HP Project and Portfolio Management Center

Software Version: 7.1

HP Deployment Management Configuration Guide

Document Release Date: March 2007
Software Release Date: March 2007

Legal Notices

Mercury provides links to external third-party Web sites to help you find supplemental information. Site content and availability may change without notice. Mercury makes no representations or warranties whatsoever as to site content or availability.

Documentation Updates

This manual's title page contains the following identifying information:

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

To check for recent updates, or to verify that you are using the most recent edition of a document, go to: http://ovweb.external.hp.com/lpe/doc_serv/.

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

Support

Mercury Product Support

You can obtain support information for products formerly produced by Mercury as follows:

- If you work with an HP Software Services Integrator (SVI) partner (www.hp.com/managementsoftware/svi_partner_list), contact your SVI agent.
- If you have an active HP Software support contract, visit the HP Software Support site and use the Self-Solve Knowledge Search to find answers to technical questions.
- For the latest information about support processes and tools available for products formerly produced by Mercury, we encourage you to visit the HP-Mercury Software Support web site at: support.mercury.com.
- Contact your HP Sales Representative if you have additional questions.

HP Software Support

You can visit the HP Software Support web site at www.hp.com/managementsoftware/services.

HP Software online support provides an efficient way to access interactive technical support tools. As a valued support customer, you can benefit by using the support site to:

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

Most of the support areas require that you register as an HP Passport user and sign in. Many also require a support contract. To find more information about access levels, go to: www.hp.com/managementsoftware/access_level.

To register for an HP Passport ID, go to: www.managementsoftware.hp.com/passport-registration.html.

Table of Contents

List of Figures

List of Tables

1 Getting Started with HP Deployment Management Configuration

In This Chapter:

Introduction to HP Deployment Management

HP Deployment Management is a HP Project and Portfolio Management Center (PPM Center) application that automates and manages the migration and deployment of changes for packaged applications, custom applications, legacy systems, web content, and more. Leveraging and enforcing best practice deployment processes, HP Deployment Management performs all tasks required to install software changes correctly across your system landscape. With HP Deployment Management, packages are deployed automatically, eliminating errors inherent with manual processes. HP Deployment Management's detailed audit trail helps you pinpoint problems quickly, roll back changes if necessary, and supports regulatory compliance requirements, such as segregation of duties (SOD), at both the role level and the process step level.

HP Deployment Management allows developers to attach changes as package line items within a package. Each package line uses an object type to tell the system the type of a change. Packages then follow a digitized process to ensure they are appropriately handled with reviews and approvals. Based on the process, each package line (change) can then be deployed to the environments defined, such as development, test, and production.

This document provides the details on how to configure a HP Deployment Management system using the PPM Workbench, and making sure your packages follow your digitized process. Chapter 1, *Getting Started with HP Deployment Management Configuration,* on page 15 (this chapter) provides an overview of how to configure HP Deployment Management to support your business processes.

HP Deployment Management Concepts

The following are the major concepts of HP Deployment Management.

- **Deployment.** Deployment is moving a file, script, code, or full application (object) between two or more instances. For example, a file can be deployed from a development instance to a testing instance and finally into one or more production instances. Deployment typically involves connecting systems together, moving files between the systems, and running support scripts.
- **Environments.** Environments can consist of a server, a single database instance, or an associated remote client machine. Not all of these components need be present in an environment. For example, it is possible to have an environment which does not contain a database.
- **Environment Groups.** Environment groups define a set of environments which can be referenced as the source and destinations for object migrations and deployments.
- **Environment Refresh.** An environment refresh replaces one environment with another environment. After the physical refresh, data in Deployment Management is updated to be consistent with the refreshed environment.
- **Object Types.** Object types are used to define the technical steps required to migrate or deploy application files or changes for packaged applications, custom applications, legacy systems, web content. For example, a File Migration object type might contain the information and commands required to transfer a file from one machine to another, while a SQL script object type might address the migration and execution of database scripts.
- **Notification Templates.** Notification templates are pre-configured notification forms that can be selected and used with the various HP Deployment Management entities, such as workflows and packages.
- **Package.** A package is a set of objects (package lines) being migrated or deployed together. The package follows an assigned workflow through the various review, approval, and migration steps. Each object in a package is defined by a separate package line. While each package line is acted upon separately, the group of package lines (the package) represent a logical unit that should be moved and tracked together.
- **Package Lines.** A package line is one object being migrated or deployed within the package. Each package line follows the assigned workflow through the various review, approval, and migration steps.

- **Physical Refresh.** A physical refresh replaces one environment with a physical copy of another environment.
- **Refresh Group.** A refresh group reflects the environment being refreshed and the environment providing the new environment.
- **Release and Release Distribution.** Release Management introduces repeatable, reliable processes surrounding software and application releases. HP Deployment Management provides an interface for grouping and processing the packages and requests associated with a specific release. Groups of related packages can then be activated from a single window.
- **User Data.** HP Deployment Management entities such as packages and workflows include a set of standard fields that provide information concerning those entities. While these standard fields are normally sufficient for day-to-day processing, you can add “user data” fields to capture additional information specific to your business process.
- **Workflow.** A workflow is a digitized process were a logical series of steps define business process. Workflow steps can range in usage from review and approvals to performing migrations and deployments.

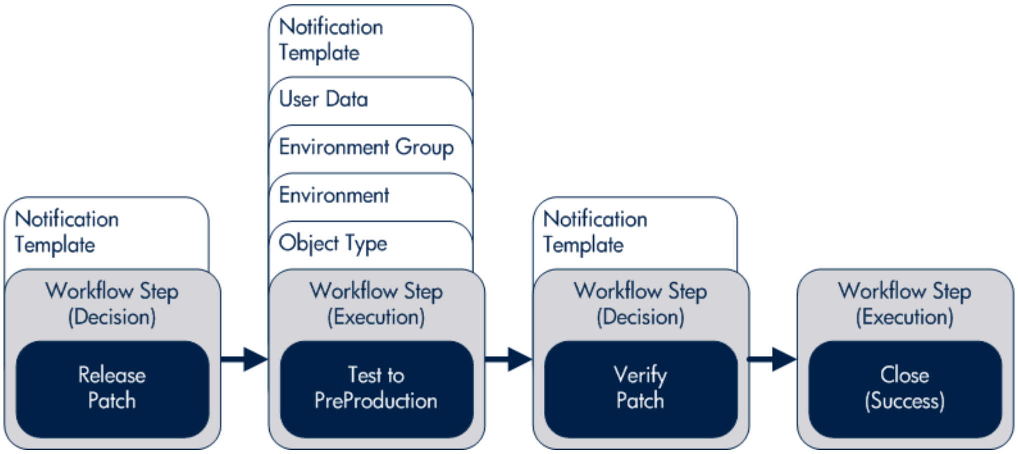
Overview of a Simplified Deployment Process

Figure 1-1 illustrates a simple four-step deployment process for a patch release.

The first step, **Release Patch,** is a decision step where a user is notified that a decision needs to be made, in this case, should the patch be released. Once manually approved the process moves to the second step.

The second step, **Test to PreProduction,** is an execution step where the patch (object) is automatically deployed from one environment to another environment. As part of this second step, the patch is deployed and a user is notified that the deployment has been completed. Once the execution step is complete, the process moves to the third step.

Figure 1-1. PPM Center components



The third step, **Verify Patch,** is a decision step where a user is notified to check and verify that the patch successfully deployed from one environment to another environment. Once verified and manually approved, the process moves to the fourth and final step.

The fourth step, **Close (Success),** is an execution step that automatically closes the deployment process and automatically notifies users of the successful closure of the patch release.

Overview of Configuring Deployment Management

A HP Deployment Management system can be configured using the following steps:

Step 1: Gather process requirements

Before configuring a HP Deployment Management system, you should collect specific information concerning your process, the types of objects to be deployed, source and destination environments, and so on.

For detailed information, see Chapter 2, *Gathering Process Requirements,* on page 23. Appendix A, *Worksheets,* on page 267 also provides a series of worksheets to help gather the information required to build a deployment system.

Step 2: Configure workflows

Configuring workflows involves setting up the required workflow steps (decision and execution), adding transitions between the steps, and configuring each workflow step for notifications, security groups, segregation of duties, and so on. For information on how to configure workflows, see Chapter 3, *Configuring Workflows,* on page 45 and Chapter 4, *Configuring Workflow Components,* on page 105. Chapter 6, *Configuring Releases and Distributions,* on page 167 also details specific configuration requirements for HP Deployment Management's release process. Appendix A, *Worksheets,* on page 267 also provides a series of worksheets to help gather the information required to configure a workflow.

Step 3: Configure object types

Object types define the technical step or steps required to "copy" a change from one environment to another environment. For information on how to configure object types, see Chapter 5, *Configuring Object Types,* on page 139. Appendix A, *Worksheets,* on page 267 also provides a series of worksheets to help gather the information required to configure an object type.

Step 4: Configure environments

Configuring environments means defining the server and database instances identified by your business or deployment process. For information about how to configure environments, environment groups, and environment refreshes, see Chapter 7, *Configuring Environments,* on page 193.

Step 5: Configure notification templates

Notification templates are preconfigured notification "formats" used by other HP Deployment Management entities, such as workflows, object types, and environments. Chapter 8, *Configuring Notification Templates,* on page 235 discusses how to configure notification templates.

Step 6: Configure user data fields

User data fields add customized data fields used by other HP Deployment Management entities, such as workflows, object types, and environments. Chapter 9, *Configuring User Data,* on page 247 discusses how to configure these user data fields.

Step 7: Configure your security and access requirements

Part of any migration or deployment process are the permissions required to perform various decisions or executions. Two of the ways in which PPM Center controls access to perform these decisions or executions are by licenses and access grants.

- **Licenses.** Licenses provide users with access to a PPM Center product such as HP Deployment Management, but they do not dictate what actions the users can perform.
- **Access Grants.** Access grants, when used with licenses, define the actions a user can perform within a PPM Center product.

For more information concerning security and access grants, see the *Security Model Guide and Reference*.

Once configured and tested: Educate your deployment system users

Once your Deployment Management system is configured and tested, you should train your users on the new business process. The following offers some guidance on how to education your Deployment Management system users:

- **Basic HP Deployment Management training.** Ensure that each user understands how to create, process, and report on packages.
- **Process-specific training.** Ensure that each user understands the new process. Consider holding a formal meeting or publishing documents on the workflow steps and packages.
- **User Responsibilities.** Ensure that each user understands their individual role in the process. For example, the QA team may be restricted to approving the testing phase of a package. You can use email notifications that are part of Deployment Management to communicate information about user roles. Notifications can be very detailed, informing the recipients of their responsibility.

Related Information

The following documents also include information related to configuring HP Deployment Management:

- *HP Deployment Management User's Guide*
- *Commands, Tokens, and Validations Guide and Reference*
- *Open Interface Guide and Reference*
- *Reports Guide and Reference*
- *Security Model Guide and Reference*
- *Creating Portlets and Modules*
- *HP-Supplied Entities Guide* (includes descriptions of all HP Deployment Management portlets, object types, and workflows)

2 Gathering Process Requirements

In This Chapter:

Overview of Gathering Process Requirements

Before configuring a HP Deployment Management process, you should gather information concerning your process, such as the steps in the workflow, the objects to be deployed, and the source and destination environments. Once this information is collected, you can then begin configuring your HP Deployment Management process.

This chapter offers you guidance on how to gather information for your deployment process. The subjects covered in this chapter are:

- **Defining workflows.** What are the steps your deployment process (workflow). Which steps require manual decisions (reviews and approvals)? Which steps require automatic executions? See *Defining Workflows* on page 25 for detailed information.
- **Defining object types.** What are you deploying? Are you deploying files? Are you deploying data? What scripts might be required to support the deployment? See *Defining Object Types* on page 33 for detailed information.
- **Defining environments.** What is the source environment for your deployment? What are the destination environments for your deployment? See *Defining Environments* on page 38 for detailed information.
- **Defining notification templates.** Is the correct notification template in place? Does your deployment require a new notification template? See *Defining Notification Templates* on page 40 for detailed information.
- **Defining user data fields.** Does your deployment require additional user information to process correctly? See *Defining User Data Fields* on page 41 for detailed information.
- **Defining security and access.** Who will be allowed to build the deployment packages? Who should receive notifications? Who will be allowed to approve the deployment at each step? See *Defining Security and Access* on page 42 for detailed information.

Defining Workflows

A workflow is a digitized process were a logical series of steps define the path followed by a package. Workflow steps can range from reviews and approvals to performing the actual migration.

When defining a deployment workflow, you must first determine the intent of the business process the deployment workflow will follow. For example:

- Are you designing a simple migration of a file from one environment to another environment with little oversight or supervision?
- Are you designing a business-wide deployment process with a great deal of oversight and supervision?

Once you understand the intent of the business process, you can then begin to define the deployment workflow itself. The following lists the basic components of a workflow:

- **Workflow steps.** Workflow steps are the events (steps) of the process (workflow)? What starts your business process? Where are decisions made? Where are actions taken? The following lists the different types of workflow steps:
    - **Decision steps.** Decision steps are workflow steps that require an external process to decide their outcome.
    - **Execution steps.** Execution steps are workflow steps that perform actual work or actions.
    - **Condition steps.** Condition steps are logic steps used for complex workflow processing.
    - **Subworkflows steps.** Subworkflow steps represent multiple workflows steps that follow a consistent pattern.
- **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.** Each workflow step includes a list of who will be notified about the workflow step. Who is notified? When does the notification occur?

Gathering Information for Workflow Steps

Workflow steps are the events of the process. HP Deployment Management employs the following types of workflow steps:

- **Decision steps.** Decision steps are workflow steps that require an external process to decide their outcome, such as reviews, approvals, or coding.
- **Execution steps.** Execution steps are workflow steps that perform actual work or actions, such as file migration, automatic time-stamping, or automatic package status changes.
- **Condition steps.** Condition steps are logic steps used for complex workflow processing, such as AND and OR.
- **Subworkflows steps.** Subworkflow steps represent multiple workflows steps that follow a consistent pattern, such as code rework or unit testing.

Gathering Information for Decision Steps

Decision steps are workflow steps that require an external process to decide their outcome, such as reviews, approvals, or coding. *Table 2-1* provides a configuration consideration checklist to help define decision steps. See Appendix A, *Worksheets,* on page 267 for a complete list of decision step considerations.

Table 2-1. Decision Workflow Checklist

Complete	Decision Step Check Item	Example
	What is the name of this workflow step?	■ Review Request ■ On Hold ■ In Rework
	What is the status of the package at this workflow step?	■ On Hold ■ New ■ In Review
	What are the transitions from this workflow step?	■ Assign ■ Review ■ On Hold

Table 2-1. Decision Workflow Checklist

Complete	Decision Step Check Item	Example
	Who or what groups can act on this step (approve, cancel, reassign)?	■ Security Groups ■ Users ■ Tokens
	How many decisions are required to exit this workflow step?	■ Only one ■ At Least One ■ All
	What event triggers the notification?	■ When the process reaches the workflow step ■ When a specific result is reached
	Who or how many receive the notification?	■ Email Address (group alias) ■ Security Group
	What is the notification message.	■ Test complete. ■ Approval required.
	Use this workflow step as a timeout? How long?	■ 1 day ■ 2 days
	Are you using segregation of duties?	■ Based on owner of the workflow? ■ Based on workflow step?

Gathering Information for Execution Steps

Execution steps are workflow steps that perform actual work or actions, such as file migration, automatic times-stamping, or automatic package status changes. *Table 2-2* provides a configuration consideration checklist to help define execution steps. See Appendix A, *Worksheets,* on page 267 for a complete list of execution step considerations.

Table 2-2. Execution Workflow Checklist

Complete	Execution Step Check Item	Example
	What is the name of this workflow step?	■ Create Package ■ Close ■ Set Temp Date
	Will this workflow step execute this command?	■ Cancel project ■ Update project
	Will the object type (package line) execute this command?	■ Open database ■ Copy file to environment
	What is the source environment?	(If required.) ■ KINTANA_SERVER
	What is the destination environment?	(If required.) ■ KINTANA_SERVER
	What are the transitions from this workflow step?	■ Succeeded ■ Failed
	Who owns this execution step?	■ Security Group ■ User
	What event triggers the notification?	■ When the process reaches the workflow step. ■ When a specific result is reached.
	Who or how many receive the notification?	■ Email Address (group alias) ■ Security Group
	What is the notification message.	■ Test complete. ■ Approval required
	Use this workflow step as a timeout? How long?	■ 1 day ■ 2 days
	Are you using segregation of duties?	■ Based on owner of the workflow? ■ Based on workflow step?

Gathering Information for Condition Steps

Condition steps are logic steps used for complex workflow processing, such as AND and OR. *Table 2-3* provides a configuration consideration checklist to help define your condition steps.

Table 2-3. Condition Workflow Checklist

Complete	Condition Step Check Item	Example
	What is the name of this workflow step?	■ AND ■ OR
	What is the status of the request at this workflow step?	■ On Hold ■ New ■ In Review
	What are the transitions from this workflow step?	■ Succeeded ■ Failed
	Who owns this workflow step?	■ Security Group ■ User ■ Standard Token
	What event triggers the notification?	■ When the process reaches the workflow step. ■ When a specific result is reached.
	Who or how many receive the notification?	■ Email Address (group alias) ■ Security Group
	What is the notification message.	■ Test complete ■ Approval required
	Use this workflow step as a timeout? How long?	■ 1 day ■ 2 days
	Are you using segregation of duties?	■ Based on owner of the workflow? ■ Based on workflow step?

Gathering Information for Subworkflow Steps

Subworkflow steps represent multiple workflows steps that follow a consistent pattern, such as code rework or unit testing. *Table 2-4* provides a configuration consideration checklist to help define your subworkflow steps. See Appendix A, *Worksheets,* on page 267 for a complete list of subworkflow step considerations.

Table 2-4. Subworkflow Workflow Checklist

Complete	Subworkflow Step Check Item	Example
	Is an existing workflow available as a subworkflow?	■ Yes ■ No
	What is the name of this subworkflow?	■ QA Test Cycle ■ QA Review Cycle
	What are the transitions from this workflow step?	■ Succeeded ■ Failed
	Who owns this workflow step?	■ Security Group ■ User
	What event triggers the notification?	■ When the process reaches the workflow step ■ When a specific result is reached
	Who or how many receive the notification?	■ Email Address (group alias) ■ Security Group
	What is the notification message.	■ QA Test Cycle Succeeded. ■ QA Test Cycle Failed.
	Use this workflow step as a timeout? How long?	■ 1 day ■ 2 days
	Are you using segregation of duties?	■ Based on owner of the workflow? ■ Based on workflow step?

General Workflow Design Guidelines

Table 2-5 provides a workflow logical guideline checklist that you can use to configure your deployment workflow.

Table 2-5. Workflow logical guidelines (page 1 of 2)

Complete	Guideline	Reason
	Workflows	
	Make one or more workflows available to process the packages.	Each workflow is assigned one workflow scope. The possible workflow scopes are: ■ Request (Demand Management) ■ Packages (Deployment Management) ■ Release Distributions (Deployment Management)
	Beginning and Closing Steps	
	Workflow must have a beginning step.	No processing can be done if the workflow has no beginning step.
	Workflow must have at least one step.	No processing can be done if the workflow has no steps.
	Workflow must have at least one Close step.	The package line cannot be closed without a Close step in the workflow.
	First workflow step cannot be a condition.	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.	The package or request will not close if a transition exists on 'Success.'
	Close step in subworkflow closes entire package line or request.	No no include a Close step in a subworkflow unless you want to close the workflow in the subworkflow.
	All Steps	
	All steps must be enabled.	Disabled steps cannot be used by the workflow and the process stops.
	Each enabled workflow step must have at least one incoming transition (except the beginning step).	It is not possible to flow to a workflow step without an incoming transition.
	Transition value is not a valid validation value (error).	The validation value has changed since the transition has been made.

Table 2-5. Workflow logical guidelines (page 2 of 2)

Complete	Guideline	Reason
	‘Other Values’ and ‘All Values’ transitions must not exist at the same step.	‘Other Values’ transition is always ignored if an ‘All Values’ transition exists.
	Each workflow step must have at least one outbound transition.	The branch of the workflow stops indefinitely without closing the package line or request.
	Each value from a list-validated validation must have an outbound transition.	Some validation values do not have transitions defined.
	Step with text or numeric validation must have an ‘Other Values’ or ‘All Values’ transition.	Because text and numeric Validations are not limited, an ‘Other Values’ or ‘All Values’ transition should be defined.
	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 defined on the **Security** tab.	No one is authorized to act on the step without a security group.
	Execution Steps	
	Each manual execution step must have at least one security group, user or token defined on the **Security** tab.	No one is authorized to act on the step without a security group.
	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.
	AND or OR condition step must have at least two incoming transitions.	An AND or OR condition with only one incoming transition will always immediately be true and have no effect.
	Subworkflows	
	Subworkflows must have at least one Return step.	Must include a Return step.
	Top-level workflow must not have a Return step.	Only subworkflows have a Return step.

Defining Object Types

Object types define the technical steps required to deploy application changes for packaged applications, custom applications, legacy systems, and web content. Many types of objects are deployed through a workflow such as files, SQL scripts, and data. Each object requires different information for proper processing. The information required to properly process an object are defined by the object type. For example, the object type for a file migration requires the following:

- The name of the file
- The type of file
- The location of the file

Additional information such as file compilation after file migration can also be specified on by the object type. A single workflow can process many different object types, as the workflow specifies the process and the environments. The package specifies each object and the associated object type as a package line in the package. Information contained on a package line (defined in the object type) works in conjunction with the workflow process to ensure that the object is correctly deployed.

For each object deployed through the process, you should collect the following information:

- The name of the object
- Object category (optional, used for reporting purposes)
- Parameters describing the object: what it is, where it is, its name, what needs to be done to it, and so on. This information will translate into object type fields. For each object type field, define the following:
    - Field name
    - Validation and component type (dictated by the validation)
    - Field behavior: whether it is displayed, required, any defaulting behavior, and so on.
- Commands required for the object being deployed. Object type commands often reference information stored in the parameters. These commands are executed at specific points (executions steps) in the workflow. For additional, you might shutdown a database before inserting new data into the database. For more information, see *Determining Commands Needed for Objects* on page 35.

For an example of conceptual workflow steps using different object types, see *Table 2-6*. See Chapter 5, *Configuring Object Types,* on page 139 for information on how to configure object types.

Table 2-6. Example workflow steps

Step Name	Workflow Step	Transition Values	Object Type Description
Migrate DEV to TEST	Decision	Approved Not Approved	Not applicable. Decision workflow step.
Stop the Server	Execution	Succeeded Failed	Connect to the server and stop the processes running on it.
Evaluate Object Type	Execution	Database File SQL Script	Evaluate the object type for each package line. Resolve the object type token.
Migrate to Database Environment	Execution	Succeeded Failed	Migrate the database changes to the TEST database environment. To do this, execute commands located in the object type.
Migrate to Server Environment	Execution	Succeeded Failed	Migrate the changes to the TEST server environment. To do this, execute commands located in the object type.
Sync	Condition	Success	Have all package lines enter this step before any continue to the next process step.
Compile Code	Execution	Succeeded Failed	Connect to the server and compile the code located on it.
Start the Server	Execution	Succeeded Failed	Connect to the server and start the processes on it.
Close FAILED	Execution	Succeeded	When a package line (object) enters this step, close the package.

Determining Commands Needed for Objects

When defining a deployment process, consider what commands need to be executed to achieve the desired results. Commands control which steps must be executed for each workflow step for the deployment process to execute properly. This can involve such things as migrating a file, executing a script, or compiling code.

At early stages of process development, it often helps to list the functional steps and desired results of the commands. It also helps to specify when in the deployment process these commands should be executed. It is then possible to use this information to build your commands adhering to HP Deployment Management's command standards, or to deliver these as design specifications for engineers in your group.

Collect the following information for each object type:

- The goal/purpose of the commands
- Functional steps within the commands
- When the commands should be run

For additional information on building commands, see the *Commands, Tokens, and Validations Guide and Reference*.

Object Type Checklist

Table 2-7 provides a configuration consideration checklist to help define your deployment system. See Appendix A, *Worksheets,* on page 267 for a complete list of object type considerations.

Table 2-7. Object type configuration checklist (page 1 of 2)

Complete	Object Type Check Item	Configuration Consideration
	Define an object type for each type of object to be deployed.	Include creating fields that describe the object and commands required to process it during deployment.
	Fields defined for the object type.	■ Fields are required to define the object. Make sure that the correct parameters describe the object to be deployed. ■ See Chapter 5, *Configuring Object Types,* on page 139. See the *Commands, Tokens, and Validations Guide and Reference*.
	Commands defined for the object type.	■ All commands needed to process and deploy the object have been constructed. ■ See Chapter 5, *Configuring Object Types,* on page 139. See the *Commands, Tokens, and Validations Guide and Reference*.
	Conditions set in commands for the object type.	■ Conditions to steps within the command that dictate when the specific command steps run have been added. See the *Commands, Tokens, and Validations Guide and Reference*.

Table 2-7. Object type configuration checklist (page 2 of 2)

Complete	Object Type Check Item	Configuration Consideration
	Is the object type enabled?	Disabled object types cannot be used.
	Object type and workflow	The following items should be coordinated between the workflow and object type: ■ Decide which workflow steps will execute the object type commands. ■ Decide which object type commands to run at specific workflow steps (using command conditions) ■ Workflow step source validations and object type field validations are in agreement. This is required when transitioning based on a field value (using token, SQL or PL/SQL execution types) ■ Allow the object type use for the workflow (set in the workflow window - **Deployment Management Settings** tab).
	Object types and environments	Specify any environment overrides must correspond to the object type commands.

Defining Environments

An environment can be a server, a single database instance, or an associated remote client machine. Deployment workflows require the source and destination environments as part of the execution workflow steps. Each environment must be carefully configured to ensure that passwords, communication protocols, and transfer protocols are configure properly. These configured environments can then be used in the deployment process.

➤ If there are multiple applications on a single environment, you can use the application codes feature in the environment definition.

Example environments settings illustrates the environments for several steps in a workflow. For information concerning the configuration of environments, see Chapter 7, *Configuring Environments,* on page 193.

Table 2-8. Example environments settings

Workflow Execution Step	Source Environment	Destination Environment
Stop the Server	DEV Server	TEST Server
Migrate to Server Environment	DEV Server	TEST Server
Compile Code	DEV Server	TEST Server
Start the Server	DEV Server	TEST Server
Migrate to Database Environment	DEV Database	TEST Database

Environments Checklist

Table 2-9 provides a configuration consideration checklist to help define your environments.

Table 2-9. Environments checklist (page 1 of 2)

Complete	Environments Check Item	Configuration Consideration
	Is the source environment defined	■ Is the source environment a server or a client? ■ Does the source environment include a database?
	Is the destination environment defined	■ Is the destination environment a server or a client? ■ Does the destination environment include a database?
	What is the connection protocol?	Is the connection protocol telnet, SSH, SSH2?
	What is the transfer protocol?	Is the transfer protocol FTP, FTP (active), FTP (passive), Secure Copy, Secure Copy 2?
	Is the environment enabled?	Disabled environments cannot be used.
	Is there an environment group?	■ What is the environment group name. ■ What is the source environment? ■ Is the execution order serial or parallel?
	Is the environment group enabled?	Disabled environment groups cannot be used.
	Is there an environment refresh?	■ What is the environment refresh name? ■ What is the source environment? ■ What is the destination environment?

Table 2-9. Environments checklist (page 2 of 2)

Complete	Environments Check Item	Configuration Consideration
	Is the environment refresh enabled?	Disabled environment refreshes cannot be used.
	Environments and workflow considerations.	Specify the source and destination environments (or environment groups) on the workflow execution steps.
	Environments and object type considerations.	Specify any environment overrides in the object type commands.

Defining Notification Templates

Notification templates are preconfigured email forms that can be used to quickly construct the body of a message. Notification templates are used by entities of HP Deployment Management, such as workflows and packages. When configuring a workflow, select the notification template you want to use for each workflow step. HP Deployment Management comes with a set of standard notification templates. You can use these existing templates or modify these templates or create new notification templates for your business process. Chapter 8, *Configuring Notification Templates,* on page 235 provides detailed information concerning the configuration of notification templates.

Notification Templates Checklist

Table 2-10 provides a configuration consideration checklist to help define your notification templates.

Table 2-10. Notification template checklist

Complete	Notification Template Check Item	Configuration Consideration
	Is the notification template enabled?	Disabled notification template cannot be used.
	Notification template and security group considerations.	Set ownership groups for these entities. Members of the ownership group (determined by associating security groups) are the only users who can edit the entities.

Defining User Data Fields

HP Deployment Management entities such as packages and workflows include a set of standard fields that provide information about those entities. While these fields are normally sufficient for day-to-day processing, you can add "user data" fields to capture additional information specific to your business process. Chapter 9, *Configuring User Data,* on page 247 provides detailed information concerning the configuration of user data fields.

User Data Checklist

Table 2-11 provides a configuration consideration checklist to help define your user data fields.

Table 2-11. User data checklist

Complete	User Data Check Item	Configuration Consideration
	Are the user data fields enabled?	Disabled user data fields cannot be used.
	User data and security group considerations	Set ownership groups for these entities. Members of the ownership group (determined by associating security groups) are the only users who can edit the user data fields.

Defining Security and Access

Included as part of a deployment process are the permissions required to perform various decisions or executions. Two of the ways in which PPM Center controls access to perform decisions and executions are by the following:

- **Licenses.** Licenses provide users with access to a PPM Center products such as HP Deployment Management, but licenses do not dictate what actions a user is authorized to perform.
- **Access Grants.** Access grants, when used with licenses, define the actions a user is authorized to perform within a PPM Center product.

For example, you can restrict a user's actions as follows:

- License, Deployment Management
  - Access Grant, View Packages - Those who can view packages
  - Access Grant, Edit Packages - Those who can edit packages

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

When designing deployment processes, use security groups or dynamic access (tokens). Avoid specifying a list of users to control an action. If the list of users changes (due to a departmental reorganization), you must update your workflow in a variety of places to keep the deployment process running correctly. By using a security group instead of a list of users, you can update the security group once, and the changes are propagated throughout the workflow.

