE_CODE` are used to define default locale for non-PPM users and invalid PPM users. The values of the server locale parameters replace the values of the columns `REGION_FORMAT_CODE` and `LANGUAGE_FORMAT_CODE` respectively.

In this case, non-PPM users and invalid PPM users will be grouped with those valid PPM users who share the same users locale as defined by the server locale parameters. In this case, non-PPM users and invalid users are regarded as valid PPM users in receiving email notifications. However, email notifications are sent in batches in different languages to recipients according to their regional settings.

Note: The parameters `SERVER_LOCALE_COUNTRY_CODE` and `SERVER_LOCALE_LANGUAGE_CODE` cannot be null, or email notifications cannot be sent to external email addresses.

Appendix B: Appendix D: Switching Off Pagination on Builder Portlets of Requests Category

Starting from version 9.30, builder portlets of the Requests category do not display the total number of records by default for better system performance. When there are entries on the next page, the page navigation buttons are enabled.

Considering the complexity of real data source SQLs used in customer's business, a switch is available for you to disable the new pagination feature at data source level. You can add the the `/*NOPAGINATION*/` tag into the particular SQL statement of your concern. This allows you to disable the feature just in case there are some corner cases that the new pagination solution fails to cover. For example, if you have a data source SQL statement, like the following, not working properly as it was,

```
select distinct request_type_id from kcrt_requests
```

You can switch to non-pagination logic by adding the `/*NOPAGINATION*/` tag into the SQL statement, as follows:

```
/*NOPAGINATION*/ select distinct request_type_id from kcrt_requests
```

Note:

- The `/*NOPAGINATION*/` tag is case-insensitive.
- This tag does not have to be added to the very beginning of a SQL statement. Actually you can add it anywhere as long as the SQL syntax is not broken.
- This tag disables pagination for this particular data source only.

However, HPE encourages you to take a look at such complex SQLs and follow the ["Best Practices on Builder Portlets Pagination"](#) below to re-factor them.

Best Practices on Builder Portlets Pagination

This section provides some best practices to help you eliminate the impact of the pagination limitation in your custom SQLs.

- **Avoid suppressing index**

- Do not use "<>" or "!=" when it is possible to use "="
- Do not use "is null" or "is not null" when it is possible to use some specified value
- Do not use "like" when the parameter value is certain
- Create function-based index when a function is used in condition expressions
- Pay attention to type mismatch. For example, "where varchar2column=1234" will suppress index on varchar2column, you should use "where varchar2column='1234'"

- **Optimize Nested Query as much as Possible**

- Nested query can appear in SELECT statements, FROM clauses, and WHERE clauses. Use as less nested queries as possible.
- Never use ORDER BY clause in inline views
- Use materialized views to replace inline views whenever possible

- **Do not use Row_number()**

This analytical function will introduce sorting as well. It may cause conflict and performance issue when the original SQL statements are transformed.

- **Optimize access control**

- Do not use KCRT_PARTICIPANT_CHECK.is_participant_of_request(). Instead, join KCRT_PARTICIPANT_CHECK_V.
- Do not involve KCRT_PARTICIPANT_CHECK_V, KNTA_ELIGIBILITY_CHECK_V, or KNTA_FIELD_SECURITY_V unless it is definitely necessary. Consider using some alternative filters, for example, create_by, create_date, status, or use a snippet from those views.

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 Demand Management Configuration Guide (Project and Portfolio Management Center 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 your_IE_team_PDL@hpe.com.

We appreciate your feedback!