Table 2-12 provides an example of which security groups can access a deployment workflow and at which workflow step.

Table 2-12. Example of workflow security groups

Workflow Step	Security Groups
Create a package	Financial Apps - Engineer Financial Apps - Database Financial Apps - Manage Deployment
Use the deployment workflow	Financial Apps - Engineer Financial Apps - Manage Deployment
Use a file object type	Financial Apps - Engineer Financial Apps - Database Financial Apps - Manage Deployment
Use a database object type	Financial Apps - Database Financial Apps - Manage Deployment

Security and User Access Checklist

Table 2-13 provides a configuration consideration checklist to help define your security and user access requirements.

Table 2-13. Security and user access checklist (page 1 of 2)

Complete	Security and User Access Check Item	Configuration Consideration
	Created security groups for access to screens and functions	Security groups to be used to grant access to certain screens and functions have been created.
	Created security groups for association with workflow steps	Security groups to allow users to act on a specific workflow step have been created.
	Set security on package creation	All available options for restricting who can create and submit packages have been set.
	Set security on package processing	All available options for restricting who can process packages have been set.
	Set security on deployment system configuration	You have specified who can modify the deployment process. This includes editing the workflow, object type, environment, security groups, and so on.

Table 2-13. Security and user access checklist (page 2 of 2)

Complete	Security and User Access Check Item	Configuration Consideration
	Security group and workflow considerations	■ Associate security groups with workflow steps. Users in the included groups can act on the step. ■ Set workflow and workflow step ownership.
	Security group and object type considerations	Set ownership groups for object types. Members of the ownership group (determined by associating security groups) are the only users who can edit the object type.
	Security group and environments considerations	Set ownership groups for environments. Members of the ownership group (determined by associating security groups) are the only users who can edit the environments.
	Security group and notification template considerations	Set ownership groups for notification templates. Members of the ownership group (determined by associating security groups) are the only users who can edit the notification templates.
	Security group and user data considerations	Set ownership groups for user data. Members of the ownership group (determined by associating security groups) are the only users who can edit user data.

3 Configuring Workflows

In This Chapter:

Overview of Workflows

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

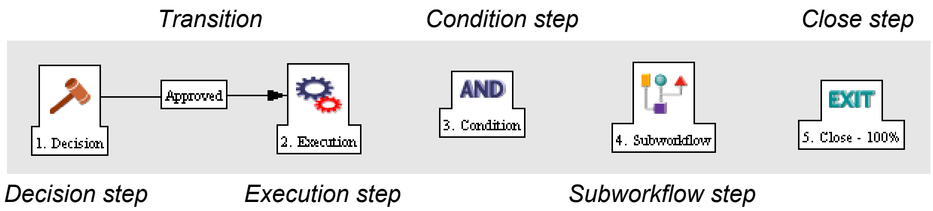
The following is a list of the basic components of a workflow:

- **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 following lists the basic workflow steps:
  - **Decision step.** Decision steps represent manual activities performed outside of PPM Center. For example, a decision step is where a user or group of users approves a request.
  - **Execution step.** Execution steps represent actions that are automated through PPM Center. Example: updating a Web page with the results of a test.
  - **Condition step.** Condition steps are logic steps used for complex workflow processing, such as allowing the workflow to proceed only when each of the workflow steps are completed.
  - **Subworkflows step.** 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. For example, the result of a decision step is either Approved or Not Approved.
- **Workflow step security.** Workflow step security determines who has permission to execute or choose a result for a workflow step. For example, for an Approve Request decision step, only the IT project manager can approve or deny the request.

- **Notification.** Notifications are emails alerts sent out at specific workflow steps. For example, for an Approve Request decision step, an email alert is sent to the product manager.
- **Close step.** Close steps indicate the end of the workflow. The close step is an execution step that marks the request as completed.

Figure 3-1 illustrates the basic workflow components in a workflow.

Figure 3-1. Workflow components



Mapping Workflows

Mapping all of the individual workflow steps into a single workflow involves the following two stages:

Stage 1. Create a block diagram. Map each Workflow Step Worksheet as one block in the diagram. On the block diagram include transitions, workflow step security, and notifications (see *Figure 3-2*).

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

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

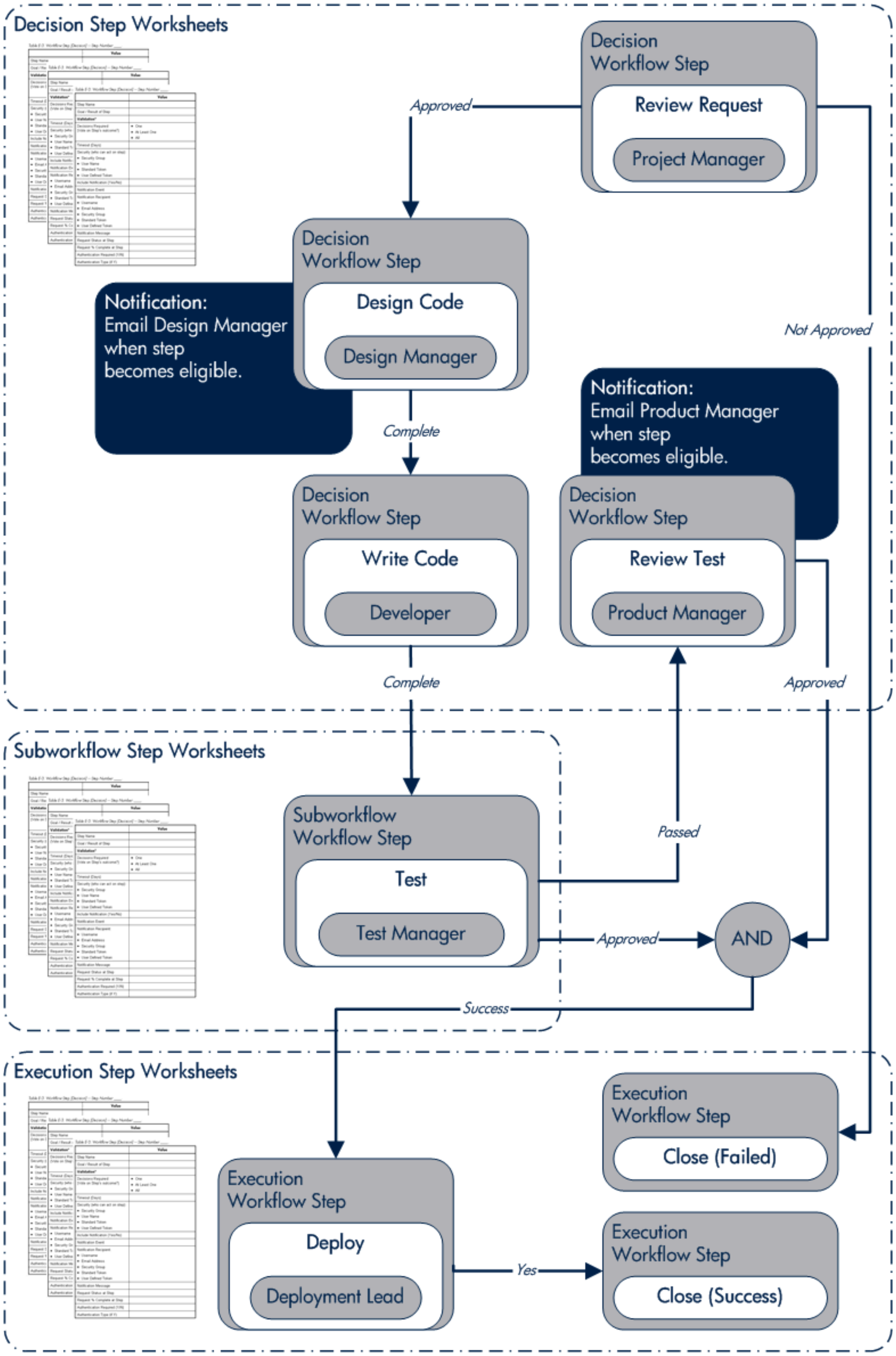
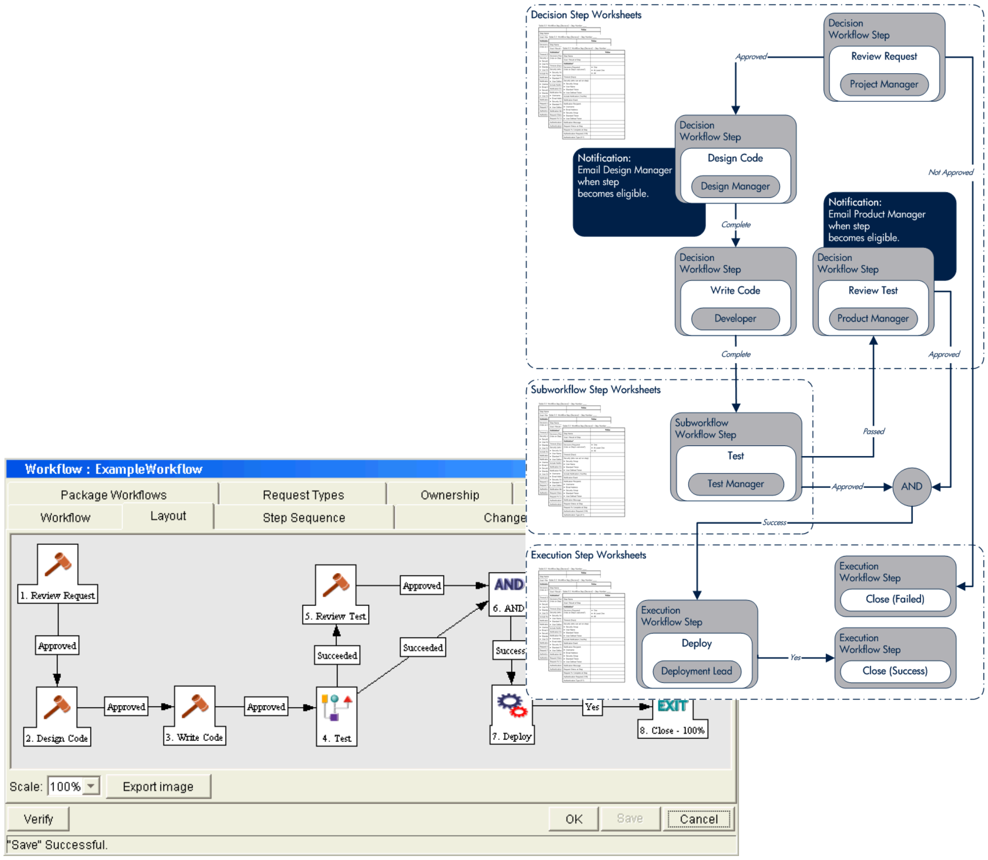


Figure 3-3. Stage 2. Create the workflow



Opening the Workflow Workbench

To open the Workflow Workbench:

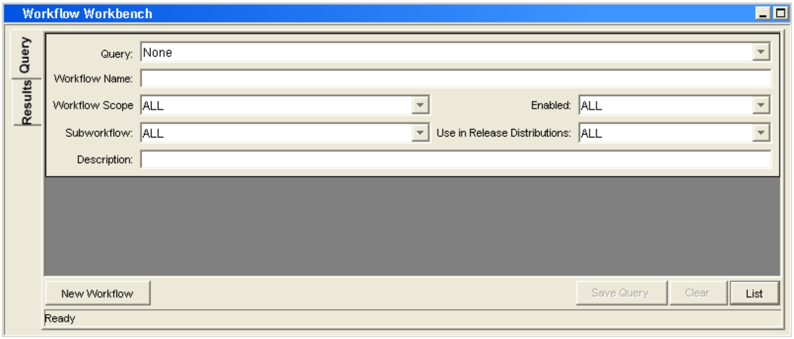
1. Log on to PPM Center.

2. From the menu bar, select **Administration > Open Workbench.**

   The PPM Workbench opens.

3 . From the shortcut bar, select **Configuration > Workflows.**

The Workflow Workbench opens.



Query: None
Workflow Name:
Workflow Scope: ALL
Enabled: ALL
Subworkflow: ALL
Use in Release Distributions: ALL
Description:

Creating Workflows

To start a new workflow, you must know how to use the Workflow Workbench. This section covers the basics on how to create a workflow.

Configuring General Information for a Workflow

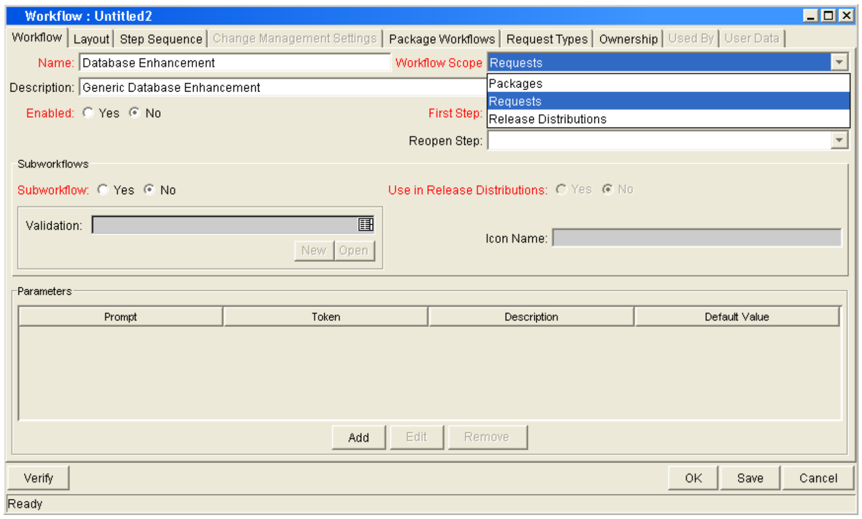
To enter basic workflow information:

1. From the Workbench shortcut bar, select **Configuration > Workflows.**

   The Workflow Workbench opens.

2. Click **New Workflow.**

   The Workflow window opens.



Name: Database Enhancement
Workflow Scope: Requests
Packages
Requests
Release Distributions
Description: Generic Database Enhancement
Enabled: ○ Yes ● No
First Step:
Reopen Step:
Subworkflow: ○ Yes ● No
Use in Release Distributions: ○ Yes ● No
Validation:
New Open
Icon Name:

Prompt	Token	Description	Default Value

3. In the **Name** field, type a name for the workflow.
4. In the **Workflow Scope** field, select:
   - For HP Deployment Management packages, select **Packages.**
   - For HP Deployment Management releases and distributions, select **Release Distributions.**
5. Click **Save.**
6. Click **OK.**

Dragging and Dropping Workflow Steps

A library of existing workflow steps resides in the Workflow Step Source window. The Workflow Step Source window includes a **Filter by** field, lets you see only the workflow steps that you can use.

Workflow steps are assembled into workflows on the **Layout** tab of the Workflow window. Drag a workflow step from the Workflow Step Sources window and drop it onto the **Layout** tab.

The Workflow Step window opens.

Use the Workflow Step window to configure the following:

- General information about the workflow step
- Workflow step security
- Notifications for the workflow step
- Timeouts for the workflow step

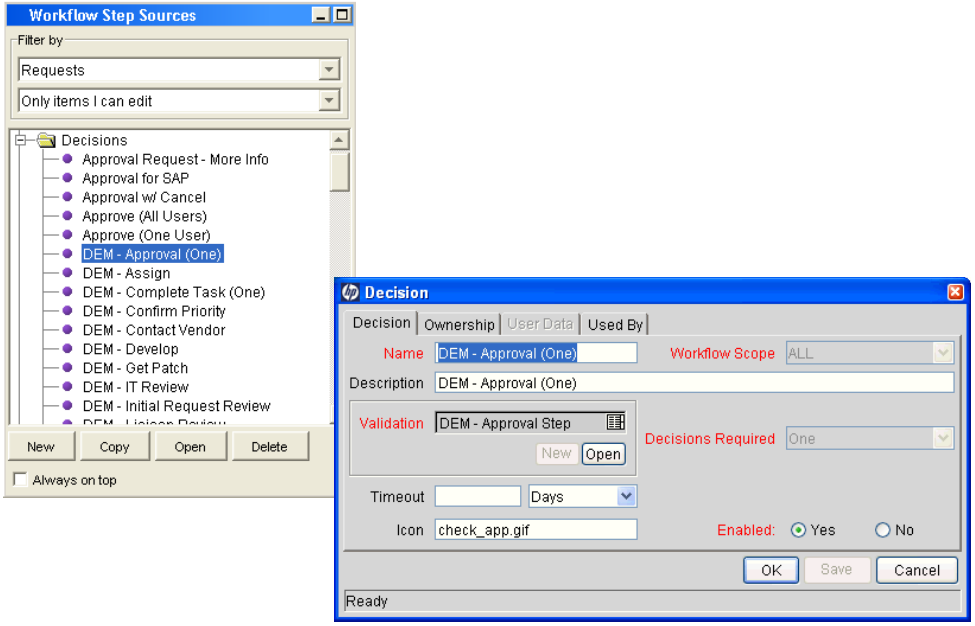
Choosing Workflow Steps

PPM Center comes with many predefined workflow steps. These workflow steps are located in the Workflow Step Source window. Workflow steps in the Workflow Step Source window are filtered using the **Filter by** field. The Workflow Step Source window contains the following folders:

- Decision
- Conditions
- Executions
- Subworkflows

To evaluate a workflow step, determine which of the these workflow folders it corresponds to. Open the Workflow Step Source folder, and then open the workflow steps that most closely suit your needs (*Figure 3-4*).

Figure 3-4. Workflow step source



Workflow Step Sources

Filter by: Requests; Only items I can edit

- Decisions
  - Approval Request - More Info
  - Approval for SAP
  - Approval w/ Cancel
  - Approve (All Users)
  - Approve (One User)
  - DEM - Approval (One)
  - DEM - Assign
  - DEM - Complete Task (One)
  - DEM - Confirm Priority
  - DEM - Contact Vendor
  - DEM - Develop
  - DEM - Get Patch
  - DEM - IT Review
  - DEM - Initial Request Review

□ Always on top

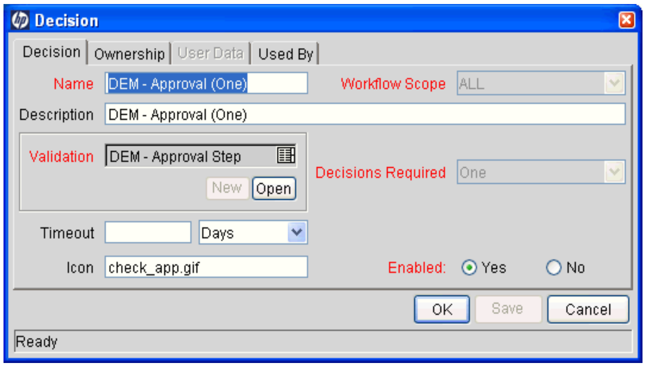
Decision

Name: DEM - Approval (One)
Workflow Scope: ALL
Description: DEM - Approval (One)
Validation: DEM - Approval Step
Decisions Required: One
Timeout: Days
Icon: check_app.gif
Enabled: ● Yes ○ No

Ready

The validation values are the acceptable values a workflow step can have (see *Configuring Validations for Workflow Steps* on page 90). Check these to see if they meet your transition requirements.

Figure 3-5. Workflow step source validation



Decision

Name: DEM - Approval (One)
Workflow Scope: ALL
Description: DEM - Approval (One)
Validation: DEM - Approval Step
Decisions Required: One
Timeout: Days
Icon: check_app.gif
Enabled: ● Yes ○ No

Ready

Overview of Decisions Workflow Steps

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

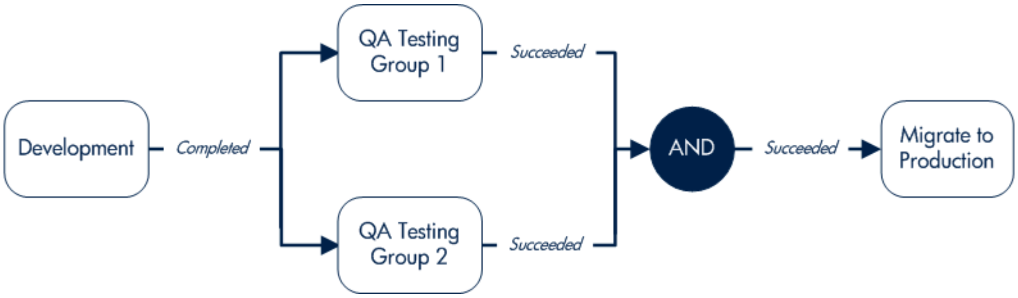
- Decisions made by committees
- Code designs and reviews

Overview of 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:

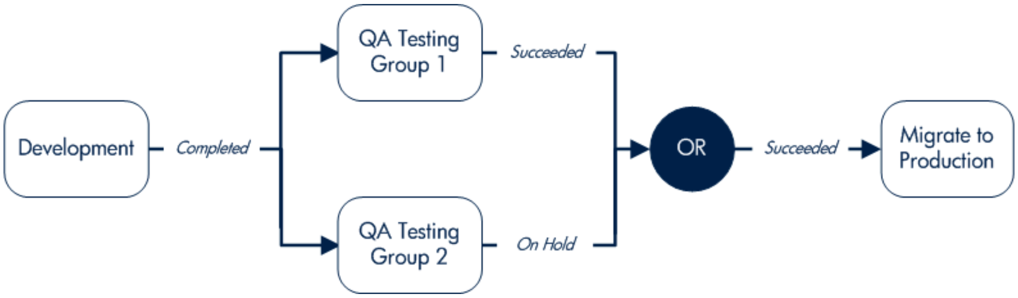
- **AND.** This condition is met only after all workflow steps leading to it reach the specified required status. *Figure 3-6* illustrates an example of the AND condition workflow step.

Figure 3-6. AND example



- **OR.** This condition is met if at least one of the workflow steps leading to it reaches the required status specified for it *Figure 3-7* illustrates an example of the OR condition workflow step.

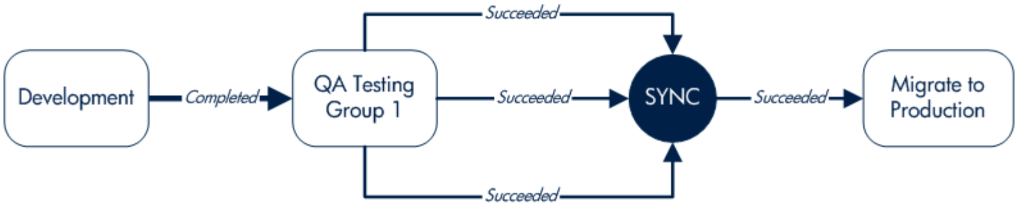
Figure 3-7. OR example



- **SYNC.** A SYNC workflow condition step is successful only if all the package lines of that package reach the status required for the workflow step immediately preceding that SYNC step.

  Consider the business process illustrated in *Figure 3-8*. According to the flow chart, after QA Testing Group 1 succeeds for all package lines, SYNC succeeds, and then the next step, Migrate to Prod, becomes eligible.

  Figure 3-8. SYNC example

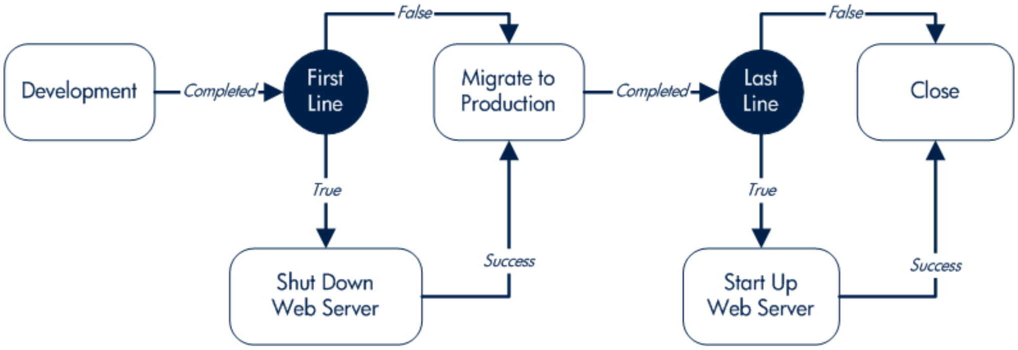
- **FIRST LINE and LAST LINE.** For FIRST LINE, only the first line to reach the condition workflow step takes the True transition. All subsequent lines take the False transition.

  For LAST LINE, only the last active line to reach the condition workflow step takes the True transition. All previous lines take the False transition.

  The business process illustrated in *Figure 3-9* could be part of a Web site maintenance life cycle. As part of this life cycle, three HTML files are processed on three respective package lines in a single package. The Web site updates are large enough to warrant shutting down the Web server during change migration.

  By including a FIRST LINE step, only the first line causes the server to shut down. The server remains down while the rest of the changes are migrated to production. By including a LAST LINE workflow step, the server remains down until the last active line reaches the condition step. The last active line takes the True transition, the Web server starts up, and the maintenance is complete.

Figure 3-9. FIRST LINE and LAST LINE example



Overview of Execution Workflow Steps

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

- Run object type commands
- Run workflow step commands
- Close the workflow (Close workflow step)

Overview of Subworkflow Workflow Steps

A subworkflow step represents multiple workflow steps (the subworkflow) within a workflow. 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.

Adding Close Workflow Steps

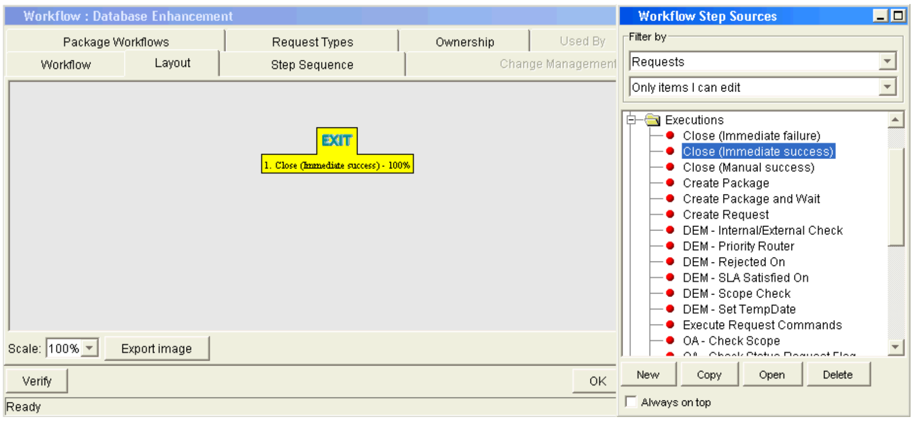
Regardless of how long or short it is, every workflow must include a close workflow step (see *Figure 3-10*). A close workflow step is a type of execution workflow step. You can find it in the Executions folder in the Workflow Step Sources window.

The three close workflow steps are as follows:

- **Close (Immediate Success).** This close workflow step immediately completes a request or package with a status of Success.
- **Close (Manual Success).** This close workflow step requires manual intervention to complete a request or package and set the request or package status to Success.
- **Close (Immediate Failure).** This close workflow 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 workflow step (*Figure 3-10*).

Figure 3-10. Close workflow step



Filter by

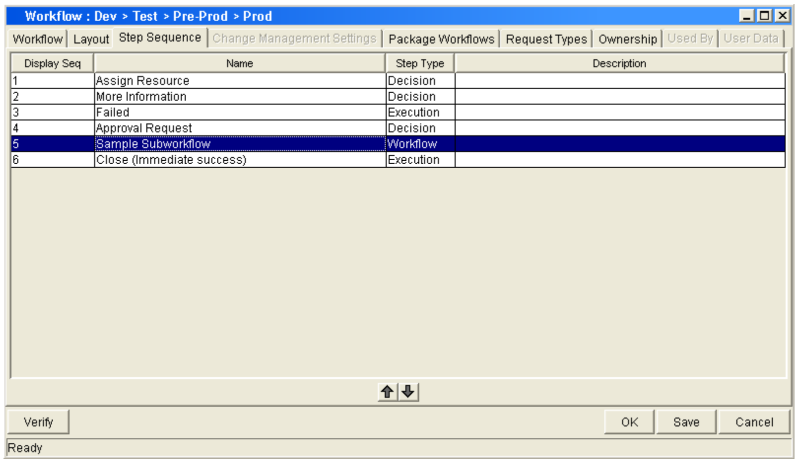
Requests

Only items I can edit

Adjusting Workflow Step Sequences

After you have assembled all of the workflow steps on **Layout** tab, you can adjust their sequence. In the Workflow window, click the **Step Sequence** tab. The **Step Sequence** tab lists all of the workflow steps. Select a workflow step, and click the arrow pointers at the bottom of the tab to move the selected workflow step up or down.

Figure 3-11. Step sequence tab

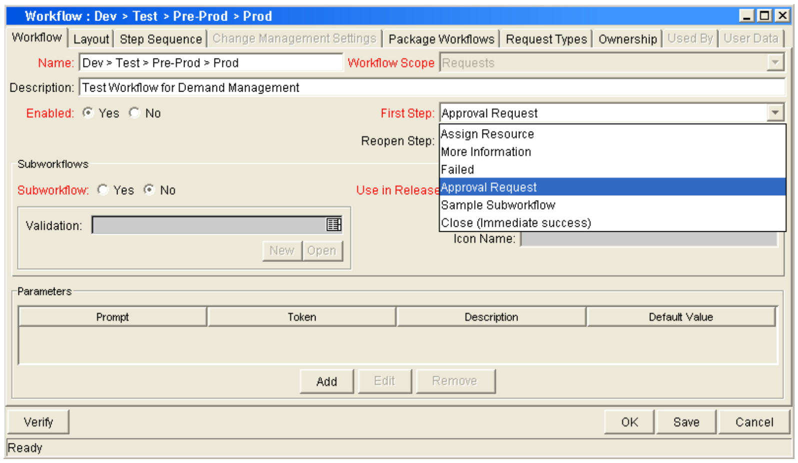


Display Seq	Name	Step Type	Description
1	Assign Resource	Decision	
2	More Information	Decision	
3	Failed	Execution	
4	Approval Request	Decision	
5	Sample Subworkflow	Workflow	
6	Close (Immediate success)	Execution	

Specifying the First Step

After you assemble all of the workflow steps in the correct sequence, specify the first step in the workflow process. To specify the first step, in the Workflow Workbench, click the **Workflow** tab (see *Figure 3-12*). In the **First Step** field, select the first step.

Figure 3-12. Workflow tab



Name: Dev > Test > Pre-Prod > Prod
Workflow Scope: Requests
Description: Test Workflow for Demand Management
Enabled: ● Yes ○ No
First Step: Approval Request
Reopen Step:
Subworkflow: ○ Yes ● No
Use in Release
Icon Name:

Verifying and Enabling Workflows

Verifying and enabling a workflow are the last steps required to make a workflow available. Verify a workflow checks to make sure the logic of the workflow is correct. Enabling a workflow makes the workflow available for use.

To verify a workflow:

1. From the Workbench shortcut bar, select **Configuration > Workflows.**

   The Workflow Workbench opens.

2. Open a workflow.

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

3. Click **Verify.**

   The logic of the workflow is checked and a status window is returned.

To enable a workflow:

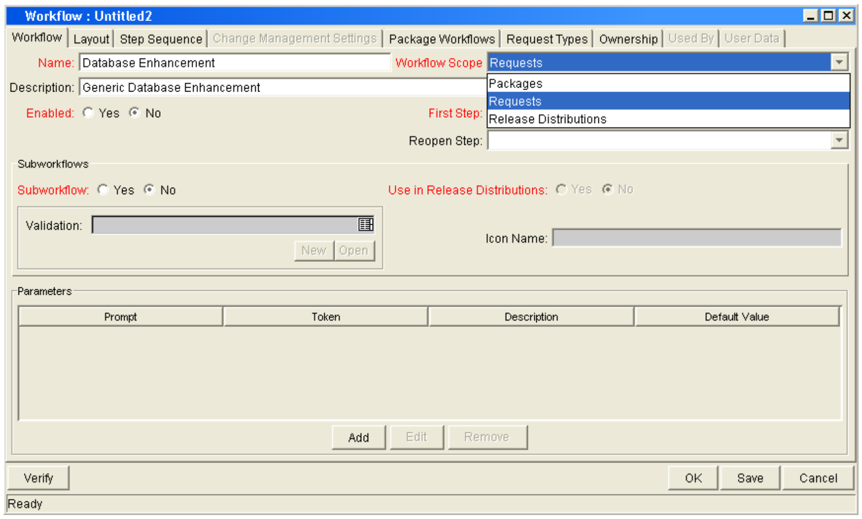
1. From the Workbench shortcut bar, select **Configuration > Workflows.**

   The Workflow Workbench opens.

2. Open a workflow.

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

3. For Enabled, click **Yes.**



Name: Database Enhancement

Workflow Scope: Requests

Description: Generic Database Enhancement

Enabled: ○ Yes ● No

First Step:

Reopen Step:

Subworkflows

Subworkflow: ○ Yes ● No

Use in Release Distributions: ○ Yes ● No

Validation:

Icon Name:

Parameters

Prompt	Token	Description	Default Value

Ready

4. Click **Save.**

Configuring Workflow Steps

Every time you drag a workflow step from the Workflow Step Source window to the **Layout** tab in the Workflow window, a Workflow Step window opens. You can enter none, some, or all of the known information at the initial window opening, or you can open the Workflow Step window later in the workflow design process.

Information that you enter in the Workflow Step window can be gathered from the corresponding Workflow Step Worksheets. The Workflow Step window contains the following tabs:

- **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.** This tab configures workflow steps to take into account segregation of duties, excluding the participants for a workflow step from participating in a different workflow step.

Configuring General Information for Workflow Steps

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

To add general information to a workflow step:

1. From the Workbench shortcut bar, select **Configuration > Workflows.**

   The Workflow Workbench opens.

2. Open a workflow.

   The Workflow window opens.

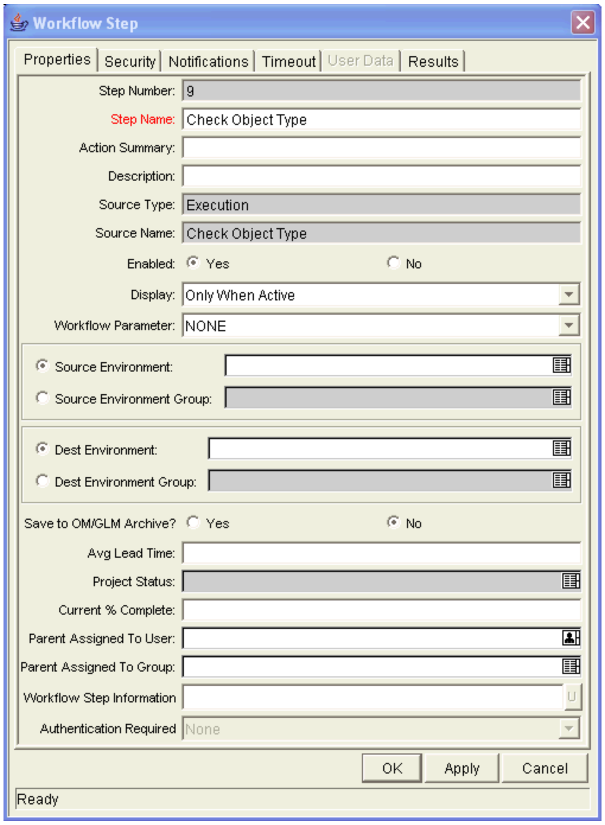
3. Click the **Layout** tab.
4. Right-click a workflow step.

   A menu window opens.

5. Click **Edit.**

   The Workflow Step window opens.

6. Complete the fields on the **Properties** tab.



Step Number: 9

Step Name: Check Object Type

Action Summary:

Description:

Source Type: Execution

Source Name: Check Object Type

Enabled: ● Yes ○ No

Display: Only When Active

Workflow Parameter: NONE

● Source Environment:

○ Source Environment Group:

● Dest Environment:

○ Dest Environment Group:

Save to OM/GLM Archive? ○ Yes ● No

Avg Lead Time:

Project Status:

Current % Complete:

Parent Assigned To User:

Parent Assigned To Group:

Workflow Step Information

Authentication Required None

Ready

7. Click **Save.**

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 Workbench shortcut bar, select **Configuration > Workflows.**

   The Workflow Workbench opens.

2. Open a workflow.

   The Workflow window opens.

3. Click the **Layout** tab.

4. Right-click a workflow step.

   A menu window opens.

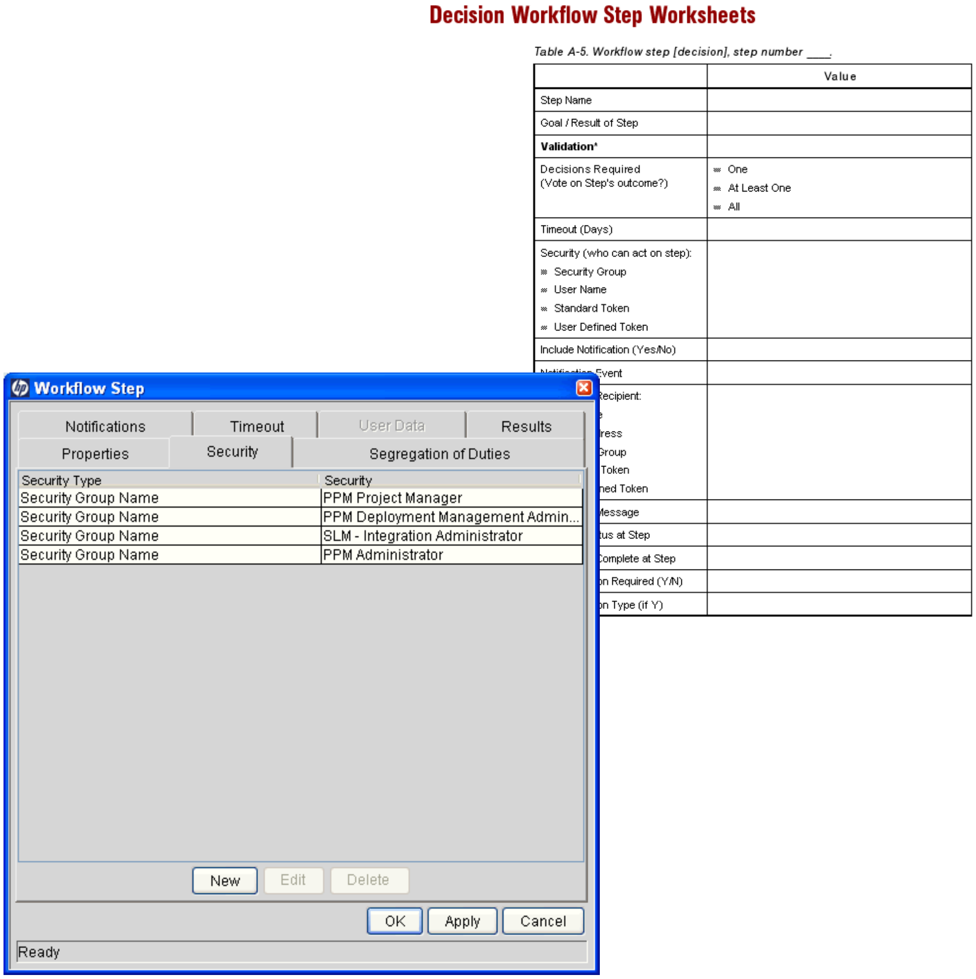
5. Click **Edit.**

   The Workflow Step window opens.

6. Click the **Security** tab.

7. Click **New.**

   The Workflow Step Security window opens.



Decision Workflow Step Worksheets

Table A-5. Workflow step [decision], step number ____.

	Value
Step Name	
Goal / Result of Step	
Validation*	
Decisions Required (Vote on Step's outcome?)	One At Least One All
Timeout (Days)	
Security (who can act on step): Security Group User Name Standard Token User Defined Token	
Include Notification (Yes/No)	
Notification Event	
Recipient: e ress Group Token ned Token	
Message	
tus at Step	
Complete at Step	
on Required (Y/N)	
on Type (if Y)	

Workflow Step

Security Type	Security
Security Group Name	PPM Project Manager
Security Group Name	PPM Deployment Management Admin...
Security Group Name	SLM - Integration Administrator
Security Group Name	PPM Administrator

8. Select one of the following security types:

   - **Enter a Security Group Name.** Select a security group to act on the workflow step. Selecting a security group changes the name of the auto-complete to Security Group. The security type changes to Security Group.

   - **Enter a Username.** Select a user to act on the workflow step. The auto-complete name changes to Username. The security type changes to Username.

   - **Enter a Standard Token.** Select a standard token to act on the workflow step. The auto-complete name changes 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.

   - **Enter a User Defined Token.** Select a user-defined token to act on the workflow step. This changes the auto-complete name to **User Defined Token.** The security type changes to a list, and the **Tokens** button is enabled. To open the Token Builder window and select a token, click **Tokens.** In the list, select one of the following:

     - **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.

9. Click **OK.**

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

    The Workflow Step window closes.

11. From the **Security** tab, click **OK.**

    The changes are added to the workflow.

Configuring Dynamic Security for Workflow Steps

Workflow steps can also be configured so that its security is determined at runtime based on information entered in the request or package.

To configure a workflow step with dynamic security:

1. Open the Workflow Workbench.

   The Workflow Workbench opens.

2. Open a workflow.

   The Workflow window opens.

3. Click the **Layout** tab.

4. Right-click a workflow step.

   A menu window opens.

5. Click **Edit.**

   The Workflow Step window opens.

6. Click the **Security** tab.

7. Click **New.**

   The Workflow Step Security window opens.

8. Select one of the following security types:

   - **Enter a Security Group name.** Select a security group to act upon the workflow step. The auto-complete name changes to Security Group. The security type changes to Security Group.

   - **Enter a Username.** Select a user to act upon the workflow step. The auto-complete name changes to Username. The security type changes to Username.

   - **Enter a Standard Token.** Select a standard token to act upon the workflow step. The auto-complete name changes 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.

- **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 dynamically changes 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 drop-down 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.

9. In the Workflow Step Security window, click **OK.**
10. In the Workflow Step window, click **OK.**
11. From the **Security** tab of the Workflow Step window, click **OK.**

Configuring Notifications for Workflow Steps

Notifications can be sent when a workflow step becomes eligible or after a workflow step is complete. 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). Notifications are defined on the **Notifications** tab of the Workflow Step window.

Review the Workflow Step Worksheet for notification information.

To add a notification to a workflow step:

1. From the Workbench shortcut bar, select **Configuration > Workflows.**
2. Open a workflow.

   The Workflow window opens.
3. Click the **Layout** tab.
4. Right-click a workflow step.

   The workflow step is highlighted. A menu window opens.

5. Click **Edit.**

   The Workflow Step window opens.

6. Click the **Notifications** tab.

7. Click **New.**

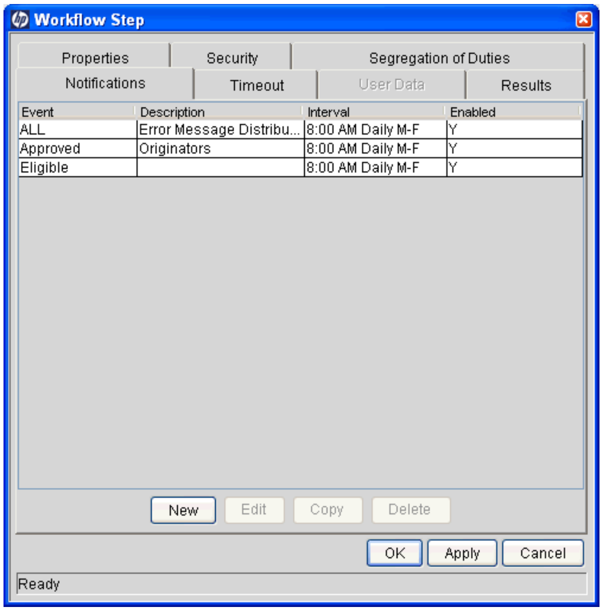
   The Add Notification for Step window opens.

8. From the **Setup** tab, configure the following:

   - When to send the notification (Event and Interval).
   - Who receives the notification (Recipients)

9. Click the **Message** tab.

10. Configure the body of the notification.



Event	Description	Interval	Enabled
ALL	Error Message Distribu...	8:00 AM Daily M-F	Y
Approved	Originators	8:00 AM Daily M-F	Y
Eligible		8:00 AM Daily M-F	Y

Ready

11. Click **OK.**

    The **Notifications** tab lists the new notification. You can send different notifications to different recipients for different events by clicking **New** and repeating this process. The following lists some of the reasons you might want to send different notifications for a single workflow step:

    - Send different notifications depending on the result of the step
    - Send different notifications depending on the type error
    - Send the notifications to a different set of users depending on the step's result or error
    - Specifying different intervals or reminders based on the type of step error

12. Click **OK.**

    The changes are added to the workflow.

Configuring Setup Tabs

You can configure a workflow step to send notifications at different times, different intervals, 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 64.

2. Click **New.**

   The Add Notification for Step window opens.

3. Click the **Setup** tab.

4. Configure the **Setup** tab as specified in the following table.

Field Name	Description	Notes
Event	Eligible	
Interval	Immediate	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. 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 will allow this batching to occur.
Send Reminder	Yes/No	(Optional) 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 All.
Enabled	Yes	

5. Click **OK.**

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

Sending Notifications when Workflow Steps have Specific Results

You can configure a notification to send when a workflow step has a specific decision or execution result. The value for these results is determined by the workflow step source's validation.

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 64.

2. Click **New.**

   The Add Notification for Step window opens.

3. Click the **Setup** tab.

4. Configure the **Setup** tab as listed in the following table.

Field Name	Description	Notes
Event	Specific Result	
Value	Select the value to trigger the Notification.	The list of values is determined by the workflow step source's validation. Therefore, this selection will always be limited to the possible results of the step.
Interval	Immediate	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's ready for approval in the morning. 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 will allow this batching to occur.
Send Reminder	Yes/No	(Optional) 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 All.
Enabled	Yes	

5. Click **OK.**

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

Sending Notifications When Workflow Steps Have Specific Errors

You can configure the notification to be sent when a workflow step has a specific error. *Table 3-1* lists the workflow step errors.

Table 3-1. Specific errors for workflow steps

Specific Error	Meaning
No consensus	When all users of all security groups, or users linked to the workflow step need to vote, and there is no consensus.
No recipients	When none of the security groups linked to the workflow step has users linked to it, no user can act on the workflow step.
Timeout	When 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 request is expecting a message from a package line that is cancelled or closed.
Child closed	For wf_receive or wf_jump steps, a package line is expecting a message from a request that is cancelled or closed.
No parent	For wf_receive or wf_jump steps, a request is expecting a message from a package line that has been deleted.
No child	For wf_receive or wf_jump steps, a package line is expecting 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	When the package level subworkflow receives multiple results from package lines that traversed through it.

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

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

   See *Configuring Notifications for Workflow Steps* on page 64.

2. Click **New.**

   The Add Notification for Step window opens.

3. Click the **Setup** tab.

4. Configure the **Setup** tab as follows:

Field Name	Description	Notes
Event	Specific Error	
Error	Select the value to trigger the Notification.	This is a standard set of errors. See *Sending Notifications When Workflow Steps Have Specific Errors* on page 69.
Interval	Immediate	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's ready for approval in the morning. 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 batching to occur.
Send Reminder	Yes/No	(Optional) 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 All.
Enabled	Yes	

5. Click **OK.**

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

Specifying the Time Notifications are Sent

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 64.

2. Click **New.**

   The Add Notification for Step window opens.

3. Click the **Setup** tab.

4. Configure the **Interval** field as follows:

   - **8:00 AM Daily M-F.** This notification is sent every 8:00 a.m.on the next available work day after the notification event occurs.

   - **Hourly M-F.** This notification is sent every hour, starting on the next available work day after the notification event occurs.

   - **Immediate.** This notification is sent immediately.

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 All.

To send follow-up notifications:

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

   See *Configuring Notifications for Workflow Steps* on page 64. The **Notifications** tab opens.

2. Click **New.**

   The Add Notification for Step window opens.

3. Click the **Setup** tab.

4. Configure the **Interval** field, as listed in the following table.

Field Name	Description	Notes
Event		Selects any value except for All.
Send Reminder	Yes	To enable the **Reminder Days** field, select **Yes.**
Reminder Days	Enter the number of days.	The number of days to wait before sending a reminder notification.

5. Click **OK.**

6. 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.

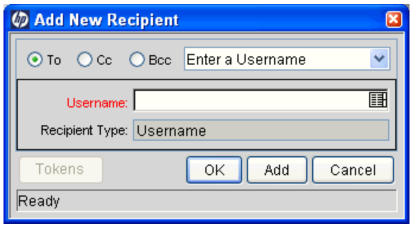
2. Click **New.**

   The Add Notification for Step window opens.

3. Click the **Setup** tab.

4. Click **New.**

   The Add New Recipient window opens.



Add New Recipient

● To ○ Cc ○ Bcc Enter a Username

Username:

Recipient Type: Username

Ready

5. In the list, select one of the following methods to use to specify the recipient(s):

   - **Enter a Security Group.** Select a specific security group, and all enabled users in the group with email addresses will receive the notification.

   - **Enter a Username.** Select a specific user to receive the notification. The user must have an email address.

   - **Enter an Email Address.** Enter any email address of the notification.

   - **Enter a Standard Token.** Select from a list of system tokens that corresponds to a user, security group, or email address.

   - **Enter a User Defined Token.** Enter any field token that corresponds to a user, security group, or email address.

   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. Enter the specific value that corresponds to the recipient type selected above.

   This can be a username, email address, security group, or a token.

   Use security groups or dynamic access (distributions) to define the notification recipients whenever possible. Avoid specifying a list of users or an individual user's email address. If the list of users changes (due to a departmental or company reorganization), that list would have to be updated manually. By using a security group instead of a list of users, the security group can be updated once, and the changes will be propagated throughout the workflow steps.

   Use distributions when sending a notification to an undetermined party. For example, the notification can be configured to be sent to the Assigned to User by specifying [REQ.ASSIGNED_TO_USERID] in the Add New Recipient window.

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 Tabs

It is possible to construct the notification's message to ensure that it contains the correct information or instructions 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's status. Additionally, the notification should include a link (URL) to the referenced request.

Notifications include the following features to make them easier to configure and use:

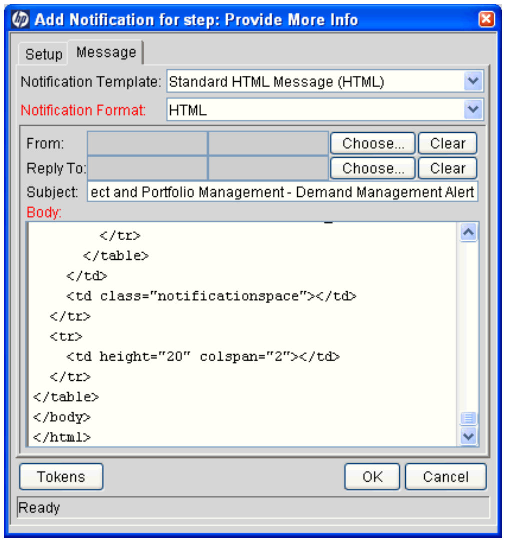
- Select from a number of preconfigured notification templates to more quickly construct the body of your message.
- The body of the notification can be plain text or HTML.
- Multiple tokens can be included in the notification. These tokens will resolve to information relevant to the recipient. For example, 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 in a notification:

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

   The Add Notification for Step window opens.

3. Click the **Message** tab.



Notification Template: Standard HTML Message (HTML)

Notification Format: HTML

From:

Reply To:

Subject: ect and Portfolio Management - Demand Management Alert

Body:

```
        </tr>
      </table>
    </td>
    <td class="notificationspace"></td>
  </tr>
  <tr>
    <td height="20" colspan="2"></td>
  </tr>
</table>
</body>
</html>
```

Ready

4. In the **Notification Template** field, you can select a template to use for the notification.

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

5. In the **Notification Format** field, select **HTML** or **Plain Text.**

   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.

6. Select values for the **From** and **Reply to** fields.

7. Construct the body of the message.

   When constructing the body, consider using the following:

   - Token for the URL to the Request Detail page. See *Table 3-2* on page 76 for a list of these tokens.

   - Token for the URL to the package (PPM Workbench or standard interface). See *Table 3-2* on page 76 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.

   - 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

It is possible to select any of the available tokens accessed through the Token Builder window to include in the body of your message. However, not all tokens will resolve in all situations. As a general rule, tokens associated with the request or workflow will resolve.

Including URLs (Smart URLs)

When you receive a notification, it is often helpful to have a link to the item needing attention. Notifications can be configured in the body of a notification to include the Web address (URL) for the following entities:

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

If you are viewing your email with a Web-based mail reader (such as Microsoft Outlook), you can click the URL in the notification and be taken 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.

The most commonly used smart URL tokens for packages and requests are described in *Table 3-2*.

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.

When using an HTML formatted message, an alternate token must be used to provide a link to requests. This token can also be used in plain-text formatted notifications. The smart URL token for requests is described in *Table 3-3*.

Table 3-3. Smart URL tokens in HTML format

Smart URL Token	Description
REQUEST_ID_LINK	Provides a link that loads the request detail page in the standard interface.

The token will resolve to the following format:

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

In the notification, the link would appear 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 for information on how to set the timeout.

To set timeouts for a workflow step:

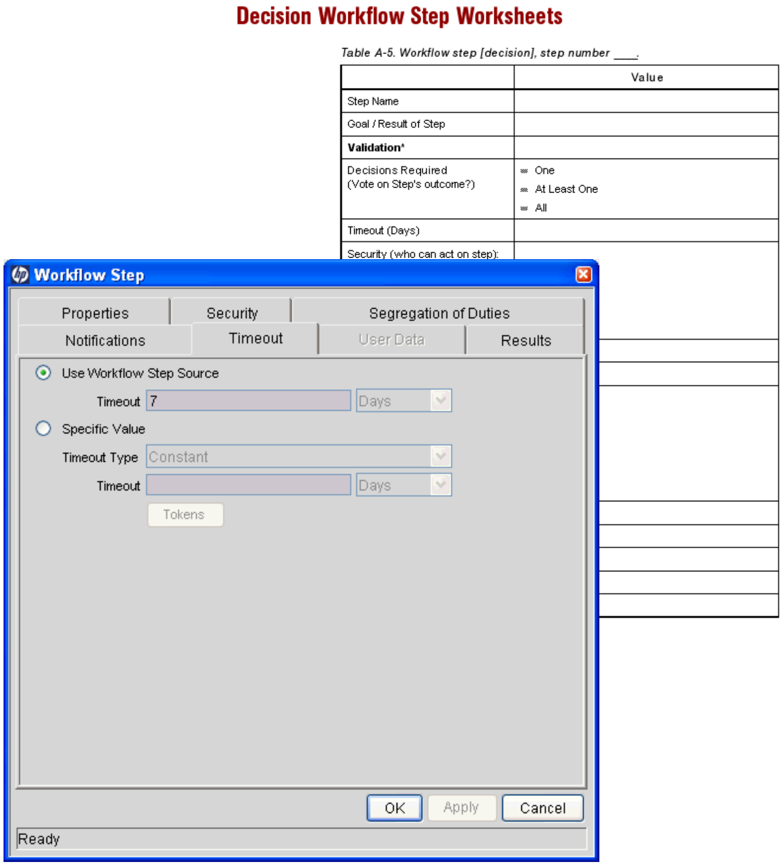
1. From the Workbench shortcut bar, select **Configuration > Workflows.**
2. Open a workflow.

   The Workflow window opens.
3. Click the **Layout** tab.
4. Right-click a workflow step.

   The workflow step is highlighted. A menu window opens.
5. Click **Edit.**

   The Workflow Step window opens.

6. Click the **Timeout** tab.



Decision Workflow Step Worksheets

Table A-5. Workflow step [decision], step number ____.

	Value
Step Name	
Goal / Result of Step	
Validation*	
Decisions Required (Vote on Step's outcome?)	○ One ○ At Least One ○ All
Timeout (Days)	
Security (who can act on step):	

Workflow Step

● Use Workflow Step Source

Timeout: 7 Days

○ Specific Value

Timeout Type: Constant

Timeout: Days

Ready

7. Configure the timeout as follows:

- **Use Workflow Step Source.** This setting determines the timeout for workflow step. The **Timeout** and **Interval** fields are disabled.

- **Specific Value.** You can enter a timeout value for the workflow step based on the Timeout Type value.

8. Click **Apply.**

Configuring Transitions for Workflow Steps

Transitions are the rules that logically connect workflow steps. You add transitions to a workflow to establish the direction a process should take, based on the results of a workflow step. For example, a request is entered 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.

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 results. Specific results are based on the workflow step's validation.
- **Other results.** All other results that do not have transitions set follow this transition.
- **All results.** All results follow this transition.
- **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. From the Workbench shortcut bar, select **Configuration > Workflows.**

   The Workflow Workbench opens.

2. Open a workflow.

   The Workflow window opens.

3. Click the **Layout** tab.

4. Right-click a workflow step.

   The workflow step is highlighted. A menu window opens.

5. Select **Add Transition.**

   The menu window closes. The workflow step remains highlighted.

6. Select the destination workflow step for the transition.

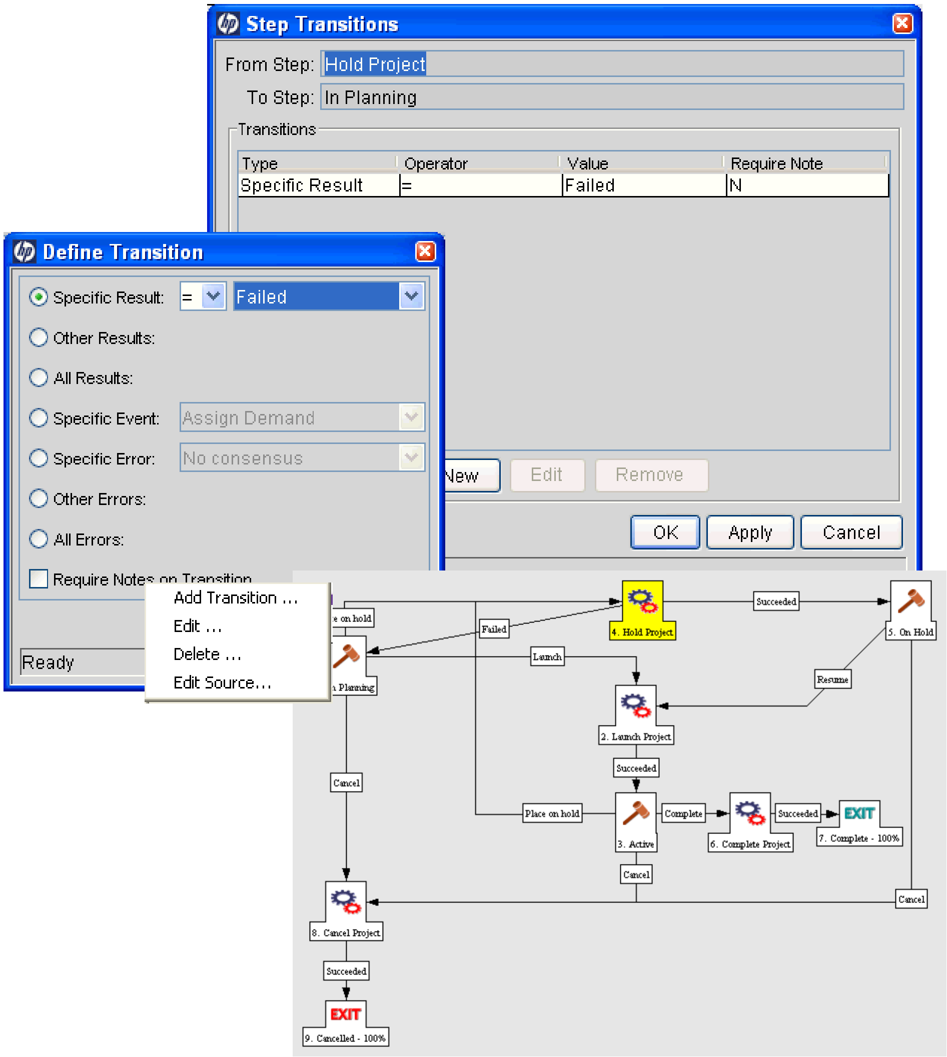
   A line with an arrowhead appears between the workflow steps. The Define Transition and Step Transitions windows opens. The Define Transition window is enabled and has many options on how to define the transition. The most common transition is Specific Results. For information on other transitions definitions, see *Adding Transitions not Based on Specific Results* on page 82.

7. In the **Specific Results** field, select the transition.

8. Click **OK.**

9. In the Step Transitions window, click **Apply** or **OK.**

   To add another validation to the transition, 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.



From Step: Hold Project

To Step: In Planning

Type	Operator	Value	Require Note
Specific Result	=	Failed	N

● Specific Result: = Failed

○ Other Results:

○ All Results:

○ Specific Event: Assign Demand

○ Specific Error: No consensus

○ Other Errors:

○ All Errors:

□ Require Notes on Transition

Ready

10. 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 following lists other transition values:

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

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.

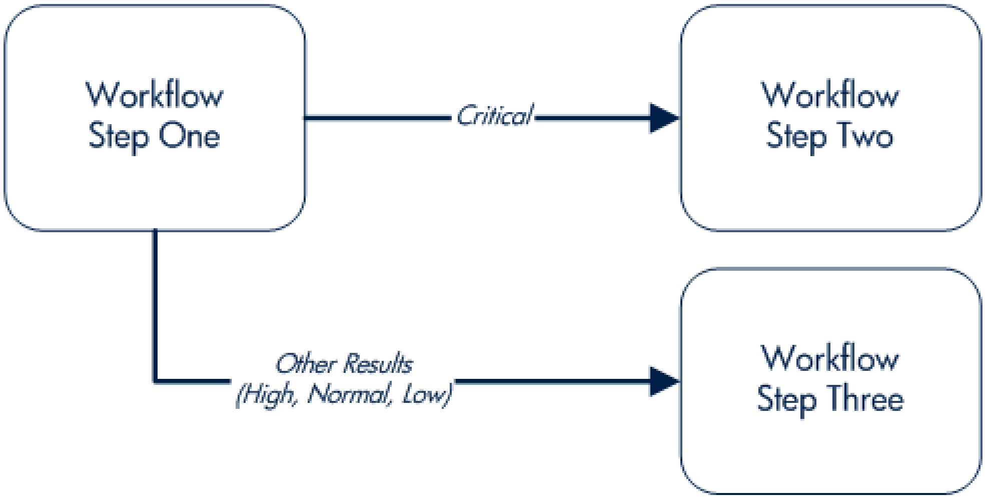
Adding Transitions Based on All But One Specific Value

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 3-13. Transitions using other results



Adding Transitions Based on All Results

It is possible to 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. Since 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 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 3-4*). As you add a transition, select Specific Error option in the Define Transition window, and then select the error.

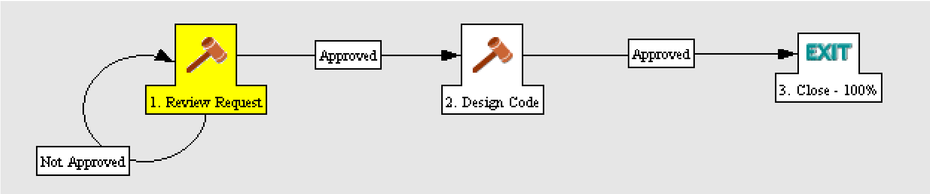
Table 3-4. Workflow transition errors

Transition Option	Meaning
Multiple Return Results	When the package level subworkflow receives multiple results from package lines that traversed through it.
No consensus	When all users of all security groups, or users linked to the workflow step need to vote, and there is no consensus.
No recipients	When none of the security groups linked to the workflow step has users linked to it, no user can act on the workflow step.
Timeout	When 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 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 3-14*).

Figure 3-14. Transitioning back to the same step



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

Do not use an immediate execution workflow step when a FAILED result is feeding directly back into the execution workflow step. This would result 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. From the Workbench shortcut bar, select **Configuration > Workflows.**

   The Workflow Workbench opens.

2. Open a workflow.

   The Workflow window opens.

3. Click the **Layout** tab.

4. Configure an immediate execution workflow step, as follows:

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

      The Execution window opens.

   b. Complete the fields as specified 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	Token
Processing Type	Immediate
Validation	Create a validation with the following validation values. ● **Succeeded** ● **Failed** ● **Failed - Reset** ● **Failed - Rejected** For details on how to create a validation, see the *Commands, Tokens, and Validations Guide and Reference*.
Enabled	Yes

   c. Click **OK.**

5. Add the new execution workflow step to the workflow.

6. Right-click the immediate execution workflow step.

   The workflow step is highlighted. A menu window opens.

7. Add the transition, as follows:

   a. Select **Add Transition.**

      The menu window closes and the workflow step remains highlighted.

   b. 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 is enabled and has many options on how to define the transition.

c. In the **Specific Results** field 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.**

d. Click **OK.**

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

   The defined transition name is added to the transition line.

8. Click **Save.**

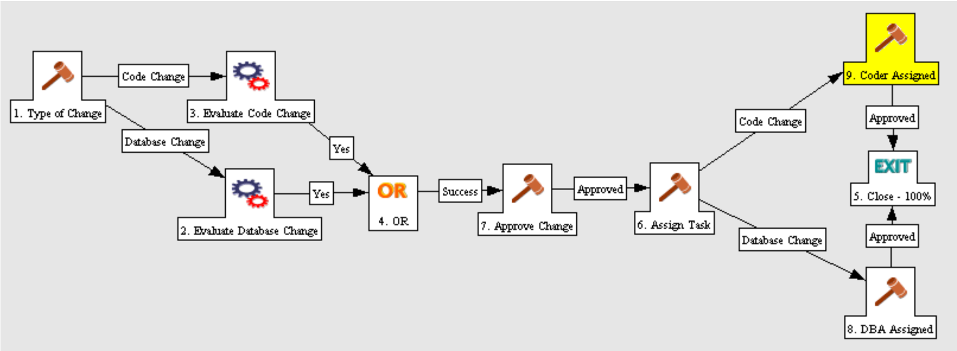
Adding Transitions Based on Previous Workflow Step Results

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, enter the workflow parameter name.

Figure 3-15 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 3-15. Add a transition based on a previous workflow step



To add a transition based on a previous workflow step:

1. From the Workbench shortcut bar, select **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.

➤ 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 3-15* on page 87.

8. Click **OK.**

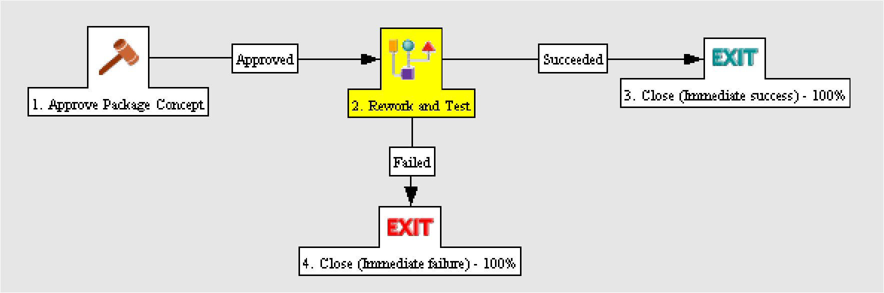
Adding Transitions To and From Subworkflows

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 3-16. Transitioning to and from subworkflows


1. Approve Package Concept
Approved
2. Rework and Test
Succeeded
EXIT
3. Close (Immediate success) - 100%
Failed
EXIT
4. Close (Immediate failure) - 100%

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

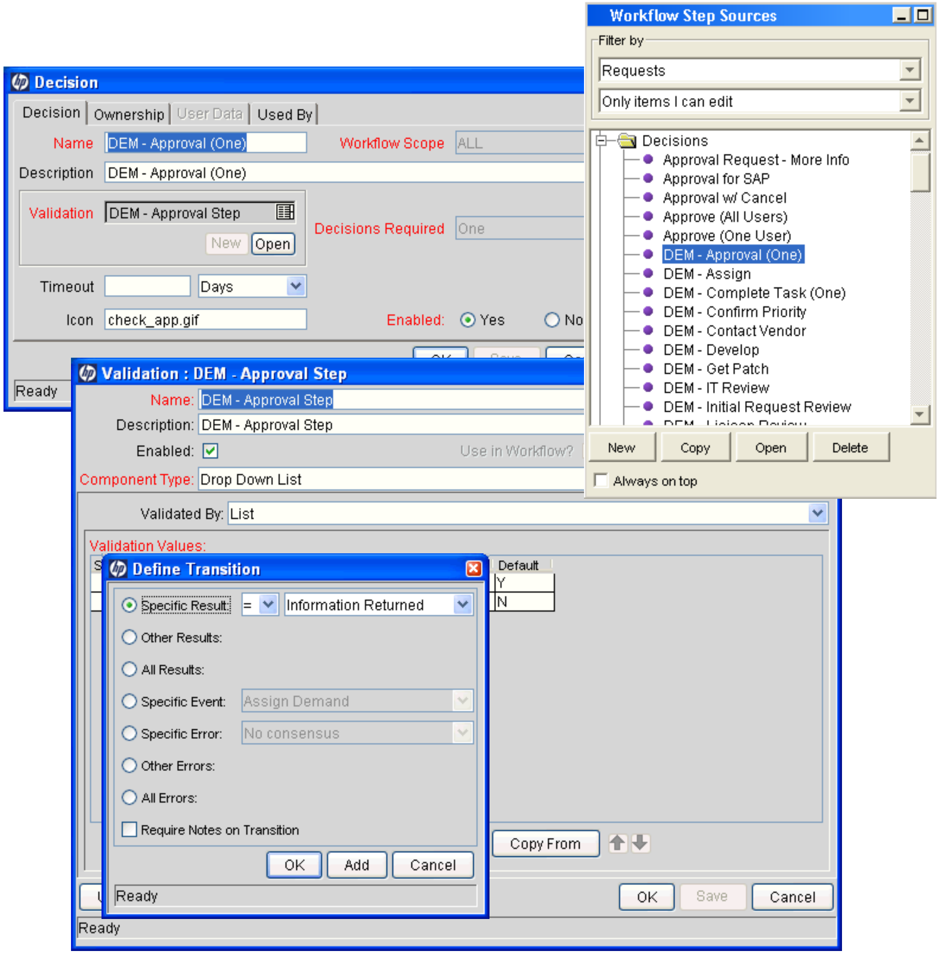
Validations determine the acceptable values for fields. Validations maintain data integrity by ensuring that the correct information is entered in a field before it is saved to the database. For workflow steps, validations ensure the correct transitions are associated with the correct workflow step.

Validations are defined for each workflow step found in the Workflow Step Source window. Opening a workflow step in the Workflow Step Source window opens the Decision window. The Decision window contains the workflow step's default information. One piece of the default information is the validation. *Figure 3-17* illustrates the Decisions window of the Approve (One User) decision workflow step and the validation listed in the Decision window. In this example, the validation is WF - Approval Step. By checking the validation, WF - Approval Step has two validation values:

- Approved
- Not Approved

Once a workflow step is added to a workflow, the transition can be added. Opening the Define Transition window for the workflow step, the validation values are displayed as the **Specific Results** field.

Figure 3-17. Workflow step sources and validations



Decision

Name: DEM - Approval (One)

Workflow Scope: ALL

Description: DEM - Approval (One)

Validation: DEM - Approval Step

Decisions Required: One

Timeout: Days

Icon: check_app.gif

Enabled: ● Yes ○ No

Validation : DEM - Approval Step

Name: DEM - Approval Step

Description: DEM - Approval Step

Enabled: ☑

Component Type: Drop Down List

Validated By: List

Validation Values:

Default
Y
N

Define Transition

● Specific Result: = Information Returned

○ Other Results:

○ All Results:

○ Specific Event: Assign Demand

○ Specific Error: No consensus

○ Other Errors:

○ All Errors:

□ Require Notes on Transition

Workflow Step Sources

Filter by: Requests; Only items I can edit

- Decisions
  - Approval Request - More Info
  - Approval for SAP
  - Approval w/ Cancel
  - Approve (All Users)
  - Approve (One User)
  - DEM - Approval (One)
  - DEM - Assign
  - DEM - Complete Task (One)
  - DEM - Confirm Priority
  - DEM - Contact Vendor
  - DEM - Develop
  - DEM - Get Patch
  - DEM - IT Review
  - DEM - Initial Request Review

□ Always on top

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 3-5 summarizes this relationship between validations and execution types.

Table 3-5. 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: ● 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.

Configuring Segregation of Duties for Workflow Steps

To set segregation of duties for a workflow step:

1. From the Workbench shortcut bar, select **Configuration > Workflows.**
2. Open a workflow.

   The Workflow window opens.
3. Right-click a workflow step.

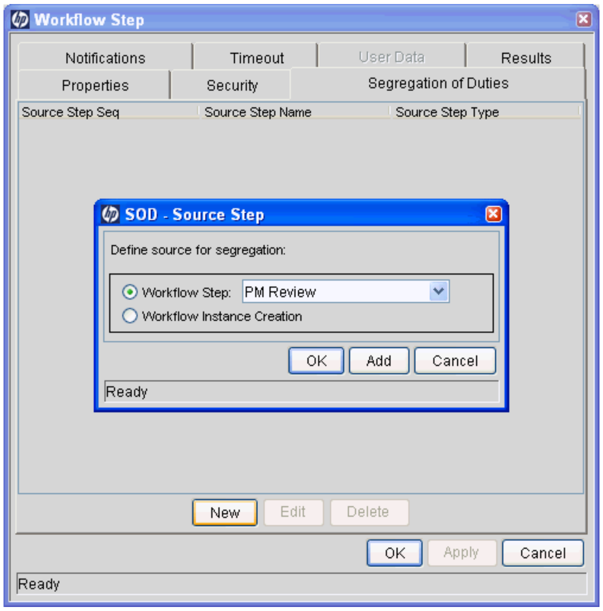
   The workflow step is highlighted. A menu window opens.
4. In the menu window, click **Edit.**

   The Workflow Step window opens.
5. In the Workflow Step window, select the **Segregation of Duties** tab.

   The **Segregation of Duties** tab opens.

6. Click **New.**

   The SOD - Source Step window opens.



Define source for segregation:

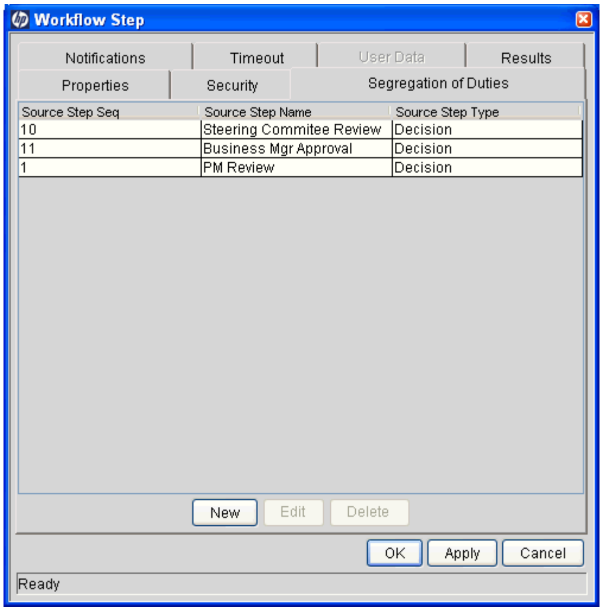
● Workflow Step: PM Review

○ Workflow Instance Creation

7. Define a segregation source for the current workflow step.

   - To segregate the current step from another workflow step, select the Workflow Step option and choose that step from the drop-down list.
   - To segregate the current step from being acted on by the creator of the package, request, or release, select the Workflow Instance Creation option.

8. In the SOD - Source Step window, click **OK** to add the segregation source to the **Segregation of Duties** tab.



Source Step Seq	Source Step Name	Source Step Type
10	Steering Commitee Review	Decision
11	Business Mgr Approval	Decision
1	PM Review	Decision

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

   The Workflow Step window closes.

10. In the Workflow window, click **Save.**

    Changes are saved to the workflow.

➤ All users who are able to act on a segregated step will be excluded from acting on the current workflow step.

Integrating Object Types and Workflows

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

Integrating Object Type Commands and Workflows

Object type commands are tightly integrated with PPM Center workflow engine. The commands contained in an object type are executed at execution workflow steps.

It is important to note the following concepts regarding command and workflow interaction:

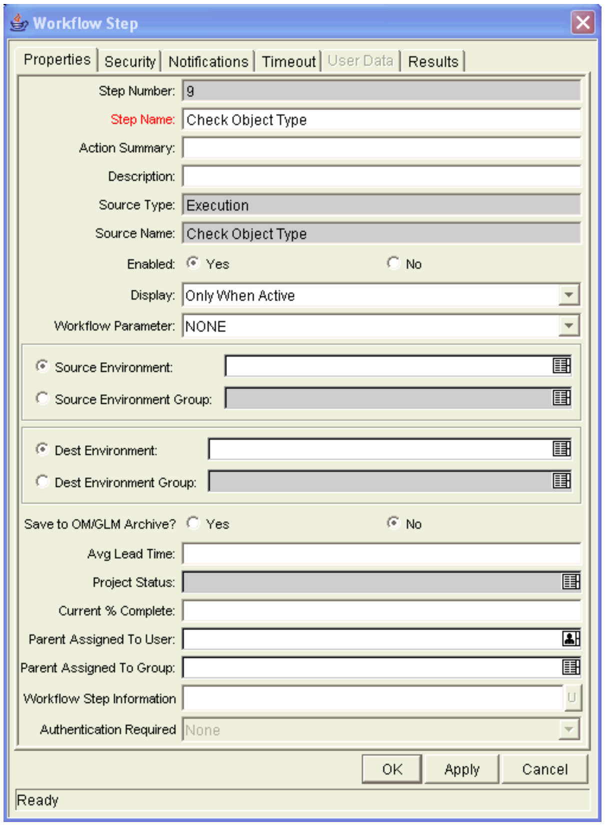
- To execute object type commands at a particular workflow step, the workflow step must be configured with the following parameters:
  - Workflow step must be an execution type step.
  - Workflow Scope = Packages
  - Execution Type = Built-in Workflow Event
  - Workflow Command = execute_object_commands
- When the object reaches the workflow step (with Workflow Command = execute_object_commands), all object type commands whose conditions are satisfied will be run in the order they are entered in the object type's command field (**Commands** tab).
- The object type can be configured to run only certain commands at a particular step.
- Each object type command can be configured so that only certain steps (within a command) are executed within a particular workflow step. This is set using conditional statements within the command.

Integrating Environments and Workflows

Environments must be linked to execution workflow steps that require connection, communication, or transfer between the clients, servers and databases used in the deployment system.

Environments are specified in the Workflow Step window, accessible from the **Layout** tab in the Workflow window. Select the source and destination environment or environment groups from the fields shown in *Figure 3-18*.

Figure 3-18. Workflow step window for environments



Step Number: 9

Step Name: Check Object Type

Action Summary:

Description:

Source Type: Execution

Source Name: Check Object Type

Enabled: ● Yes ○ No

Display: Only When Active

Workflow Parameter: NONE

● Source Environment:

○ Source Environment Group:

● Dest Environment:

○ Dest Environment Group:

Save to OM/GLM Archive? ○ Yes ● No

Avg Lead Time:

Project Status:

Current % Complete:

Parent Assigned To User:

Parent Assigned To Group:

Workflow Step Information

Authentication Required None

Ready

Choosing Source Environments Based on Application Code

Environment groups can be used to dynamically determine the source environment based on the application code for a package line. The application codes are selected based on the environments associated with the environment groups. All application codes associated with an environment are inherited by the environment group.

To enable the dynamic selection of a source environment:

1. From the Workbench shortcut bar, select **Configuration > Workflows.**

   The Workflow Workbench opens.

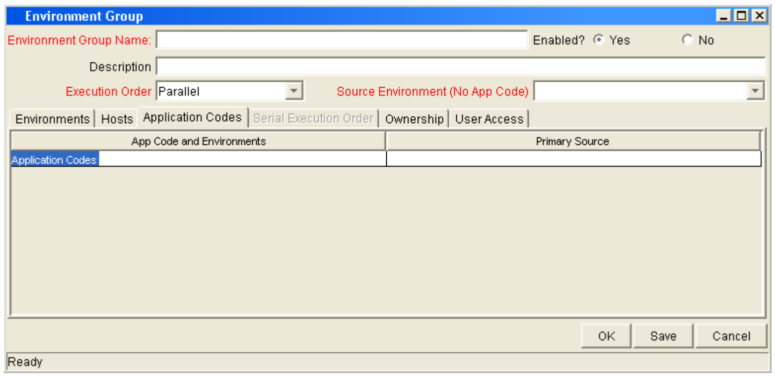
2. From the menu, select **Environments > Environment Groups.**

   The Environments Workbench opens.

3. Select an environment group or create a new environment group.

   The Environment Group window opens.

4. Click the **Application Codes** tab.



Environment Group

Environment Group Name:

Enabled? ● Yes ○ No

Description

Execution Order: Parallel

Source Environment (No App Code)

App Code and Environments	Primary Source
Application Codes	

Ready

5. For each application code, select **Primary Source Environment.**

   The primary source environment is selected automatically as the source environment in the workflow step when the associated application code is used in a specified package line.

6. Click **OK.**

Integrating Request and Package Workflows

Request (Demand Management) and package workflows (Deployment Management) can be configured to work together, communicating at key points in the request and package processes. A request workflow step can actually jump to a preselected package workflow step. The package workflow step receives the request workflow step and acts on it to go to the next step in the process.

Packages and requests can also be integrated at a level that does not rely on the workflow configuration. Attach packages and requests to each entity as references. Dependencies can then be set on these reference to control the behavior of the request or package. For example, you can specify a request as a predecessor to the package. This means the package cannot continue until the request closes.

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

As an example of when this kind of communication is 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.

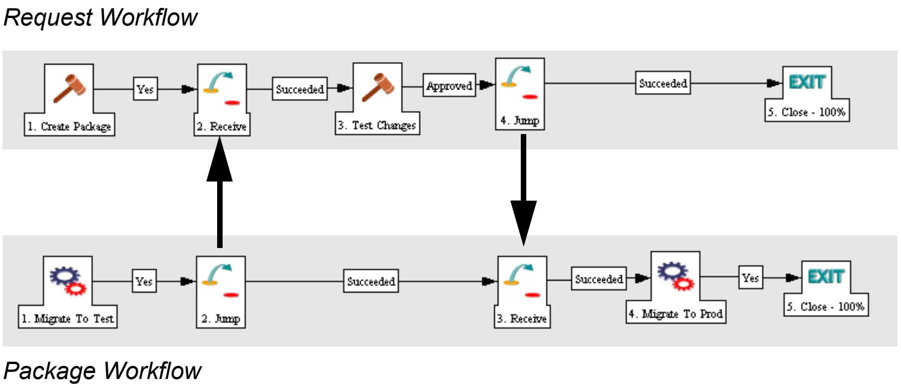
3. After the Approval step succeeds, the process jumps back to, and is received by, the request. The request then undergoes more testing and changes in the QA Environment.

4. After successfully completing the QA Test, the process jumps from the request and is received by the package.

5. Since the step has succeeded, the process can now migrate the code changes to the Production Environment.

This process is illustrated in *Figure 3-19*.

Figure 3-19. Jump/Receive workflow steps



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 for entering into and exiting 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 group 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* on page 100.

2. Create a jump workflow step using the **wf_jump Built-in Workflow Event.**

   See *Step 2. Generating Jump Step Sources* on page 101.

3. Create a receive workflow step using the **wf_receive Built in Workflow Event.**

   See *Step 3. Generating Receive Step Sources* on page 102.

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 103.

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

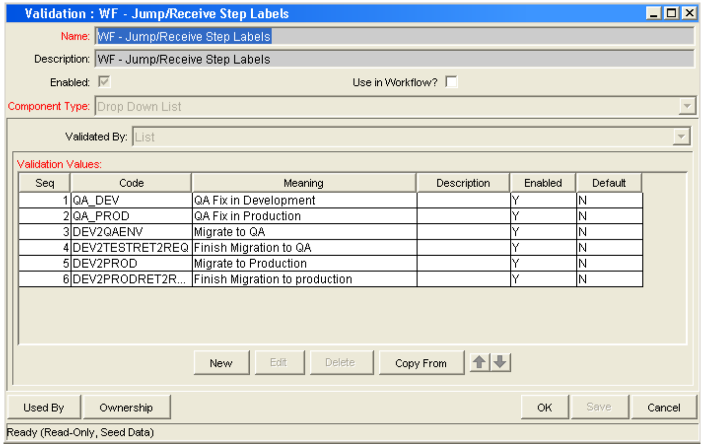
To set up the **WF - Jump/Receive Step Labels** validation:

1. From the Workbench shortcut bar, select **Configuration > Validations.**

   The Validation Workbench opens.

2. In the Validation Workbench, open **WF - Jump/Receive Step Labels.**

   The Validation window opens.



Validation : WF - Jump/Receive Step Labels

Name: WF - Jump/Receive Step Labels

Description: WF - Jump/Receive Step Labels

Enabled: ☑ Use in Workflow? □

Component Type: Drop Down List

Validated By: List

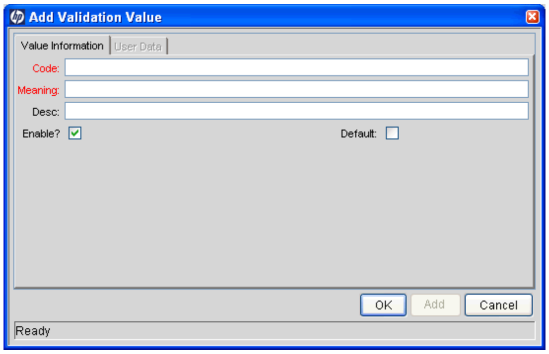
Validation Values:

Seq	Code	Meaning	Description	Enabled	Default
1	QA_DEV	QA Fix in Development		Y	N
2	QA_PROD	QA Fix in Production		Y	N
3	DEV2QAENV	Migrate to QA		Y	N
4	DEV2TESTRET2REQ	Finish Migration to QA		Y	N
5	DEV2PROD	Migrate to Production		Y	N
6	DEV2PRODRET2R...	Finish Migration to production		Y	N

Ready (Read-Only, Seed Data)

3. To define a new validation value to use to link two workflows, click **New.**

   The Add Validation Value window opens.



Add Validation Value

Code:

Meaning:

Desc:

Enable? ☑ Default: □

Ready

4. Enter the code, meaning and a description.

5. Click **OK.**

   The Validation window is enabled.

6. To select which ownership groups can edit this validation, click **Ownership.**

7. Click **OK.**

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

For More Information

For more information concerning configuring 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. From the Workbench shortcut bar, select **Configuration > Workflows.**

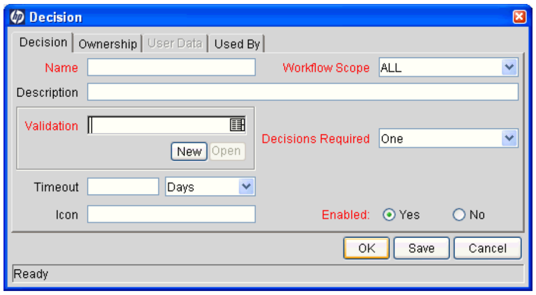
   The Workflow Workbench opens.

2. Open a workflow.

   The Workflow window opens.

3. In the Workflow Step Sources window, in the Executions folder, click **New.**

   The Execution window opens.



Workflow Scope ALL

Decisions Required One

Timeout Days

Enabled: ● Yes ○ No

4. Select either **Packages** or **Requests** from the **Workflow Scope** field, depending on the desired application of the workflow.

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

5. In the **Execution Type** field, select **Built-in Workflow Event.**

6. In the **Workflow Event** field, select **wf_jump.**

7. In the **Validation** field, select or create a validation to use to transition out of this workflow step.

   The validation values exiting the Jump workflow step must match the possible validation values entering the Jump workflow step.

8. Enter any other required or optional information, such as name, description, or processing type.

9. Click the **Ownership** tab.

10. Specify the Ownership Groups that can edit this execution workflow step.

11. Click **OK.**

    The workflow step is added to the Workflow Step Sources window.

This workflow step can now be used in any new or existing workflow within the step's defined workflow scope. Remember 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. From the Workbench shortcut bar, select **Configuration > Workflows.**

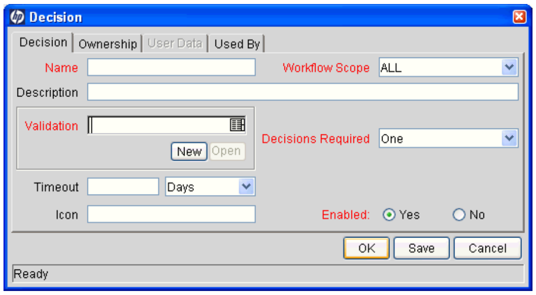
   The Workflow Workbench opens.

2. Open a workflow.

   The Workflow window opens.

3. In the Workflow Step Sources window, in the Executions folder, click **New.**

   The Execution window opens.



Workflow Scope: ALL

Decisions Required: One

Timeout: Days

Enabled: ● Yes ○ No

Ready

4. In the **Workflow Scope** field, select either **Packages** or **Requests,** depending on the workflow application.

5. In the **Execution Type** field, select **Built-in Workflow Event.**

6. In the **Workflow Event** field, select **wf_receive.**

7. Select or create a validation to use to transition out of this workflow step.

   The validation values exiting the Receive workflow step must match the possible validation values entering and exiting the Jump workflow step.

8. Enter any other required or optional information, such as name, description, or processing type.

9. Click the **Ownership** tab.

10. Select the Ownership Groups that are to be allowed to edit this execution workflow step.

11. Click **OK.**

    The Workflow Step Sources window now lists the workflow step.

This workflow step can now 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

With the jump workflow and receive workflow steps generated (see *Step 2. Generating Jump Step Sources* and *Step 3. Generating Receive Step Sources*), the two workflow steps must now be included in the 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 jump and receive pair.

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

1. From the Workbench shortcut bar, select **Configuration > Workflows.**

   The Workflow Workbench opens.

2. Open a workflow.

   The Workflow window opens.

3. From the Executions folder, add the jump workflow step.

   a. Drag the jump workflow step from the Workflow Step Sources window to the **Layout** tab for the workflow.

      The Workflow Step window opens.

   b. 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 a matching Jump/Receive Step Label field. The Jump/Receive Step Label field must be unique for any jump and receive pair.

   c. In the Workflow Step window, enter any additional workflow step information, and then click **OK.**

4. In the Executions folder, add the receive workflow step.

   a. Drag the **Receive** step from the Workflow Step Sources window to the **Layout** tab for the workflow.

      The Workflow Step window opens.

   b. 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 a matching Jump/Receive Step Label field. The Jump/Receive Step Label field must be unique for any jump and receive pair.

   c. In the Workflow Step window, enter any additional workflow step information, 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 value selected in the **Jump/Receive Step Label** field, for example **Migrate to Production.**

6. Click **Save** to save and close the workflow.

4 Configuring Workflow Components

In This Chapter:

Overview of Workflow Step Sources

This chapter covers information about Deployment 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 available steps specify the following common attributes, which are expected to remain consistent across all workflows which 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 within 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 the above settings

Configuring and Using Workflow Step Source Restrictions

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.

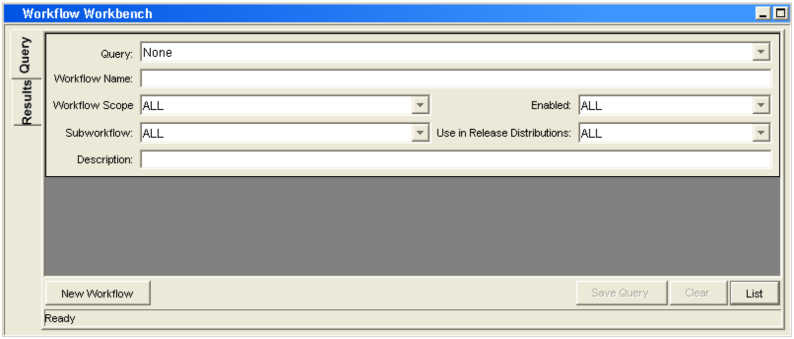
Opening the Workflow Workbench

To open the Workflow Workbench:

4. Log on to PPM Center.
5. From the menu bar, select **Administration > Open Workbench.**

   The PPM Workbench opens.
6. From the shortcut bar, select **Configuration > Workflows.**

   The Workflow Workbench opens.



Workflow Workbench

Query: None

Workflow Name:

Workflow Scope ALL

Enabled: ALL

Subworkflow: ALL

Use in Release Distributions: ALL

Description:

Ready

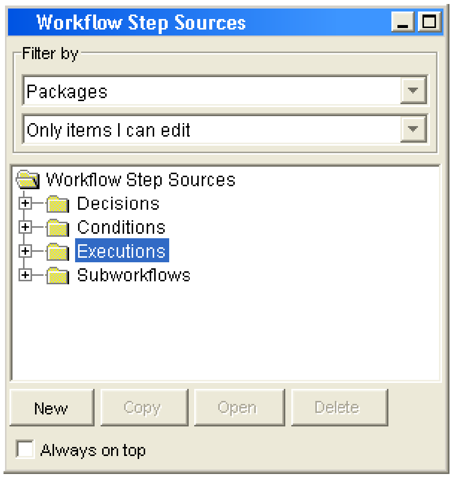
Overview of Creating Workflow Step Sources

It is possible to create new decision and execution workflow step sources from the Workflow Step Sources window. Subworkflow workflow steps are created by configuring a standard workflow to be a subworkflow (see *Creating Subworkflow Workflow Step Sources* on page 127). Condition steps cannot be added to, deleted or modified.

To create a new workflow step source:

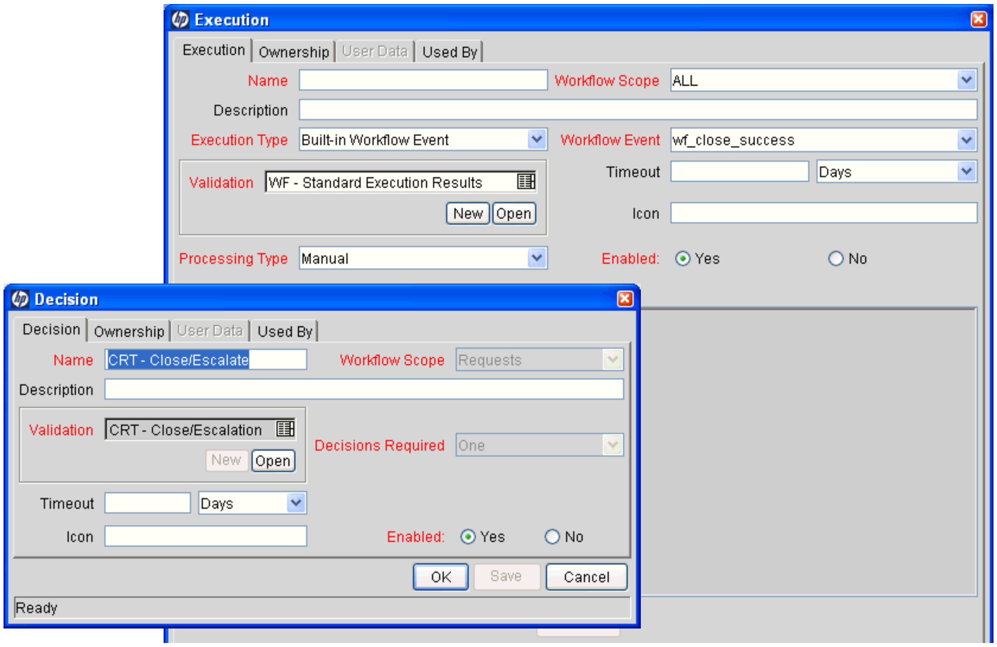
1. From the Workbench shortcut bar, select **Configuration > Workflows.**
2. Open a workflow.

   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.


Execution
Execution | Ownership | User Data | Used By
New
Open
Decision
Decision | Ownership | User Data | Used By
New
Open
OK
Save
Cancel

Execution

Name:
Workflow Scope: ALL
Description:
Execution Type: Built-in Workflow Event
Workflow Event: wf_close_success
Validation: WF - Standard Execution Results
Timeout: Days
Icon:
Processing Type: Manual
Enabled: ● Yes ○ No

Decision

Name: CRT - Close/Escalate
Workflow Scope: Requests
Description:
Validation: CRT - Close/Escalation
Decisions Required: One
Timeout: Days
Icon:
Enabled: ● Yes ○ No

Ready

8. Enter 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 page 111 or *Creating Execution Workflow Step Sources* on page 115.

9. Configure the ownership of the workflow step source.

   For information on configuring the ownership of a workflow step source, see: *Configuring Ownership of Workflow Step Sources* on page 110.

10. For **Enabled,** click **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:

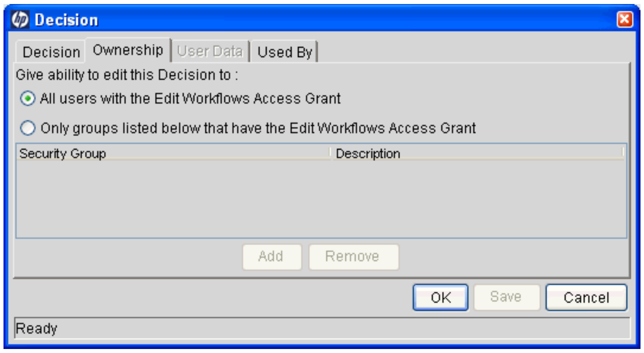
1. From the Workbench shortcut bar, select **Configuration > Workflows.**
2. Open a workflow.

   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.



Give ability to edit this Decision to :

● All users with the Edit Workflows Access Grant

○ Only groups listed below that have the Edit Workflows Access Grant

Security Group	Description

Ready

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.

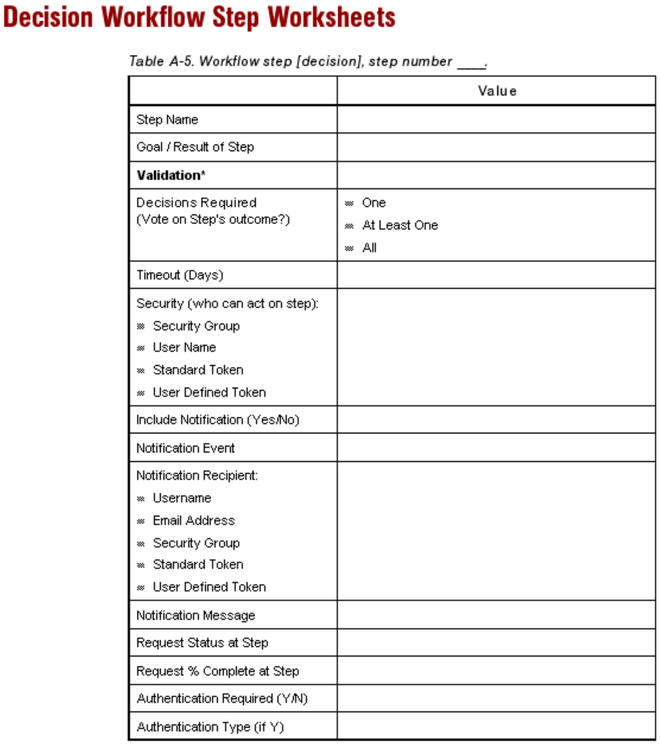
9. From the **Ownership** tab, click **OK.**

   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

Before creating a decision workflow step source, check the Decision Step Worksheet. The Decision Step Worksheet contains the information required to properly configure the workflow step source. *Figure 4-1* illustrates the Decision Step Worksheet.

Figure 4-1. Information used to create the decision step source



Decision Workflow Step Worksheets

Table A-5. Workflow step [decision], step number ____.

	Value
Step Name	
Goal / Result of Step	
Validation*	
Decisions Required (Vote on Step's outcome?)	• One • At Least One • All
Timeout (Days)	
Security (who can act on step): • Security Group • User Name • Standard Token • User Defined Token	
Include Notification (Yes/No)	
Notification Event	
Notification Recipient: • Username • Email Address • Security Group • Standard Token • User Defined Token	
Notification Message	
Request Status at Step	
Request % Complete at Step	
Authentication Required (Y/N)	
Authentication Type (if Y)	

To create a new decision workflow step source:

1. From the Workbench shortcut bar, select **Configuration > Workflows.**
2. Open a workflow.

   The Workflow window opens.

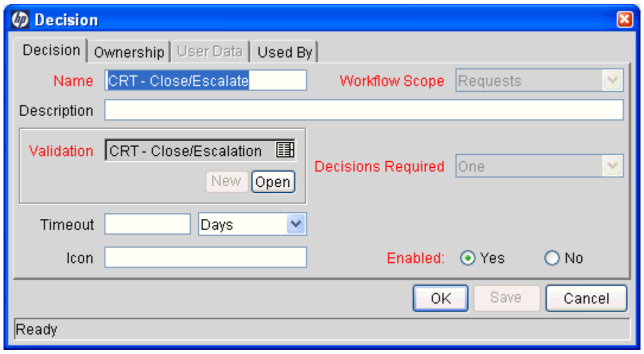
3. In the first **Filter by** field, select **Requests, Packages,** or **Release Distributions,** depending on the type of workflow.

4. Select the **Workflow Step Sources.**

5. Under Workflow Step Sources, select **Decisions.**

6. Click **New.**

   The Decision window opens.



Name: CRT - Close/Escalate

Workflow Scope: Requests

Description:

Validation: CRT - Close/Escalation

Decisions Required: One

Timeout: Days

Icon:

Enabled: ● Yes ○ No

Ready

7. From the **Decision** tab, enter the information as listed 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: • **ALL.** For all workflow types. • **Requests.** For HP Demand Management request workflows. • **Packages.** For HP Deployment Management package workflows. • **Release Distributions.** For HP 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.

Field Name	Description
Decisions Required	Defines the number of decisions required for the workflow step. Use the list to select a value. The following lists the possible values: • **One.** If **One** is 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 **At Least One** is 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 **All** is 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. 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.

Field Name	Description
Timeout	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). If this workflow step remains eligible for the value entered 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 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.

9. From the **Ownership** tab, 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 110.

10. 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.

11. Click the **Used By** tab.

    The **Used By** tab displays reference information concerning the workflow step.

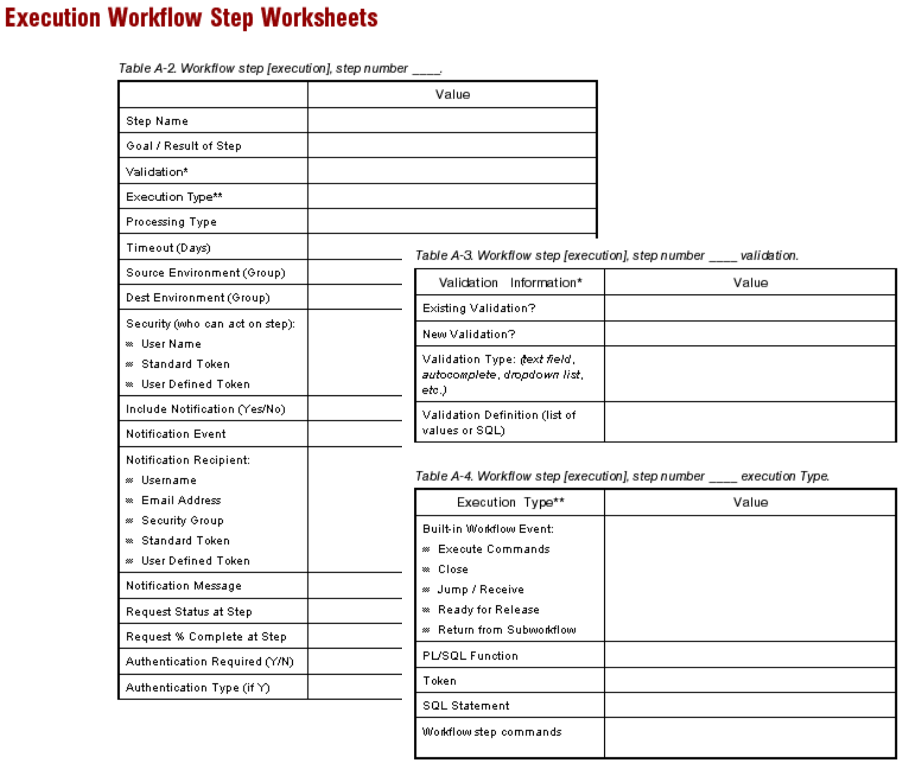
12. Click **OK.**

    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

Before creating an execution workflow step source, check the Execution Step Worksheet. The Execution Step Worksheet contains the information required to properly configure the workflow step source. *Figure 4-2* illustrates the Execution Step Worksheet.

Figure 4-2. Information used to create the execution step source



Execution Workflow Step Worksheets

Table A-2. Workflow step [execution], step number ____.

	Value
Step Name	
Goal / Result of Step	
Validation*	
Execution Type**	
Processing Type	
Timeout (Days)	
Source Environment (Group)	
Dest Environment (Group)	
Security (who can act on step): ※ User Name ※ Standard Token ※ User Defined Token	
Include Notification (Yes/No)	
Notification Event	
Notification Recipient: ※ Username ※ Email Address ※ Security Group ※ Standard Token ※ User Defined Token	
Notification Message	
Request Status at Step	
Request % Complete at Step	
Authentication Required (Y/N)	
Authentication Type (if Y)	

Table A-3. Workflow step [execution], step number ____ validation.

Validation Information*	Value
Existing Validation?	
New Validation?	
Validation Type: (*text field, autocomplete, dropdown list, etc.)*	
Validation Definition (list of values or SQL)	

Table A-4. Workflow step [execution], step number ____ execution Type.

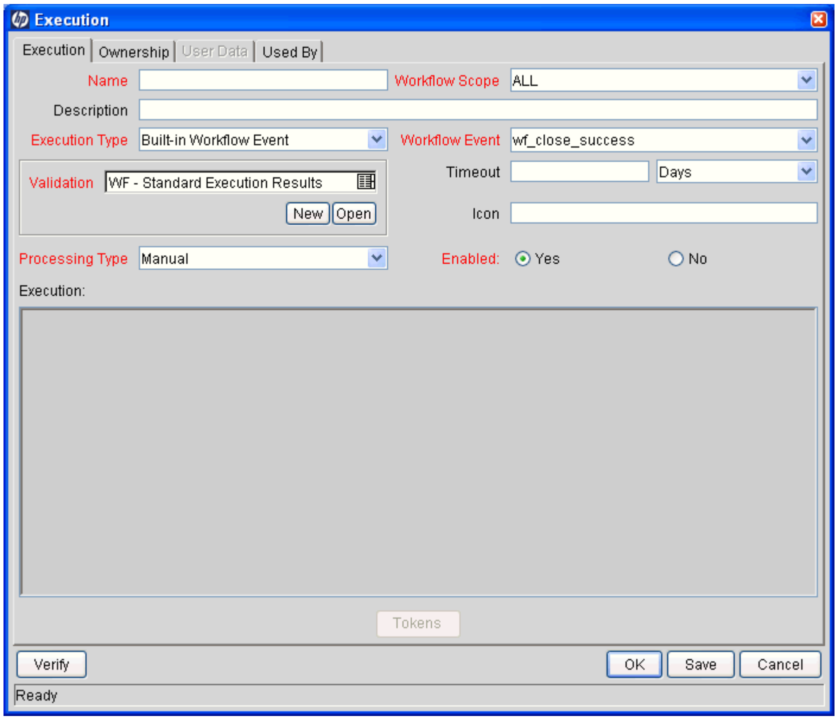
Execution Type**	Value
Built-in Workflow Event: ※ Execute Commands ※ Close ※ Jump / Receive ※ Ready for Release ※ Return from Subworkflow	
PL/SQL Function	
Token	
SQL Statement	
Workflow step commands	

To create a new execution workflow step source:

1. From the Workbench shortcut bar, select **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.



Execution

Name

Workflow Scope: ALL

Description

Execution Type: Built-in Workflow Event

Workflow Event: wf_close_success

Validation: WF - Standard Execution Results

Timeout: Days

Icon

Processing Type: Manual

Enabled: ● Yes ○ No

Execution:

Ready

7. Enter the information as listed 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: ● **ALL.** For all workflow types. ● **Requests.** For HP Demand Management request workflows. ● **Packages.** For HP Deployment Management package workflows. ● **Release Distributions.** For HP Deployment Management release workflows.
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: ● **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. ● **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.

Field Name	Description
Workflow Event	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: ● **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 HP Deployment Management package. ● **rm_ready_for_release.** Adds the current package to an existing release. Note that the existing release must be in an Open state. ● **create_package_and_wait.** Generates an HP 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.** (HP Deployment Management and HP Demand Management) Instructs the workflow to proceed to a corresponding Receive Workflow Step in another workflow. ● **wf_receive.** (HP Deployment Management and HP 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.** (HP Deployment Management and HP Demand Management) Used to route a subworkflow process back to its parent workflow.
PL/SQL Function	For Execution Type PL/SQL Function, the actual function to run. The results of the function will determine the outcome of the step. The results of the function must be a subset of the validation values for that workflow step.
Token	For Execution Type Token, the token that will be resolved. The results of the token resolution will determine the outcome of the workflow step.

Field Name	Description
SQL Statement	For Execution Type SQL Statement, the actual query to run. The results of the query will determine the outcome of the workflow step. The results of the query must be a subset of the validation values for that step.
Workflow step commands	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.
Processing Type	Defines when the execution is performed. Use the list to select a processing type. The following lists the possible values: ● **Immediate.** Executes the workflow step when the workflow step becomes eligible. ● **Manual.** Executes the workflow step manually by a user.
Validation	Validations determine the transition values for the workflow step. Use the list to select a validation.

Field Name	Description
Timeout	The amount of time that a step is eligible before completing with an error. Timeouts can be by minute, hour, weekday or week. Timeout parameters for executions are a combination of a numerical timeout value and a timeout unit, such as weekdays. If this workflow step remains eligible for the value entered in the timeout value, the request, package line, or release can be configured to send an appropriate notification. 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. 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.
Icon	You can select a different graphic to represent this steps of this workflow 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.

   The **Ownership** tab configures which security groups will have the ability to 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 the **Ownership** tab, see *Configuring Ownership of Workflow Step Sources* on page 110.

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.

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 will not be able to proceed.

Defining Execution Types

Execution workflow steps are used to perform specific actions. HP Deployment Management provides a number of 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 also create custom executions based on SQL, PL/SQL, token resolution, and custom commands.

Executing Object Type Commands

Different objects stored in the package line require unique processing at different points in a process. For example, the commands needed to migrate a file are different than the commands needed to migrate data. Therefore, it is possible to program the commands on an object type basis. The workflow can then be configured to execute the object type commands at a specific step in the process. Each package line will run its own commands, ensuring the correct execution for that object type.

HP Deployment Management includes the execution workflow step source DLV Execution w/ Reset that executes object type commands. Use this step source unless it does not meet the required specifications, such as validation or processing type.

To create this execution step source, make a copy of execution workflow step source DLV Execution w/ Reset and changes the field values as defined in *Table 4-1* on page 121.

Table 4-1. Execution window values to execute object type commands

Field Name	Description
Name	Enter a descriptive name for the step source.
Workflow Scope	Packages
Execution Type	Built-in Workflow Event
Workflow Event	execute_object_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

Closing Packages as Success

It is possible to create an execution workflow step that close a package line and marks the package line as **Success.** When all package lines are closed, the package will close. Each package workflow should resolve with a closed package. Later, the packages that were closed successfully can be reported on.

This type of step is also required when integrating request and package workflows. If a package has been created using the request workflow step Create Package and Wait, the request workflow will not proceed until the package workflow has closed.

HP Deployment Management includes the execution workflow step sources Close (Immediate success) and Close (Manual success) that performs 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 changes the field values as defined in *Table 4-2* on page 122.

Table 4-2. Execution window values for closing packages as success

Field Name	Description
Name	Enter a descriptive name for the workflow step source.
Workflow Scope	Packages
Execution Type	Built-in Workflow Event

Table 4-2. Execution window values for closing packages as success

Field Name	Description
Workflow Event	wf_close_success
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

Closing Packages as Failed

It is possible to create an execution step that closes a package and marks the package as **Failed.** Each package workflow should resolve with a closed package.

This type of step is also required when integrating request and package workflows. If a package has been created using the request workflow step Create Package and Wait, the request workflow will not proceed until the package workflow has closed.

HP Deployment Management includes the execution workflow step source Close (Immediate failure) 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 Close (Immediate failure) and changes the field values as defined in *Table 4-3* on page 122.

Table 4-3. Execution window values for closing packages as failed (page 1 of 2)

Field Name	Description
Name	Enter a descriptive name for the workflow step source.
Workflow Scope	Packages
Execution Type	Built-in Workflow Event
Workflow Event	wf_close_failure

Table 4-3. Execution window values for closing packages as failed (page 2 of 2)

Field Name	Description
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

Marking Packages Ready for Release

You can configure an execution workflow step source to feed a package into Deployment Management.

HP Deployment Management includes the execution workflow step source Ready for Release 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 Ready for Release and change the field values listed and described in *Table 4-4*.

Table 4-4. Execution window values for marking packages as Ready for Release

Field Name	Description
Name	Enter a descriptive name for the workflow step source.
Workflow Scope	Packages
Execution Type	Built-in Workflow Event
Workflow Event	rm_ready_for_release
Processing Type	Manual or Immediate
Validation	RM - Ready for Release
Enabled	Yes

Executing PL/SQL Functions and Creating Transitions Based on the Results

PL/SQL function execution workflow steps are used when a workflow needs to 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 4-5*.

Table 4-5. Execution window values for executing PL/SQL functions

Field Name	Description
Name	Enter a descriptive name for the workflow step source.
Workflow Scope	Packages
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 create a validation validated by SQL. Use the same SQL from the execution minus the WHERE clause.
Execution	Enter the SQL query.
Enabled	Yes

Executing SQL Statements and Creating Transitions Based on the Results

SQL statement execution workflow steps are used when a workflow needs to 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.

When creating the SQL statement, you must obey the following rules:

- Use only SELECT statements
- Tokens can be used within the WHERE clause
- A query must return only one value

Create a new execution step source with the field values described in *Table 4-6*.

Table 4-6. Execution window values for executing SQL statements

Field Name	Description
Name	Enter a descriptive name for the workflow step source.
Workflow Scope	Packages
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. You can create a validation validated by SQL. Use the same SQL defined for the execution minus the WHERE clause.
Execution	Enter the SQL query.
Enabled	Yes

Evaluating Tokens and Creating Transitions Based on the Results

HP Demand Management includes workflow execution steps that may be used to set up data-dependent rules for the routing of 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 4-7*.

Table 4-7. Execution window values for evaluating tokens (page 1 of 2)

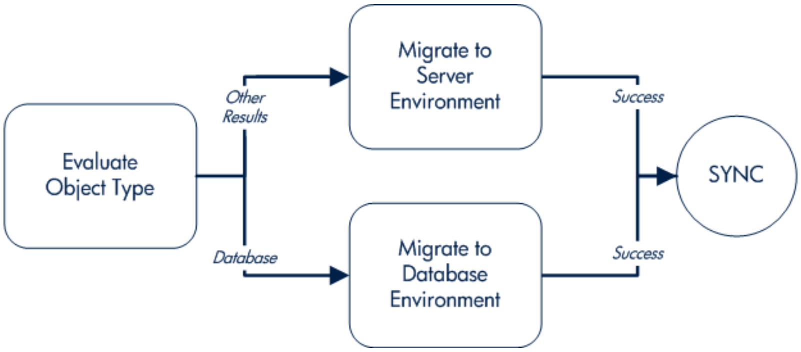
Field Name	Description
Name	Enter a descriptive name for the workflow step source.
Workflow Scope	Packages
Execution Type	Token
Processing Type	Manual or Immediate

Table 4-7. Execution window values for evaluating tokens (page 2 of 2)

Field Name	Description
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	Enter the token for the value that the transition will be based on.
Enabled	Yes

For example, IT needs to deploy changes to different servers depending on the type of object being deployed.

Figure 4-3. Transitioning based on a token


Evaluate Object Type
Other Results
Database
Migrate to Server Environment
Migrate to Database Environment
Success
Success
SYNC

IT decides to use an execution workflow step to automatically evaluate the object type and route the package line accordingly. To accomplish this, an execution workflow step source, Evaluate Object Type, is configured with the parameters listed in *Table 4-8*.

Table 4-8. Example of execution window values for evaluating tokens

Field Name	Description
Name	Evaluate Object Type
Workflow Scope	Packages
Execution Type	Token
Processing Type	Immediate
Validation	DLV - Object Type - Enabled
Execution	[PKGL.OBJECT_TYPE]
Enabled	Yes

Executing Multiple System Level Commands

System level commands can be run for execution steps of the following execution type:

- Built-in Workflow Event (execute_object_commands)
- Workflow Step Commands

When either the workflow or the object 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

A subworkflow is any workflow that is referenced from within another workflow. Use subworkflows 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.

Subworkflows are defined in the PPM Workbench using the same process as when configuring a workflow. When creating a subworkflow, be sure to set the following:

- Set the **Sub-workflow** option to **Yes.**
- Make sure that the validation for the step leaving the subworkflow layout matches the subworkflow step in the parent workflow.

Subworkflows Returning to Deployment Management Workflows

Execution workflow steps can be configured to automatically return from a subworkflow to its parent Deployment Management workflow.

For a 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. You must verify that the validation defined for the subworkflow step is synchronized with the transitions entering the return step.

HP Deployment Management includes the Return from Subworkflow execution workflow step source, which 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 changes the field values as defined in *Table 4-9*.

Table 4-9. Execution window values for returning from subworkflows

Field Name	Description
Name	Enter a descriptive name for the work step source.
Workflow Scope	Packages
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 a different validation or create a new one.)
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:

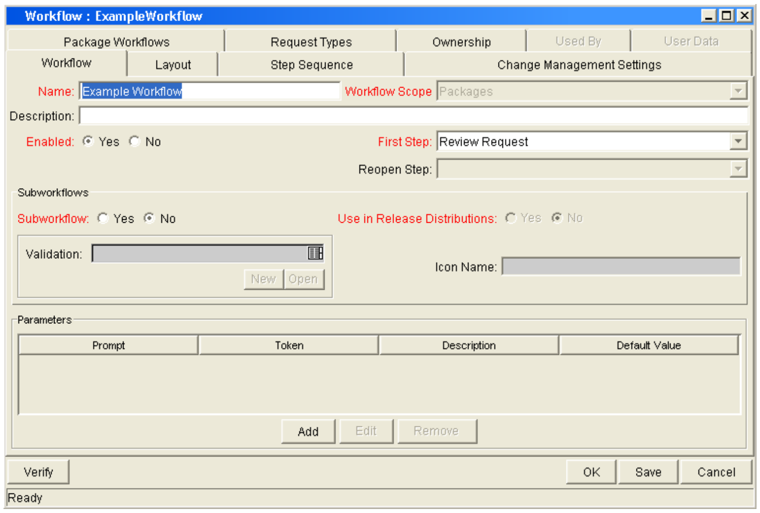
- You can use the WFI.P token prefix to reference workflow parameters.
- You can use workflow parameters in PL/SQL and SQL workflow step executions.

Creating Workflow Parameters

To create a workflow parameter:

1. From the Workbench shortcut bar, select **Configuration > Workflows.**
2. Open a workflow.

   The Workflow window opens.



Workflow : ExampleWorkflow

Name: Example Workflow

Workflow Scope: Packages

Enabled: ● Yes ○ No

First Step: Review Request

Reopen Step:

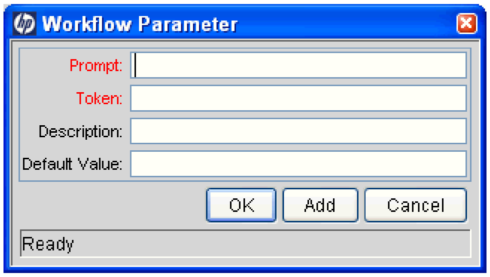
Subworkflow: ○ Yes ● No

Use in Release Distributions: ○ Yes ● No

Ready

3. From the **Workflow** tab, click **Add.**

   The Workflow Parameter window opens.



Workflow Parameter

Ready

4. Enter information in the Workflow Parameter window as specified in the following table:

Field Name	Description
Prompt	The name of the workflow parameter.
Token	The name of the token. For example, LOOP_COUNTER.
Description	A description of the workflow parameter.
Default Value	The initial value given to the workflow parameter.

5. In the **Parameters** section of the **Workflow** tab, click **Add.**

6. Click **OK.**

7. From the **Workflow** tab, click **OK.**

Example: Building a Loop Counter Using Workflow Parameters

A workflow parameter can be used to generate a counter for the number of times a workflow step enters a state.

To build a loop counter using workflow parameters:

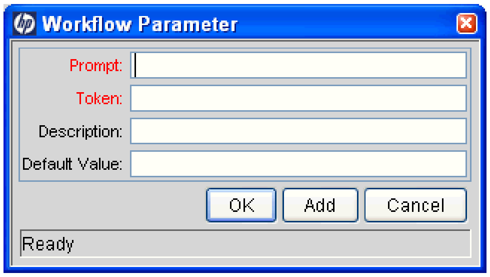
1. From the Workbench shortcut bar, select **Configuration > Workflows.**

2. Open a workflow.

   The Workflow window opens.

3. From the **Workflow** tab, click **Add.**

   The Workflow Parameter window opens.

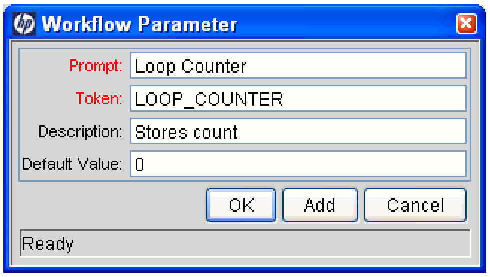


4. Enter the information as listed in the following table.

Field Name	Description
Prompt	Loop Counter
Token	LOOP_COUNTER
Description	Stores count.
Default Value	0

5. In the Parameters section of the **Workflow** tab, click **Add.**

6. Click **OK.**



Prompt: Loop Counter
Token: LOOP_COUNTER
Description: Stores count
Default Value: 0
Ready

7. From the **Workflow** tab, click **OK.**

8. Create a new immediate SQL execution workflow step.

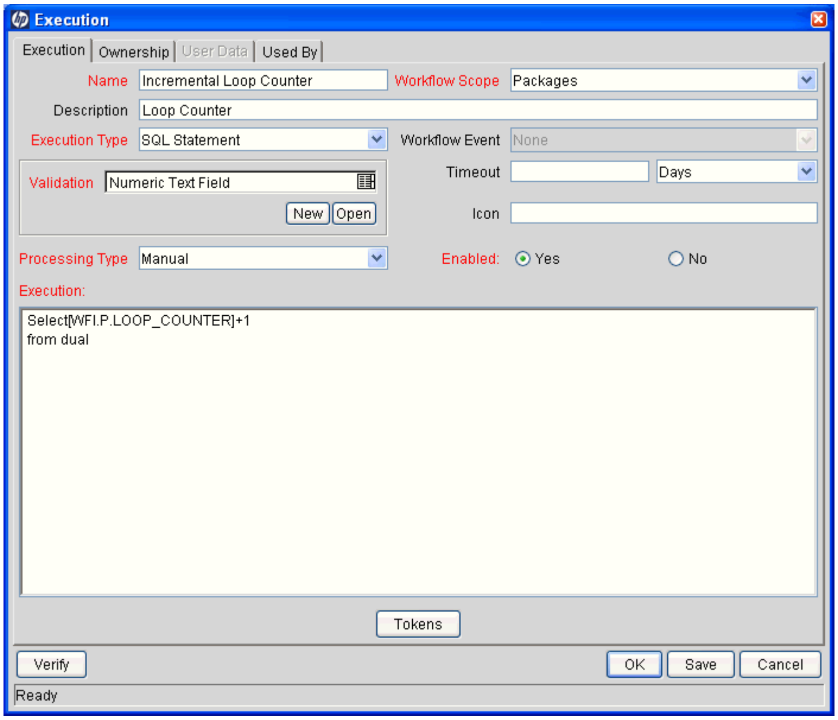
   For details on how to create an SQL execution workflow step, see *Creating Execution Workflow Step Sources* on page 115.

   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 illustrates the Execution window for the SQL execution workflow step.



Name: Incremental Loop Counter

Workflow Scope: Packages

Description: Loop Counter

Execution Type: SQL Statement

Workflow Event: None

Validation: Numeric Text Field

Timeout: Days

Icon:

Processing Type: Manual

Enabled: ● Yes ○ No

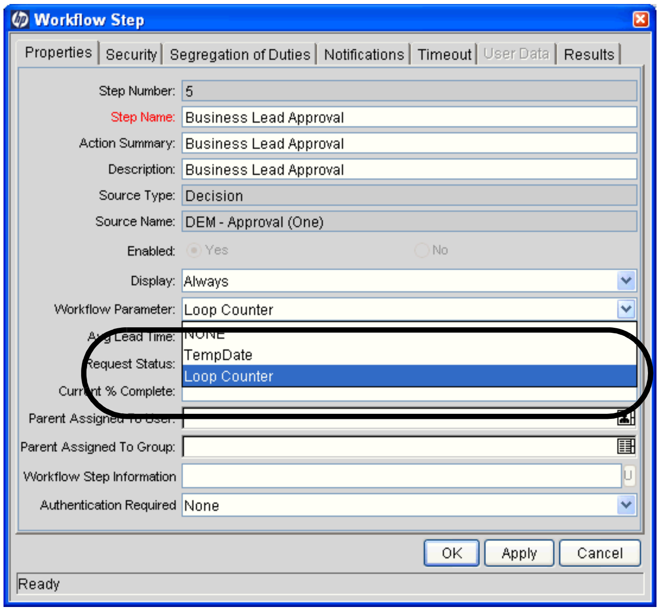
Execution:

```
Select[WFI.P.LOOP_COUNTER]+1
from dual
```

Ready

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.



Step Number: 5
Step Name: Business Lead Approval
Action Summary: Business Lead Approval
Description: Business Lead Approval
Source Type: Decision
Source Name: DEM - Approval (One)
Enabled: ● Yes ○ No
Display: Always
Workflow Parameter: Loop Counter
NONE
TempDate
Loop Counter
Avg Lead Time:
Request Status:
Current % Complete:
Parent Assigned To User:
Parent Assigned To Group:
Workflow Step Information
Authentication Required None

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.

It is possible to make changes to workflows currently in use with the same procedures and windows that you used to define the workflows. All of these procedures are performed in the Workflow Workbench window.

When modifying workflows that are being used, rules exist for which entities can be added, changed, deleted or renamed. These rules are described in *Table 4-10*.

Table 4-10. 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 currently in progress and are using that workflow.

When modifying a workflow that is in use, this 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 being stuck in that workflow step.

Performance Considerations

Updating an existing workflow step's security with a specific configuration can impact system performance. When adding dynamic security to a step, such as based on a standard or user defined token, in the Workflow Step window in the **Layout** tab, product database tables are updated to handle this new configuration. Due to the scope of these database changes, Database Statistics need to be re-run on your database.

For information about how to run database statistics on your database, see the *System Administration Guide and Reference*. For help with this procedure, contact your instance administrator.

➤ This also applies when migrating a workflow with these types of changes into an instance of the PPM Center.

Migrating a workflow with these types of changes into an instance of the PPM Center can impact system performance. Product database tables must be updated to handle this new workflow. Due to the scope of these database changes, Database Statistics need to be re-run on your database.

For information about how to run database statistics on your database, see the *System Administration Guide and Reference*. For help with this procedure, contact your instance administrator.

Copying and Testing Trial Versions of Workflows

Before modifying a workflow that is being used, 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 make sure it works correctly.
4. Determine if the workflow step is in use.To determine which steps are currently eligible, remove the incoming transition to the step that will be deleted and run the following reports. The reports will indicate if the step to be deleted is eligible for action by package lines or requests.

   - 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 are ready 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 4-10*, a step can not be deleted from a workflow when it is in use. It can only be disabled. 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 and add a new step:

1. Remove transitions to the existing workflow step you no longer want to use.

2. Add a new workflow step to the workflow.

3. Redirect the transitions to the new workflow step.

Redirecting Workflows

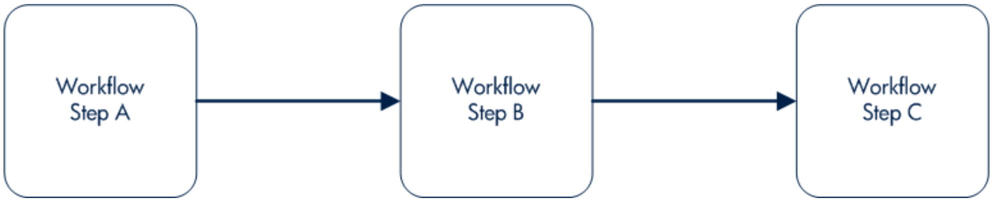
When disabling a workflow step that is currently **Eligible** for user action, the requests or package lines in that step will become stuck. Since the step is now disabled, the user cannot take action on it and will not be able to progress any further through the workflow.

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.

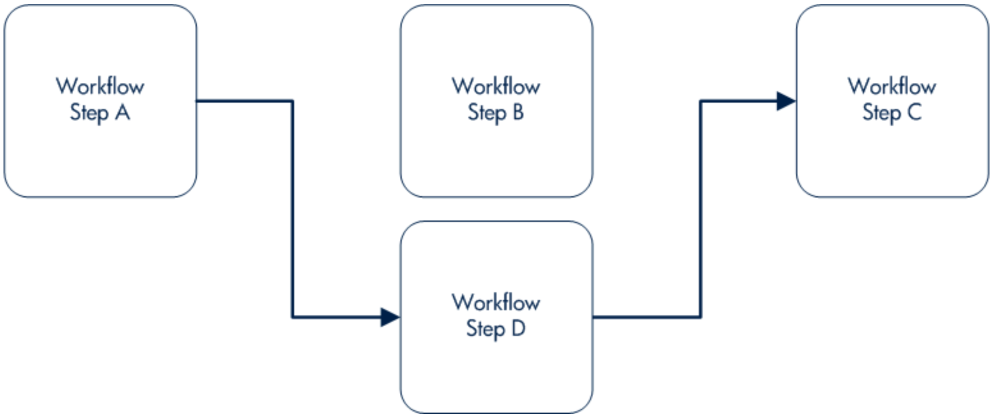
For example, consider a workflow where you wanted to disable workflow step B in the sequence shown in *Figure 4-4*.

Figure 4-4. Redirecting the workflow, step 1



After removing the incoming and outgoing transitions to B, add a new workflow step D which would connect steps A and C and let the workflow continue to process requests or package lines (see *Figure 4-5*).

Figure 4-5. Redirecting the workflow, step 2



Run the appropriate report(s) again to be sure there are no entities Eligible for action by the user in the step that was disabled.

Moving Requests or Packages Out of Steps

If the requests or packages are stuck in a step after a transition has been 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.

5 Configuring Object Types

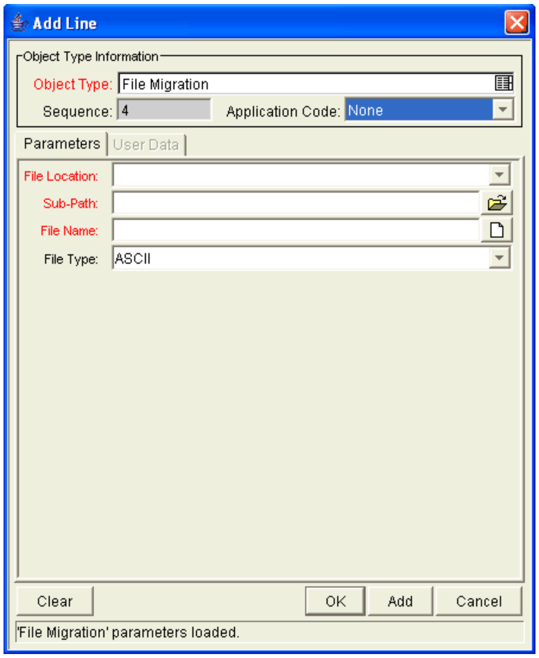
In This Chapter:

Overview of Object Types

HP Deployment Management automates complex software deployment processes. While PPM Center workflows define the process, object types are used to define the technical steps required to deploy a particular object. For example, a File Migration object type may contain the information and commands required to transfer a file from one machine to another, while a SQL script object type might address the migration and execution of database scripts.

Object types are used to create and process packages. Each package line in a package consists of one object of a specific type. To define a package line, the user selects an object type in the Add Line window in the package screen (see *Figure 5-1*). The fields required to process that object type are dynamically displayed.

Figure 5-1. Example of an object type

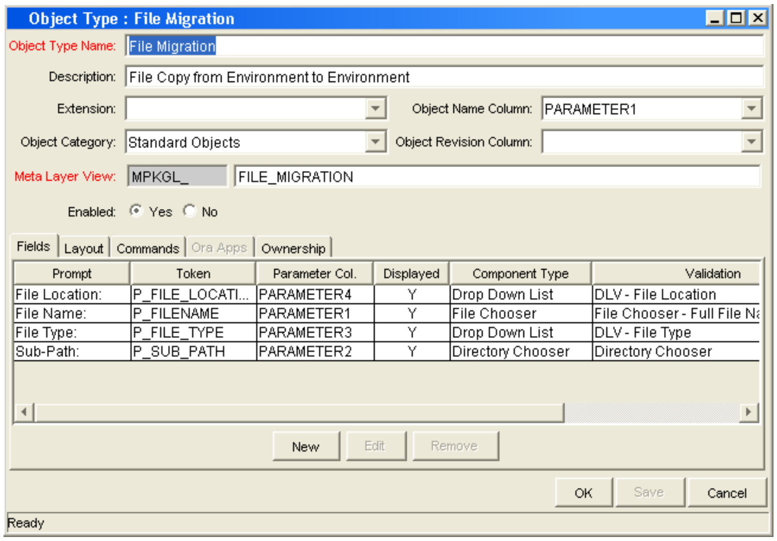


Object Type: File Migration
Sequence: 4
Application Code: None
File Location:
Sub-Path:
File Name:
File Type: ASCII

'File Migration' parameters loaded.

You use the Object Type window to create and configure object types (see *Figure 5-2*).

Figure 5-2. Object Type window



Object Type : File Migration

Object Type Name: File Migration

Description: File Copy from Environment to Environment

Extension: Object Name Column: PARAMETER1

Object Category: Standard Objects Object Revision Column:

Meta Layer View: MPKGL_ FILE_MIGRATION

Enabled: ● Yes ○ No

Prompt	Token	Parameter Col.	Displayed	Component Type	Validation
File Location:	P_FILE_LOCATI...	PARAMETER4	Y	Drop Down List	DLV - File Location
File Name:	P_FILENAME	PARAMETER1	Y	File Chooser	File Chooser - Full File Na
File Type:	P_FILE_TYPE	PARAMETER3	Y	Drop Down List	DLV - File Type
Sub-Path:	P_SUB_PATH	PARAMETER2	Y	Directory Chooser	Directory Chooser

Ready

The following is a list of the main components of an object type:

- **General information.** General information includes basic information concerning the object type, such as the object type name and the object type category. See *Configuring General Information for Object Types* on page 142.

- **Fields.** Every object type includes fields. The **Fields** tab is used to create fields for the object type. See *Creating Object Type Fields* on page 144.

- **Layout.** Once all of the fields are created for a object type, the layout of those fields can be configured using the **Layout** tab. See *Configuring Layouts for Object Types* on page 154.

- **Commands.** Commands can also be used to control certain behavior of object type fields. At specific workflow execution steps in a deployment process, it is possible to select to run the commands stored in the object type. These commands can then manipulate the data inside a object type field. This provides an advantage over the defaulting features on the **Field** tab, which can only default based on a single parameter stored on the same object type. See *Configuring Commands for Object Types* on page 158.

- **OraApps.** HP Deployment Management Extension for Oracle E-Business Suites requires specially configured object types. If HP Deployment Management Extension for Oracle E-Business Suites is not installed, the **OraApps** tab is disabled.

- **Ownership.** Configure who can edit the object type. See *Configuring Ownership for Object Types* on page 164.

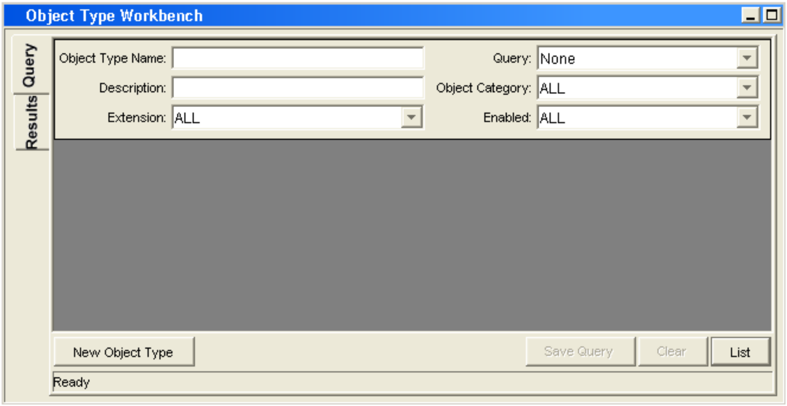
Opening the Object Type Workbench

To open the Object Type Workbench:

1. Log on to PPM Center.
2. From the menu bar, select **Administration > Open Workbench.**

   The PPM Workbench opens.
3. From the shortcut bar, select **Deployment Mgmt > Object Types.**

   The Object Type Workbench window opens.



Object Type Name:

Description:

Extension: ALL

Query: None

Object Category: ALL

Enabled: ALL

Ready

Configuring General Information for Object Types

To configure the general information of an object type:

1. From the Workbench shortcut bar, select **Deployment Mgmt > Object Types.**

   The Object Type Workbench opens.
2. Open an object type.

   The Object Type window opens.

3. Enter the information listed in the following table:

Field Name	Description
Object Type Name	The name of the object type.
Description	A useful description of how the object type is used.
Extension	For object types created for a HP Deployment Management extension. Select the extension.
Object Category	Object categories provide a way of categorizing object types in HP Deployment Management. The default values are **Custom Object** and **Standard Objects.** You can configure the values by changing the validation. Validation: DLV - Object Category - Enabled
Object Name Column	When defining an object type, this field represents the name of the object in a package line. This field shout link to a field defined for the object type. Typically, selecting an entry for the **Object Name** column is done after all of the object type fields are created. For more information, see *Setting Object Names* on page 157.
Object Revision Column	If integrating with a version control system using this object type, indicates which parameter field represents the revision number for the object. The object revisions column should be set after defining the object type fields. It is also possible to create a field in the object type to represent the revision number of the object. This field will often be a numeric text field. The deployment process can then be configured to consider the object revision number when processing the package. Typically, selecting an entry for the **Object Revision** column is done after all of the object type fields are created. For more information, see *Setting Object Names* on page 157.
Meta Layer View	Meta layer views relate information specific to PPM Center.
Enabled	Indicates whether or not the object type is available to PPM Center. Selecting **Yes** makes the object type available to the system.

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

Creating Object Type Fields

You can configure each field to behave in a certain way using the Field configuration window in the Object Type window. The Field window contains three tabs:

- **Attributes.** The **Attributes** tab is used to set basic display, edit, and requirement field properties.
- **Default.** The **Default** tab is used to set the value in the field.
- **Dependencies.** The **Dependencies** tab is used set clearing, display and requirement field properties based on values in other object type fields.

From the Field window, configure whether the field:

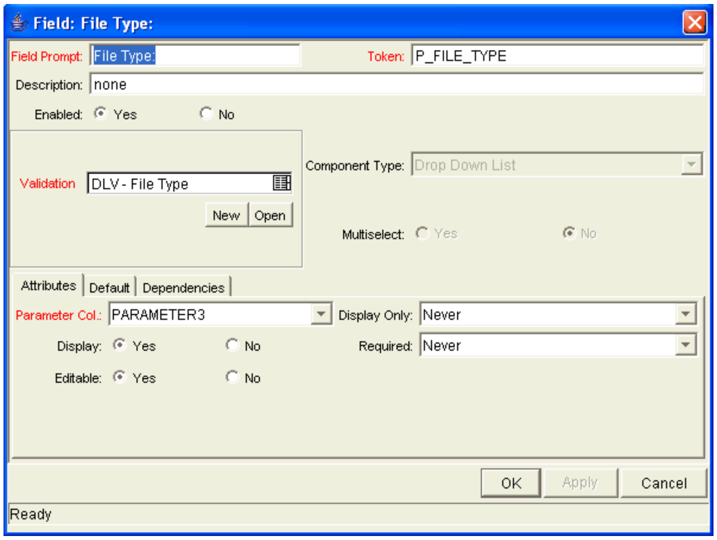
- Is displayed (for example, you might need to store a value for later use in commands, but do not want to clutter the package line)
- Can be edited under different circumstances
- Is required under different circumstances
- Defaults to a certain value
- Is dependent on values in other fields in the object type
  - Clear the field's value when another field changes
  - Display only when another field has a specific value
  - Required only when another field has a specific value

Overview of Object Type Field Validations

When configuring the object type, you can specify a different validation for each field. The validation dictates the possible values that can be entered in the field and the field type (such as text field, drop-down list, or date field).

For example, XYZ Corporation requires a **File Type** field to capture the objects to be deployed. On their object type, they set up the field as shown in *Figure 5-3*.

Figure 5-3. Field window



Field Prompt: File Type:
Token: P_FILE_TYPE
Description: none
Enabled: ● Yes ○ No
Validation: DLV - File Type
Component Type: Drop Down List
Multiselect: ○ Yes ● No
Parameter Col.: PARAMETER3
Display Only: Never
Display: ● Yes ○ No
Required: Never
Editable: ● Yes ○ No
Ready

The validation is validated by a list. This is an appropriate choice because the selection is not expected to change.

Selecting Validations

If no validation meets the necessary requirements (such as having the appropriate values), you must create one. You can also select a validation that has been configured for use at your site.

Be careful if you use a validation that is configured for use in another process. If the owner of the other process changes the validation, the validation also changes for the items in your process. Consider copying the existing validation to create a new one. You can then control who can alter the validation values by setting validation ownership.

Creating Object Type Fields

To create an object type field:

1. From the Workbench shortcut bar, select **Deployment Mgmt > Object Types.**

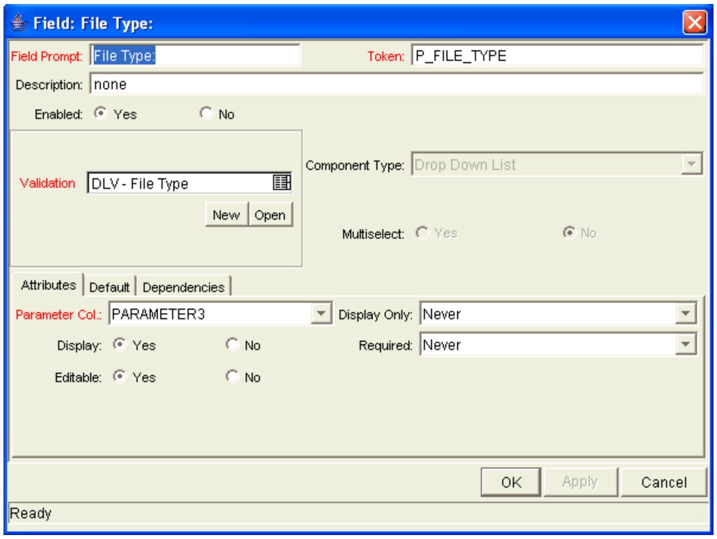
   The Object Type Workbench window opens.

2. Open an object type.

   The Object Type window opens.

3. Click **New.**

   The Field window opens.



Field Prompt: File Type
Token: P_FILE_TYPE
Description: none
Enabled: ● Yes ○ No
Validation: DLV - File Type
Component Type: Drop Down List
Multiselect: ○ Yes ● No
Parameter Col.: PARAMETER3
Display Only: Never
Display: ● Yes ○ No
Required: Never
Editable: ● Yes ○ No

4. Complete the fields in the Field window as specified in the following table:

Field Name	Description
Field Prompt	Enter the name of the new field.
Token	Type a unique name for the token.
Description	Type a description of the new field.
Enabled	Select **Yes** to make the field available to the system.

5. In the Field window, in **Validation** field, select a validation.

   See *Overview of Object Type Field Validations* on page 144 for information regarding the choosing of a validation. If no existing validation meets the requirements, create a one that does.

6. Configure field behavior.

   This consists of setting options in the field's **Attributes, Default,** and **Dependencies** tabs. See *Creating Object Type Fields* on page 144. Note that some field behavior is dependent on other object type fields. You may need to revisit this step after you create the other fields in the object type.

   To configure field behavior:

   a. Complete the fields in the **Attributes** tab, as in the following table:

Field Name	Description
Parameter Col.	Indicates the internal database column that the field value will be stored in. These values are then stored in the corresponding column in the package lines table for each Line of the given object type. Information can be stored in up to 30 columns and thus allow up to 30 fields/parameters. No two fields in an object type can use the same column.
Display Only	Indicates whether to display a field using the following options: ▪ **Always** ▪ **Never** ▪ **Use Dependency Rules** Select **Use Dependency Rules** to use the logic defined on the **Dependencies** tab.
Display	Indicates if this field is visible in the package line region of the package window.
Required	Indicates if a value needs to be specified for this field using the following options: ▪ **Always** ▪ **Never** ▪ **Use Dependency Rules** Select **Use Dependency Rules** to use the logic defined in the **Dependencies** tab.
Updateable	After a package line has been entered and submitted, it starts moving through its workflow. This attribute determines if the field can still be updated. For example, it may be necessary to ensure that a **Filename** field is not updateable once package lines of File object type start getting processed.

b. Complete the fields in the **Default** tab as specified in the following table:

Field Name	Description
Default Type	Determines whether the field has a default value. Either default the field with a constant value, default it from the value in another field, or default to a parameter.
Visible Value	If a default type of constant is selected, enter the constant value here.
Depends On	If defaulting from another field, enter the token name of that field. At runtime, when using this object type, every time a value is entered or updated in the source field, it is automatically entered or updated in this destination field.

c. Complete the fields on the **Dependencies** tab, as specified in the following table:

Field Name	Description
Clear When ___ Changes	Indicates that the current field is to be cleared if the specified field changes.
Display Only When	Indicates that the current field is to be editable only if certain logical criteria are met. This field functions with two adjacent fields: a list containing the logical qualifier and a text field. To use this functionality, select **Use Dependency Rules** from the first list.
Required When	Indicates that the current field should be required when certain logical criteria are satisfied. This field functions with two adjacent fields: a list containing logical qualifier and a text field. To use this functionality, select **Use Dependency Rules** from the first list.

7. Click **OK.**

The changes to the object type are saved.

Configuring Field Dependencies

Field behavior and properties can be linked to the value of other fields defined for that entity. For example, an object type field can become required when the value in another field in that object type equals the text Critical.

A field can be configured to:

- Clear when another field changes.
- Become read only when another field meets a logical condition defined in the following table (*Table 5-1*).
- Become required when another field meets a logical condition defined in *Table 5-2*.

Table 5-1. Field dependencies

Logical qualifier	Description
`like`	A `like` condition looks for the specified value to find any matching values in the chosen field.
`not like`	A `not like` looks for values in the chosen field that are not equal to the specified value.
`is equal to`	An `is equal to` looks for an exact match of the specified value to the contents of the field chosen.
`is not equal to`	An `is not equal to` is true when there are no results exactly matching the value of the field contents.
`is null`	An `is null` is true when the field selected is blank.
`is not null`	An `is not null` is true when the field selected is not blank.
`is greater than`	An `is greater than` looks for a numerical value in excess of the value entered in the value field.
`is less than`	An `is less than` looks for a numerical value below the value entered in the value field.
`is less than equal to`	An `is less than equal to` looks for a numerical value below, or the same as, the value entered in the value field.
`is greater than equal to`	An `is greater than equal to` looks for a numerical value in excess of, or the same as, the value entered in the value field.

To configure a field dependency:

1. From the Workbench shortcut bar, select **Deployment Mgmt > Object Types.**

   The Object Type Workbench window opens.

2. Open an object type.

   The Object Type window opens.

3. Select and edit an existing field or create a new field.

   The Field window opens.

4. Click the **Dependencies** tab.

5. Set the field dependencies, using one of the following options:

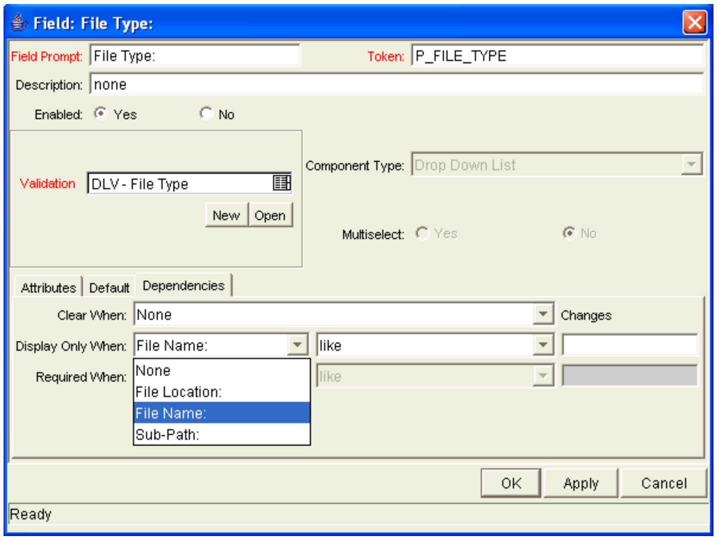
   - In the **Clear When** field, select a field name to indicate that the current field is to be cleared if the selected field changes.

   - In the **Display Only When** field, select a field name to indicate that, if certain logical criteria are satisfied, the current field is to be read-only.

     This field functions with two adjacent lists that contain logical qualifier, and a field which dynamically changes to a date field, list, or text field, depending on the validation of the selected field.

   - In the **Required When** field, select a field name to indicate that the current field is to be required if certain logical criteria are satisfied.

     This field functions with two adjacent lists that contain logical qualifier, and a field that dynamically changes to a date field, list, or text field, depending on the validation of the selected field.



Field: File Type:

Field Prompt: File Type:

Token: P_FILE_TYPE

Description: none

Enabled: ● Yes ○ No

Validation: DLV - File Type

Component Type: Drop Down List

Multiselect: ○ Yes ● No

Clear When: None

Display Only When: File Name: like

Required When: like

6. Click **OK.**

   This adds the field dependencies to the **Fields** tab of the Object Type window and closes the Field: New window.

7. Click **OK.**

Copying Object Type Fields

To copy an object type field:

1. From the Workbench shortcut bar, select **Deployment Mgmt > Object Types.**

   The Object Type Workbench opens.

2. Open an object type.

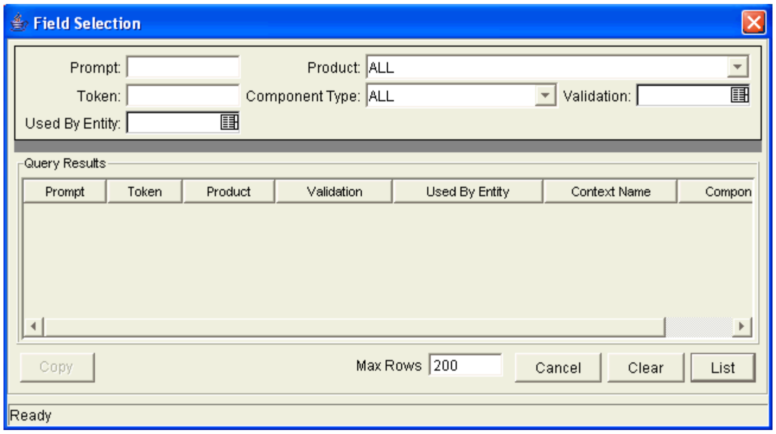
   The Object Type window opens to the **Fields** tab.

3. Click **New.**

   The Field window opens.

4. Click **Copy From.**

   The Field Selection window opens.



Product: ALL

Component Type: ALL

Max Rows 200

5. Query for the field you want to copy.

   You can use several criteria, including token name or field prompt, to query fields. You can perform more complex queries. For example, you can list all fields that reference a given validation or that a specific entity uses. Because of the number of PPM Center fields, limit the list of fields by one or more of the query criteria.

6. In the Field Selection window, select the desired field and click **Copy.**

   The Field Selection window closes and the definition of the selected field is copied to the New Field window.

7. In the New Field window, modify the fields as required.

8. Click **OK.**

   The new field is added to the **Fields** tab.

9. Click **OK.**

Editing Object Type Fields

Changes to fields for object types already used by existing package lines can have a significant impact. For example, if the column where a field value gets stored in is changed, all existing package lines for this object type will now have incorrect data. Also, remember that tokens can be used in object type commands and notifications. Any changes to these could disrupt the system.

Changing information like the field prompt or description should not affect the behavior of existing package lines.

To edit an existing field:

1. From the Workbench shortcut bar, select **Deployment Mgmt > Object Types.**

   The Object Type Workbench opens.

2. Open an object type.

   The Object Type window opens.

3. From the **Fields** tab, select a field, and then click **Edit.**

   The Field window opens.

4. Modify the field as required, and then click **OK.**

5. Click **OK.**

Removing Fields

Removing a field from an object type does not change the historical information for existing package lines using the given object type. Any values for the deleted field remain in the package lines table in the column specified in the field definition.

To remove a field permanently from an object type:

1. From the Workbench shortcut bar, select **Deployment Mgmt > Object Types.**

   The Object Type Workbench opens.

2. Open an object type.

   The Object Type window opens.

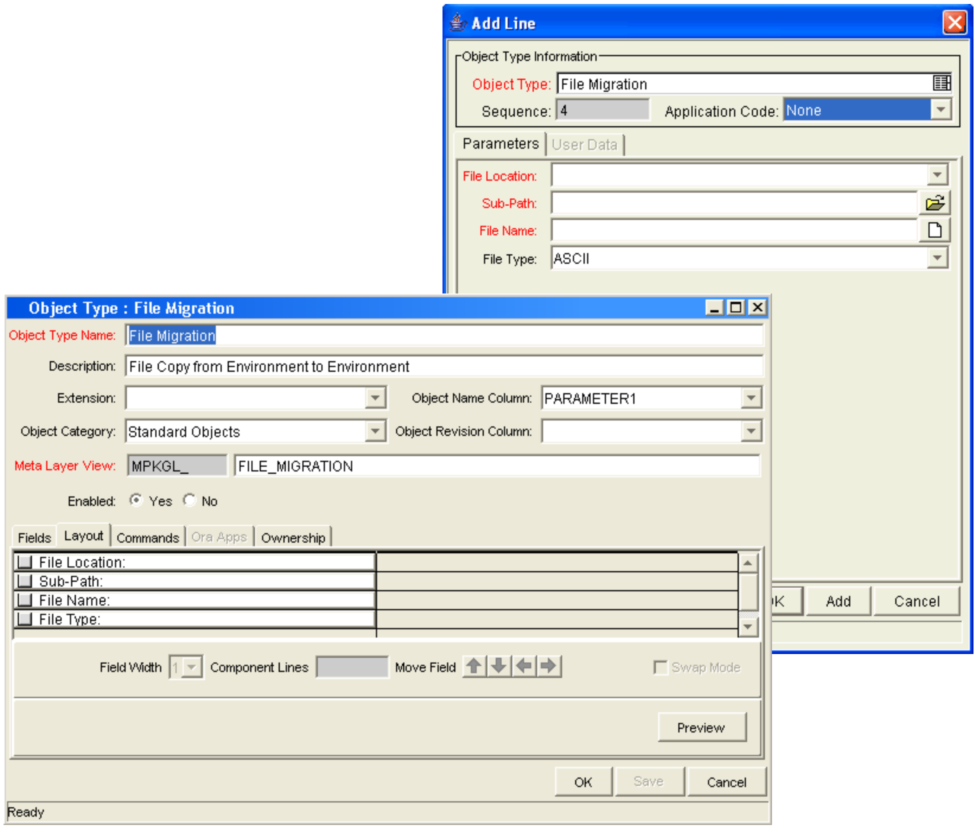
3. From the **Fields** tab, select a field, and then click **Remove.**

4. Click **OK.**

Configuring Layouts for Object Types

You can change the look of an object type using the **Layout** tab of the Object Type window. Fields can be wide or narrow. Wide fields span two columns. Narrow fields span a single column. You can also move wide fields up and down. Narrow fields can move up and down, and side to side. *Figure 5-3* shows the **Layout** tab of an object type and what the object type looks like.

Figure 5-4. Example of an object type and Layout tab



Add Line

Object Type Information

Object Type: File Migration

Sequence: 4 Application Code: None

Parameters | User Data

File Location:

Sub-Path:

File Name:

File Type: ASCII

Object Type : File Migration

Object Type Name: File Migration

Description: File Copy from Environment to Environment

Extension:

Object Name Column: PARAMETER1

Object Category: Standard Objects

Object Revision Column:

Meta Layer View: MPKGL_ FILE_MIGRATION

Enabled: ● Yes ○ No

Fields | Layout | Commands | Ora Apps | Ownership

□ File Location:
□ Sub-Path:
□ File Name:
□ File Type:

Field Width 1 Component Lines Move Field □ Swap Mode

Ready

Changing Field Widths

To change the width of an object type field:

1. From the Workbench shortcut bar, select **Deployment Mgmt > Object Types.**

   The Object Type Workbench opens.

2. Open an object type.

   The Object Type window opens.

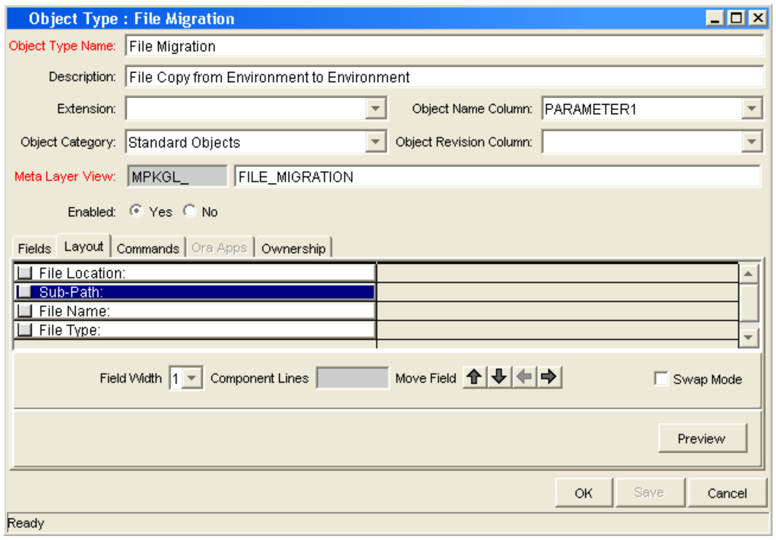
3. Click the **Layout** tab.

4. Select a field.

5. In the **Field Width** field, select **1** or **2.**

   The Layout editor does not let you 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 displays by clicking the Text Area type field and changing the value in the Component Lines attribute. If the selected field is not of type Text Area, this attribute is blank and not updateable.



Object Type : File Migration

Object Type Name: File Migration

Description: File Copy from Environment to Environment

Extension:

Object Name Column: PARAMETER1

Object Category: Standard Objects

Object Revision Column:

Meta Layer View: MPKGL_ FILE_MIGRATION

Enabled: ● Yes ○ No

- File Location:
- Sub-Path:
- File Name:
- File Type:

Field Width: 1

Component Lines:

□ Swap Mode

Ready

6. Click **OK.**

Moving Fields

To move an object type field:

1. From the Workbench shortcut bar, select **Deployment Mgmt > Object Types.**

   The Object Type Workbench opens.

2. Open an object type.

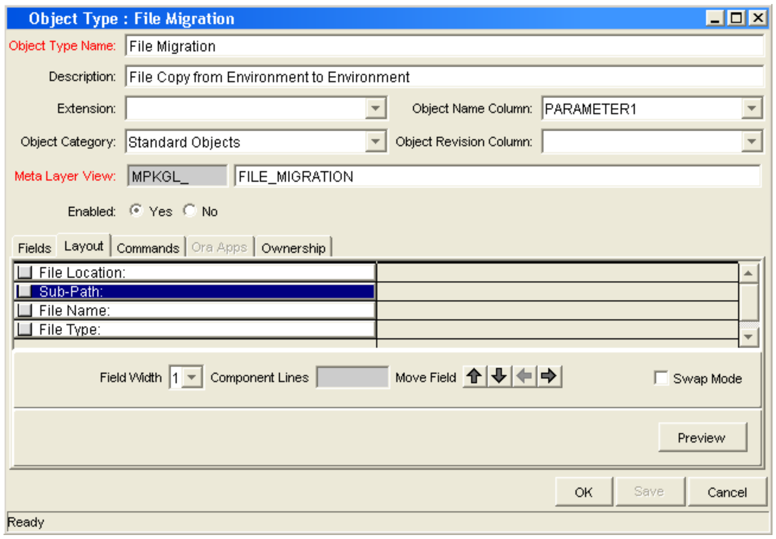
   The Object Type window opens.

3. Click the **Layout** tab.

4. Select a field.

   To select a range of fields, use the **`Shift`** key. (You cannot use the **`Ctrl`** key to select nonadjacent field names.)

5. Use the arrow pointers to reposition the fields in the layout builder.



Object Type : File Migration

Object Type Name: File Migration

Description: File Copy from Environment to Environment

Extension:

Object Name Column: PARAMETER1

Object Category: Standard Objects

Object Revision Column:

Meta Layer View: MPKGL_ FILE_MIGRATION

Enabled: ● Yes ○ No

Field Width 1

Component Lines

□ Swap Mode

Ready

6. To switch the positions of two fields:

   a. Select the first field, and then select the **Swap Mode** option.

      An S is displayed in the option section of the selected field.

   b. Double-click the second field.

      After the fields change positions, the Swap Mode option is cleared.

7. To open a window that displays a preview of your layout, click **Preview.**

   Note that:

   - If all of the fields are one column wide, then all displayed columns automatically span the available section whenever a user views a package line of the given object type.

   - Rows that contain no fields are ignored. They are not displayed as blank lines.

   - Any fields that are not displayed do not affect the layout. They are considered the same as a blank field.

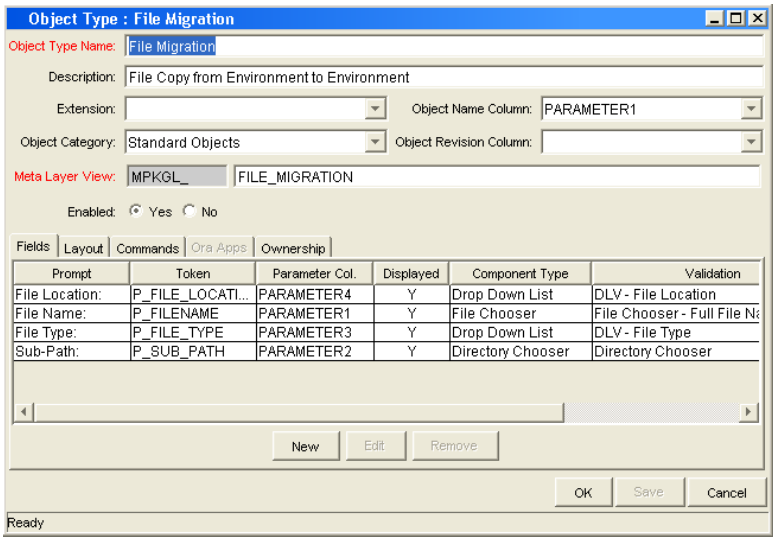
8. Click **OK.**

   The changes to the object type are saved.

Setting Object Names

When defining an object type, it is important to choose one field to represent the name of this object in a package line. This field is the object name. To designate a field as the object name, select that field's Parameter Column in the **Object Name Column** field.

For example, to designate File Name as the object name field for a File Migration object type, select the File Name field's Parameter Column in the **Object Name Column** field.



Object Type : File Migration

Object Type Name: File Migration

Description: File Copy from Environment to Environment

Extension:

Object Name Column: PARAMETER1

Object Category: Standard Objects

Object Revision Column:

Meta Layer View: MPKGL_ FILE_MIGRATION

Enabled: ● Yes ○ No

Prompt	Token	Parameter Col.	Displayed	Component Type	Validation
File Location:	P_FILE_LOCATI...	PARAMETER4	Y	Drop Down List	DLV - File Location
File Name:	P_FILENAME	PARAMETER1	Y	File Chooser	File Chooser - Full File Na
File Type:	P_FILE_TYPE	PARAMETER3	Y	Drop Down List	DLV - File Type
Sub-Path:	P_SUB_PATH	PARAMETER2	Y	Directory Chooser	Directory Chooser

In the package window, the object name for each package line is displayed in the **Object Name** column on the **Status** tab.

The object name field drives additional functionality:

- If the object name field is a file chooser or an auto-complete, multi-selection is automatically enabled on this field when users add a line to a package with this object type. If multiple values are selected, a new package line for each value will be created, allowing users to add multiple lines to a package simultaneously.
- All migrations are tracked in the database tables KENV_ENV_ CONTENTS and KENV_ENV_CONTENTS_HIST. The value of a package line's object name field is stored in these tables (along with other relevant data) whenever a migration occurs.
- You can use the Object Type Workbench to query the Object Name.

Setting Object Revisions

You can create a field on the object type to represent the revision number of the object. This field is often a numeric text field. You can then configure the deployment process to consider the object revision number during package processing.

Configuring Commands for Object Types

Object types can have many commands and each command can have many command steps. A command can be viewed as a particular function for an object type. 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 needs to take place, and these events are defined in the command steps.

An additional level of flexibility is introduced if some commands can only be executed under certain circumstances. This is powered by the **Condition** field of the commands (see *Command Conditions* on page 163).

For More Information

For more information about how to create commands and special commands, see the *Commands, Tokens, and Validations Guide and Reference*.

Adding Commands to Object Types

To add commands to object types:

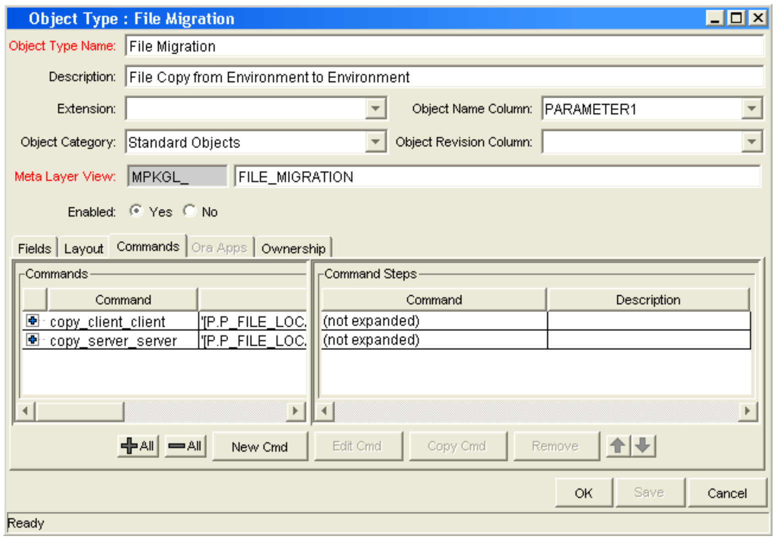
1. From the Workbench shortcut bar, select **Deployment Mgmt > Object Types.**

   The Object Type Workbench opens.

2. Open an object type.

   The Object Type window opens.

3. Click the **Commands** tab.



Object Type : File Migration

Object Type Name: File Migration

Description: File Copy from Environment to Environment

Extension:

Object Name Column: PARAMETER1

Object Category: Standard Objects

Object Revision Column:

Meta Layer View: MPKGL_ FILE_MIGRATION

Enabled: ● Yes ○ No

Commands

	Command	
⊞	copy_client_client	'[P.P_FILE_LOC.
⊞	copy_server_server	'[P.P_FILE_LOC.

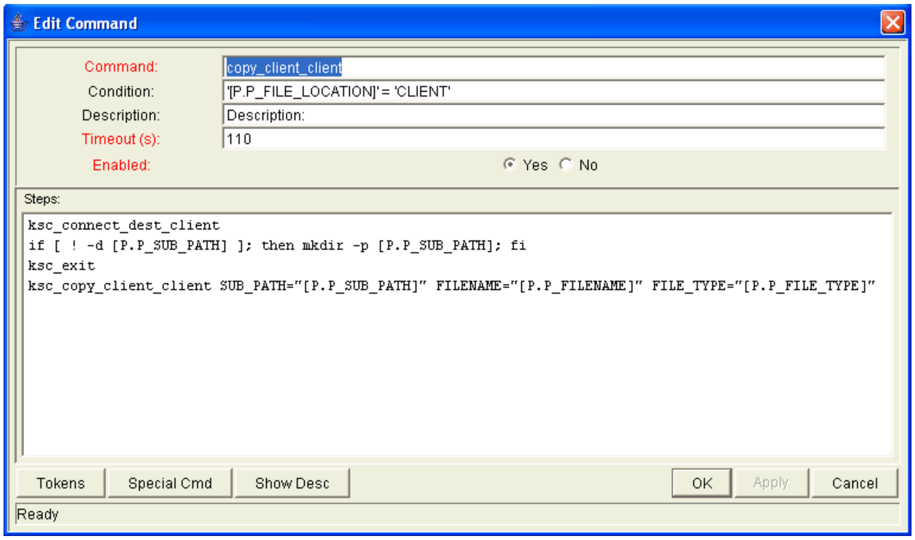
Command Steps

Command	Description
(not expanded)	
(not expanded)	

Ready

4. Click **New Cmd.**

   The New Command window opens.



Command: copy_client_client

Condition: '[P.P_FILE_LOCATION]' = 'CLIENT'

Description: Description:

Timeout (s): 110

Enabled: ● Yes ○ No

Steps:

```
ksc_connect_dest_client
if [ ! -d [P.P_SUB_PATH] ]; then mkdir -p [P.P_SUB_PATH]; fi
ksc_exit
ksc_copy_client_client SUB_PATH="[P.P_SUB_PATH]" FILENAME="[P.P_FILENAME]" FILE_TYPE="[P.P_FILE_TYPE]"
```

5. Complete the fields in the New Command window as specified in the following table:

Field Name	Description
Command	A simple name for the command.
Condition	A condition that determines whether the steps for the command are executed or not. For more information, see *Command Conditions* on page 163.
Description	A description of the command.
Timeout	The amount of time the command can run before its process is terminated. This mechanism is used to abort commands that are hanging or taking an abnormal amount of time to run.
Enabled?	Indicates whether the command is enabled for execution.

6. Click **OK.**

   The **Commands** tab lists the new command.

7. Click **OK.**

Editing Commands of Object Types

To edit a command on an object type:

1. From the Workbench shortcut bar, select **Deployment Mgmt > Object Types.**

   The Object Type Workbench opens.

2. Open an object type.

   The Object Type window opens.

3. Click the **Commands** tab.

4. Click **Edit Cmd.**

   The Edit Command window opens.

5. Select the command to edit.

6. Enter the information specified in the following table:

Field Name	Description
Command	A simple name for the command.
Condition	A condition that determines whether the steps for the command are executed or not. (See *Command Conditions* on page 163 for more information.)
Description	A description of the command.
Timeout	The amount of time the command will be allowed to run before its process is terminated. This mechanism is used to abort commands that are hanging or taking an abnormal amount of time.
Enabled?	Indicates whether the command is enabled for execution.

7. Click **OK.**

   The **Commands** tab lists the edited command.

8. Click **OK.**

Copying Commands in Object Types

To copy a command in an object types:

1. From the Workbench shortcut bar, select **Deployment Mgmt > Object Types.**

   The Object Type Workbench opens.

2. Open an object type.

   The Object Type window opens.

3. Click the **Commands** tab.

4. Select the command to copy, and then click **Copy Cmd.**

   The command is copied to another line in the **Commands** tab.

5. Click **OK.**

   The changes to the object type are saved.

Deleting Commands in Object Types

To copy a command in an object types:

1. From the Workbench shortcut bar, select **Deployment Mgmt > Object Types.**

   The Object Type Workbench opens.

2. Open an object type.

   The Object Type window opens.

3. Click the **Commands** tab.

4. Select the command to delete, and then click **Remove.**

5. Click **OK.**

Command Conditions

Depending on the execution context, it might be necessary to run a different set of commands. This flexibility is achieved through the use of conditional commands. 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. Some example conditions are detailed in *Table 5-2*. Be sure to place single quotes around string literals or tokens that are to evaluate strings.

Table 5-2. 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.

For More Information

The condition can include tokens. For more information concerning tokens, see the *Commands, Tokens, and Validations Guide and Reference*.

Configuring Ownership for Object Types

To define 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 an object type can edit, copy, or delete this object type.

If a security group is disabled or loses the ability to edit object types, its members also can no longer edit an object type.

➤ For more information on access grants, see the *Security Model Guide and Reference*.

Adding Ownerships to Object Types

To add an ownership:

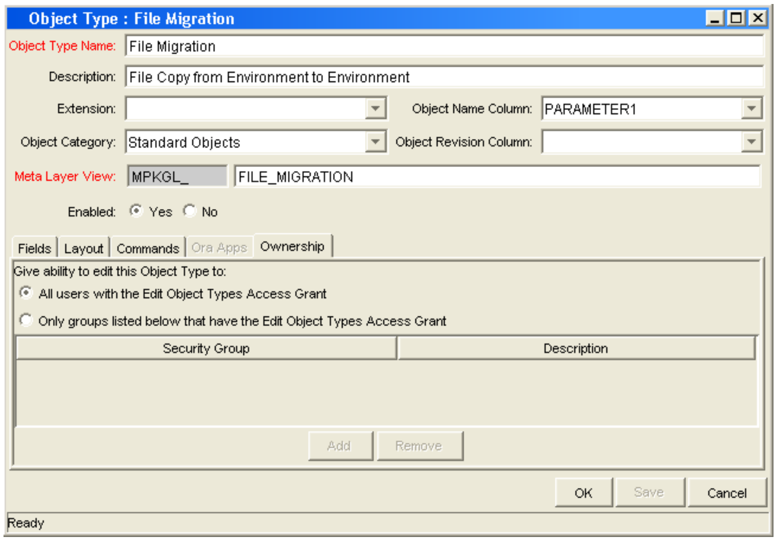
1. From the Workbench shortcut bar, select **Deployment Mgmt > Object Types.**

   The Object Type Workbench opens.

2. Open an object type.

   The Object Type window opens.

3. Click the **Ownership** tab.



Object Type : File Migration

Object Type Name: File Migration

Description: File Copy from Environment to Environment

Extension:

Object Name Column: PARAMETER1

Object Category: Standard Objects

Object Revision Column:

Meta Layer View: MPKGL_ FILE_MIGRATION

Enabled: ● Yes ○ No

Give ability to edit this Object Type to:

● All users with the Edit Object Types Access Grant

○ Only groups listed below that have the Edit Object Types Access Grant

Security Group	Description

Ready

4. Select one of the following:

   - **All users with the Edit Object Type Access Grant.**
   - **Only groups listed below that have the Edit Object Type Access Grant.**

   For **Only groups listed below that have the Edit Object Type Access Grant:**

   a. Click **Add.**

      The Add Security Groups window opens.

   b. Select the security groups and click **OK.**

      The Add Security Groups window lists the selected security groups.

   c. Click **OK.**

      The security group is added to the **Ownership** tab.

5. Click **OK.**

Deleting Ownerships from Object Types

To delete an ownership:

1. From the Workbench shortcut bar, select **Deployment Mgmt > Object Types.**

   The Object Type Workbench opens.

2. Open an object type.

   The Object Type window opens.

3. Click the **Ownership** tab.

4. Select an ownership.

➤ The **All users with the Edit Object Type access grant** option assigns ownership of the object type to users who can edit object types.

5. Click **Remove.**

6. Click **OK.**

Using Commands to Change Field Values

Commands can also be used to control certain behavior of object type fields. At specific points (execution workflow steps) in the deployment process, it is possible to run the commands stored in the object type. These commands can then manipulate the data inside an object type field. 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 field located in the same object type.

The ksc_store special command can perform this function. For information on using this and other commands, see the *Commands, Tokens, and Validations Guide and Reference*.

You may find it useful to use commands to control field values in the following situations:

- To store a value from an execution in a custom field
- To clear a field after you evaluate several parameters

6 Configuring Releases and Distributions

In This Chapter:

Overview of Releases and Distributions

A release is a group of packages and related requests that need to be deployed together. Release management provides an interface through which users can group, view and execute these packages. You can add packages to a release either by including a Ready for Release step in the package workflow or by the release manager through the Release window.

For example, a software company has a product update release scheduled five months from now. In order to ensure a smooth product delivery, they decide to track all changes to their original code using release management. As developers complete their packages, those packages are included in a release and processed together. By grouping every required change in the release, the company is able to quickly and easily assess the state of the product delivery.

Workflow Scope

Each workflow has an associated workflow scope. The workflow scope determines which release management entities can be processed through that workflow. The workflow scope can be one of the following:

- Packages
- Requests
- Release Distributions

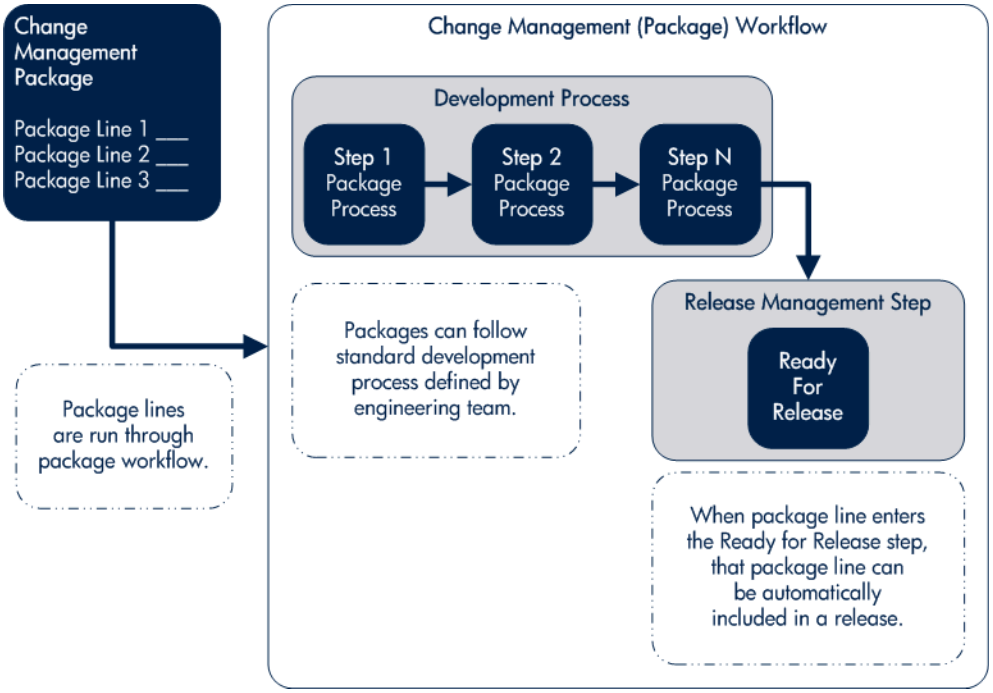
Release management uses workflows with Packages and Release Distribution scopes. Certain workflow configuration restrictions are enforced by the workflow scope. For example, release distribution workflows cannot include the wf_jump and wf_receive workflow events.

Release Management and Package Workflows

You can configure your standard package workflows to feed packages into a release. A Ready for Release workflow step can be included in the package workflow. When a package line enters the Ready for Release step, the developer (or other release management user responsible for that package) can select which release they would like to add the package to. The user selects the release and adds the package and its associated package lines to the release. When all of the package lines are confirmed in the Ready for Release step, the package is ready to be used in the release.

Figure 6-1 illustrates the process by which developers can add packages to a release.

Figure 6-1. Ready for release step in workflow


Change Management Package
Package Line 1 ___
Package Line 2 ___
Package Line 3 ___
Package lines are run through package workflow.
Change Management (Package) Workflow
Development Process
Step 1 Package Process
Step 2 Package Process
Step N Package Process
Packages can follow standard development process defined by engineering team.
Release Management Step
Ready For Release
When package line enters the Ready for Release step, that package line can be automatically included in a release.

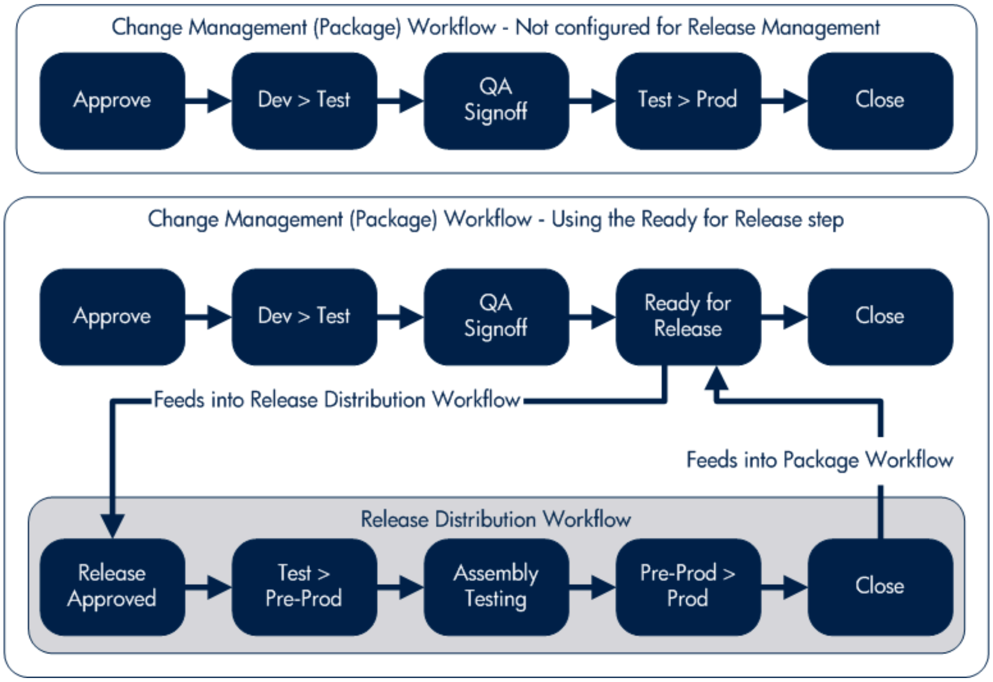
Release Distribution Workflows

Just as the inclusion of appropriate packages and requests is integral to the release definition, so is the process by which the packages are processed in a release distribution. Distribution workflows are used to define the process by which the release's packages are properly tested, approved, and executed against any required environments.

Release distribution workflows need to include package level subworkflows to perform key package level processing. All package line (object type) execution will occur in the subworkflow.

Figure 6-2 illustrates the relationships between packages and release workflows.

Figure 6-2. Role of the distribution workflow

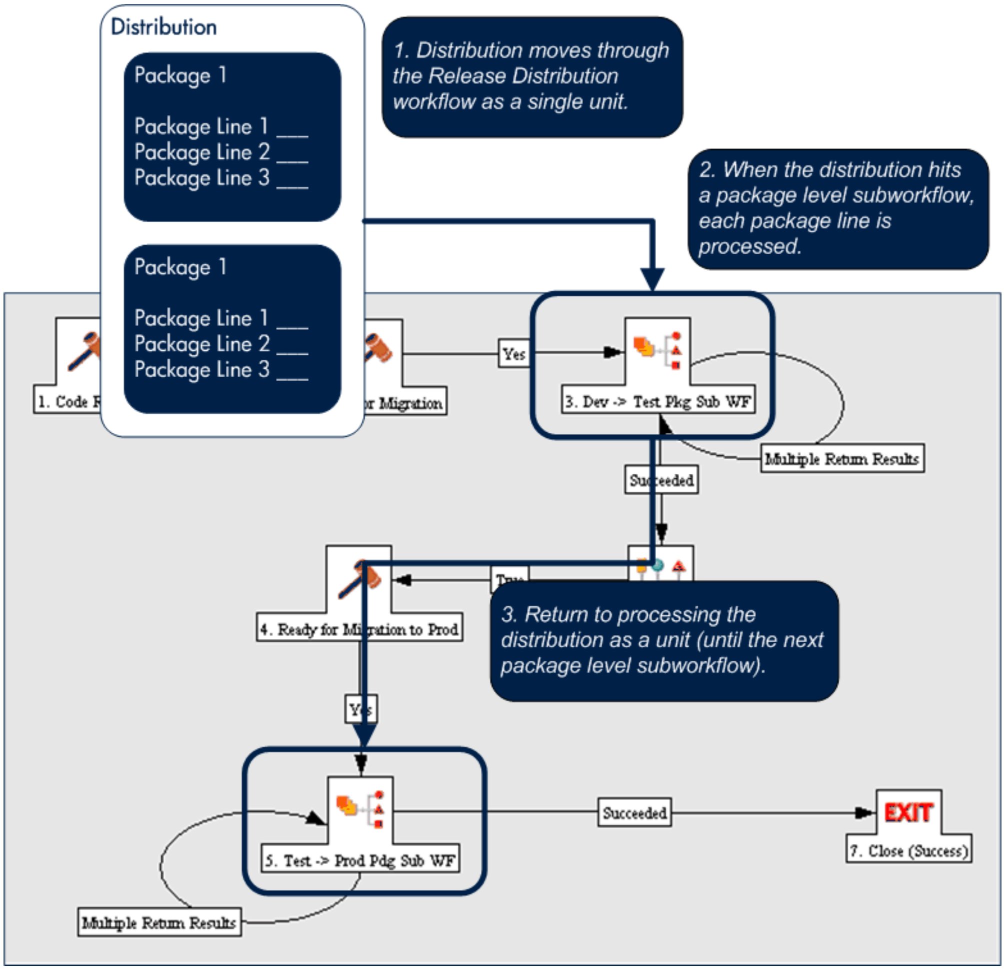

Change Management (Package) Workflow - Not configured for Release Management
Approve
Dev > Test
QA Signoff
Test > Prod
Close
Change Management (Package) Workflow - Using the Ready for Release step
Approve
Dev > Test
QA Signoff
Ready for Release
Close
Feeds into Release Distribution Workflow
Feeds into Package Workflow
Release Distribution Workflow
Release Approved
Test > Pre-Prod
Assembly Testing
Pre-Prod > Prod
Close

The release distribution workflow provides a way in which the release manager can ensure that all files associated with the release deploy properly. As with any workflow, the distribution workflow can be configured to model your existing or best-practice release processes.

Package Level Subworkflows

Release distributions include package level subworkflows, which are used to perform key package level processing. Package level subworkflows are any package subworkflows that have the Use in Release Distributions flag set to **Yes.** All package line (object type) execution occurs in these subworkflows. Also, all package and package line tokens are resolved as these workflows are traversed.

Figure 6-3. Distribution workflow


Distribution
Package 1
Package Line 1 ___
Package Line 2 ___
Package Line 3 ___
Package 1
Package Line 1 ___
Package Line 2 ___
Package Line 3 ___
1. Distribution moves through the Release Distribution workflow as a single unit.
2. When the distribution hits a package level subworkflow, each package line is processed.
3. Return to processing the distribution as a unit (until the next package level subworkflow).
1. Code R
r Migration
Yes
3. Dev -> Test Pkg Sub WF
Multiple Return Results
Succeeded
True
4. Ready for Migration to Prod
Yes
5. Test -> Prod Pdg Sub WF
Succeeded
EXIT
7. Close (Success)
Multiple Return Results

Package level subworkflows are used within release distribution workflows. The release distribution workflow is typically used for release approvals and executing system commands (such as starting or stopping servers). The distribution is processed as a single unit as it proceeds through the release distribution workflow. When the distribution hits a package-level subworkflow, each package line within the distribution is processed. The package subworkflows are used to process package lines and execute object type commands.

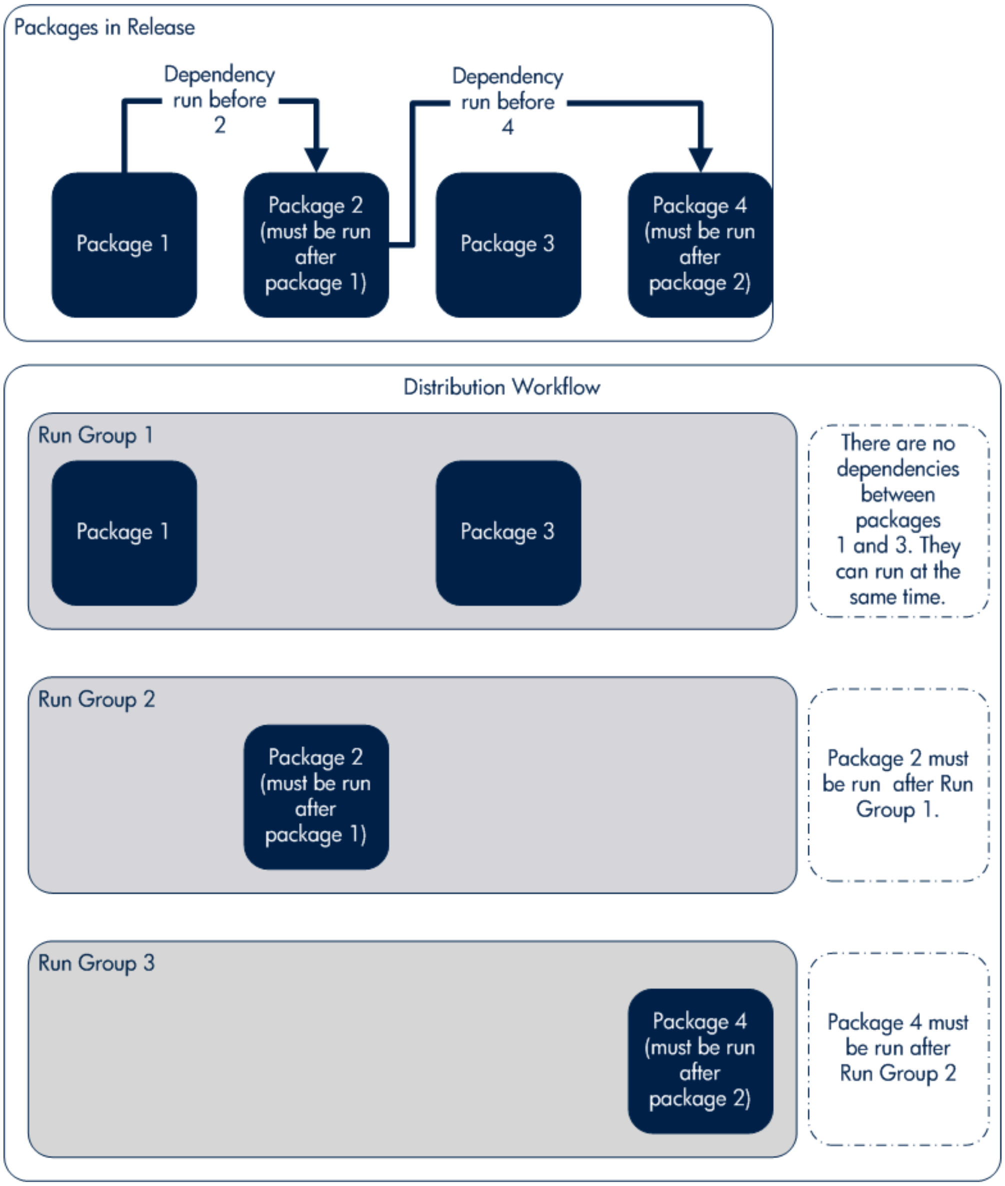
Dependencies and Run Groups

Within a release, the release manager can configure the order in which the packages are processed. The release manager can select certain packages to run before or after other packages in the release. The ordering of packages segregates them into run groups. When a distribution enters an execution step in a package-level subworkflow, all package lines in the first run group will be executed before the package lines in the next run group can begin.

Run groups are automatically determined as you set dependencies between packages. Run groups present an efficient way to process packages which can be run in parallel without having to serially wait for unrelated dependencies. *Figure 6-4* illustrates how package dependencies result in different run groups.

Run groups are automatically determined when a release distribution is created, based on package dependencies specified in the release.

Figure 6-4. Dependencies and run groups


Packages in Release
Dependency run before 2
Dependency run before 4
Package 1
Package 2 (must be run after package 1)
Package 3
Package 4 (must be run after package 2)
Distribution Workflow
Run Group 1
Package 1
Package 3
There are no dependencies between packages 1 and 3. They can run at the same time.
Run Group 2
Package 2 (must be run after package 1)
Package 2 must be run after Run Group 1.
Run Group 3
Package 4 (must be run after package 2)
Package 4 must be run after Run Group 2

Opening Releases

When a release manager first creates a release, only release management users with permission to edit the Release window can add packages. The release manager can enable other users to add packages to the release by clicking **Open Release** in the Release window. By creating an open release, developers processing a Ready for Release workflow step will have the option of adding the package to that release. If a release is not opened, it is not displayed in the list used for that Ready for Release step.

Submitting Releases

When a release manager submits a release, the release enters a code freeze state. In this state, packages cannot be added or removed from the release by anyone other than the release manager. When the release is submitted, a distribution is automatically created. You can then process the distribution or cancel it and create a new one later.

Overview of Using Release Management - Process

Release management introduces repeatable, reliable processes surrounding software and application releases. Release management provides an interface for grouping and processing the packages and requests associated with a specific release.

Release Management Pre-Configuration

Planning for an application or software release should begin immediately upon recognition that a release is pending. Before the release manager creates a release, it is often necessary to pre-configure the following in release management:

1. Modify package workflows to include the Ready for Release workflow step.

2. Create all required distribution workflows (including all package-level subworkflows).

By adding the Ready for Release workflow step to workflows, you provide developers with the ability to add a required package to a release at the appropriate time. Development package workflows will typically address necessary approval and execution steps directly related to that package. The Ready for Release workflow step indicates that the developer has signed off on the package and the package is ready to be integrated and shipped with other packages related to the release.

The release manager should also create the distribution workflows. This includes defining any subworkflows (package level or distribution level) that will be used in the release distribution workflow. These workflows define the process by which the release's packages are properly tested, approved, and executed against any required environments. The release distribution workflow and associated subworkflows ensure that all files associated with the release are properly deployed. As with any workflow, the distribution workflow can be configured to model your existing or best-practice release processes.

Creating Releases

To create a release in release management:

1. Create a new release in the Release window.
2. Open the release by clicking **Open Release.**

   This allows developers working in the Deployment Management Package window to add packages to the release via the Ready for Release workflow step.
3. Add packages and requests to the release.

   Packages can be added through the Ready for Release workflow step in the Packages window or directly by the release manager through the Release window.
4. Click **Dependencies** in the **Package** tab to set package dependencies.
5. Configure dependencies between packages in the release.
6. Verify the release.

Processing Releases

When the appropriate data is collected in the release (packages, requests and dependencies) and the appropriate workflows have been created, the release manager can then process the release.

To process a release the release manager will:

1. Submit the release.

    a. Create a distribution. This consists of selecting a workflow for the distribution and disabling any package lines that should not run with that distribution.

    b. Submit the distribution.

2. Send feedback to packages. Send feedback to the packages at any time from the Distribution window. Select the value from the feedback drop-down list and click **Feedback.** This value is sent back to the package line in the Ready for Release workflow step and is used to transition out of that step.

3. Close the release only if there is no need to create additional distributions for use at a later date.

    - When the release is closed, any in-progress distributions are cancelled.

    - If a distribution is not submitted, it is not cancelled when the release is closed. However, after the release is closed the distribution cannot be edited.

Distributions

A distribution is a deployment of a release. In a distribution, the release manager specifies which workflow will control the release process and which of the release's packages will be included.

For example, a software company has a product update release scheduled five months from now. As a part of their release process, they need to update their testing, production, and training instances of the product. The processes required for delivering the product to these different environments differs in the following ways:

- Not all of the packages in the release need to be applied to each instance, such as the training instance requires custom code to establish additional product security which is not required in the production instance.

- There is a different review process for each instance, such as the testing instance does not require the department head sign-off for each iteration of the release.

The software company creates a distribution for each of these release instances. For each distribution the release manager defines:

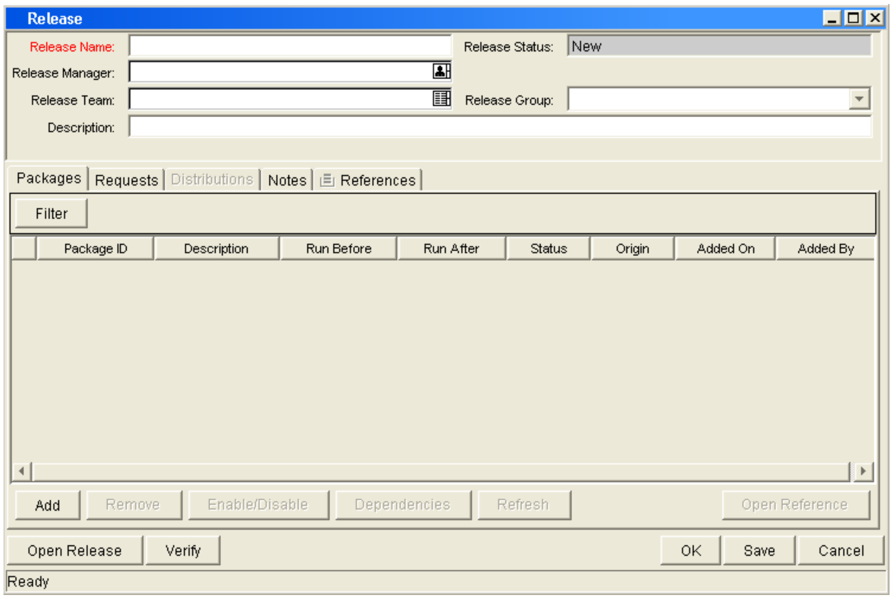
- Which packages are included in the specific release instance.
- Which workflow the release follows.

Overview of Configuring Releases

Setting up a successful software or application release requires a comprehensive view of the release process. Release management provides the tools for capturing the entire release process. See *Overview of Using Release Management - Process* on page 174 for an overview of the items and processes involved in creating a release.

One of the first steps in establishing a controlled release is to create a new release in the Release Workbench.

Figure 6-5. Release window



Release Status: New

Ready

The following lists the major section found in the Release window:

- **General information.** General information includes basic information concerning the environment, such as the environment name and description. See *Creating Releases* on page 179.
- **Packages.** The **Packages** tab allows packages to be added to the release. See *Adding Packages to Releases* on page 180.
- **Requests.** The **Requests** tab allows requests to be added to the release. See *Adding Requests to Releases* on page 185.
- **Distributions.** The **Distributions** tab creates distributions for releases. See *Creating Distributions* on page 188.
- **Notes.** The **Notes** tab lets you add notes to a release.
- **References.** The **References** tab tracks information regarding the release.

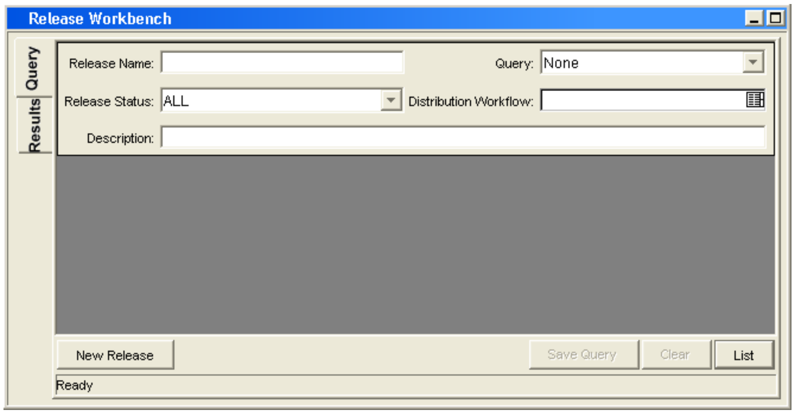
Opening the Release Workbench

To open the Release Workbench:

1. Log on to PPM Center.
2. From the menu bar, select **Administration > Open Workbench.**

   The PPM Workbench opens.
3. From the shortcut menu, select **Deployment Mgmt > Releases.**

   The Release Workbench opens.



Query: None

Release Status: ALL

Creating Releases

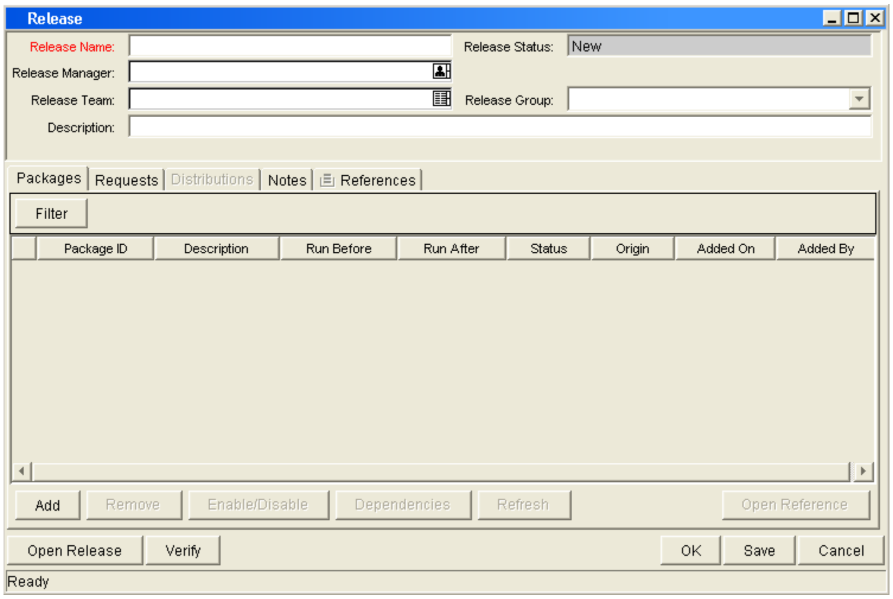
To configure general information and create a release:

1. From the Workbench shortcut bar, select **Deployment Mgmt > Releases.**

   The Release Workbench opens.

2. Click **New Release.**

   The Release window opens.



Release Status: New

3. Complete the fields as specified in the following table:

Field Name	Description
Release Name	The name of the release.
Release Status	The current status of the Release. ■ **New** ■ **Open** ■ **Code Freeze** ■ **Closed**
Release Manager	The name of the release management user who has control over the particular release. Only release management users who can manage releases are listed.

Field Name	Description
Release Team	The users who have access to the particular release. This is a validated list of security groups and is used for informational purposes only.
Release Group	A generic grouping of releases which allows the release manager to group releases into logical categories such as customization.
Description	The description of the release.

4. Add packages to the release.

   For information on how to add packages, see *Adding Packages to Releases* on page 180.

5. Add requests to the release.

   For information on how to add requests, see *Adding Requests to Releases* on page 185.

6. To allow other release management users to add packages and requests to the release, click **Open Release.**

7. Click **OK.**

Adding Packages to Releases

You can add packages to release in one of the following ways.

Adding Packages Through the Release Window

When defining a release in the Release window, the release manager can decide to manually add packages to the release.

To manually add a package to a release:

1. From the Workbench shortcut bar, select **Deployment Mgmt > Releases.**

   The Release Workbench opens.

2. Open a release.

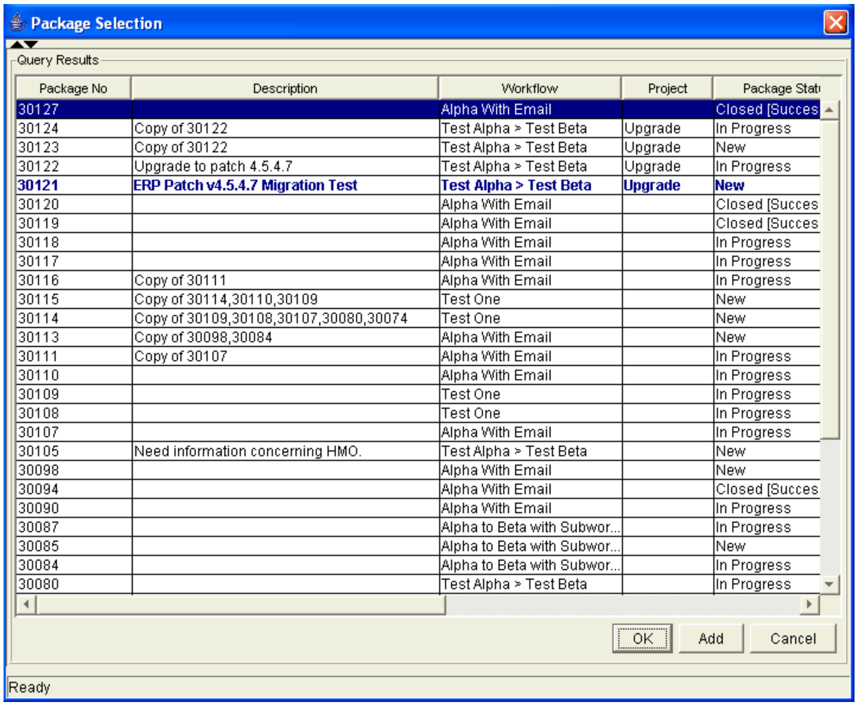
   The Release window opens to the **Packages** tab.

3. Click **Add.**

   The Package Selection window opens.

4. Specify your search criteria, and then click **List.**

   The **Query Results** field displays the packages that match the search criteria.



Package No	Description	Workflow	Project	Package Stat
30127		Alpha With Email		Closed [Succes
30124	Copy of 30122	Test Alpha > Test Beta	Upgrade	In Progress
30123	Copy of 30122	Test Alpha > Test Beta	Upgrade	New
30122	Upgrade to patch 4.5.4.7	Test Alpha > Test Beta	Upgrade	In Progress
30121	**ERP Patch v4.5.4.7 Migration Test**	**Test Alpha > Test Beta**	**Upgrade**	**New**
30120		Alpha With Email		Closed [Succes
30119		Alpha With Email		Closed [Succes
30118		Alpha With Email		In Progress
30117		Alpha With Email		In Progress
30116	Copy of 30111	Alpha With Email		In Progress
30115	Copy of 30114,30110,30109	Test One		New
30114	Copy of 30109,30108,30107,30080,30074	Test One		New
30113	Copy of 30098,30084	Alpha With Email		New
30111	Copy of 30107	Alpha With Email		In Progress
30110		Alpha With Email		In Progress
30109		Test One		In Progress
30108		Test One		In Progress
30107		Alpha With Email		In Progress
30105	Need information concerning HMO.	Test Alpha > Test Beta		New
30098		Alpha With Email		New
30094		Alpha With Email		Closed [Succes
30090		Alpha With Email		In Progress
30087		Alpha to Beta with Subwor...		In Progress
30085		Alpha to Beta with Subwor...		New
30084		Alpha to Beta with Subwor...		In Progress
30080		Test Alpha > Test Beta		In Progress

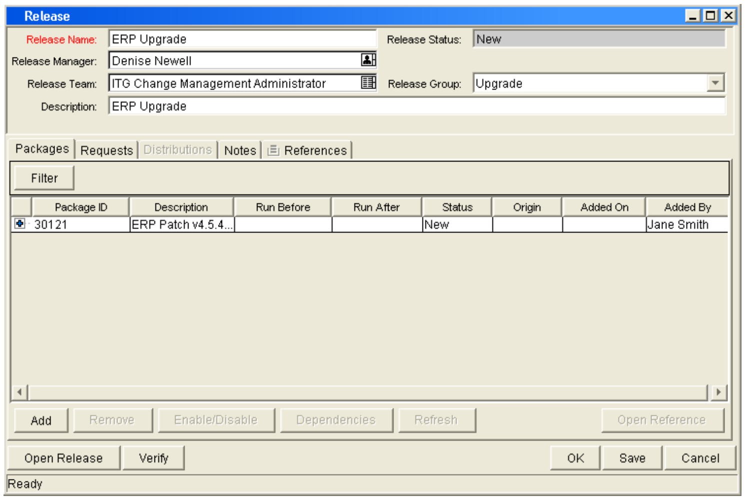
5. In the **Query Results** field, select the packages to add to the release.

6. Click **Add.**

   If the packages reference any other packages or requests, you are prompted to indicate whether to include or exclude them.

7. Click **Close.**

   The Release window now lists the packages you added.



Release Name: ERP Upgrade

Release Status: New

Release Manager: Denise Newell

Release Team: ITG Change Management Administrator

Release Group: Upgrade

Description: ERP Upgrade

Package ID	Description	Run Before	Run After	Status	Origin	Added On	Added By
30121	ERP Patch v4.5.4...			New			Jane Smith

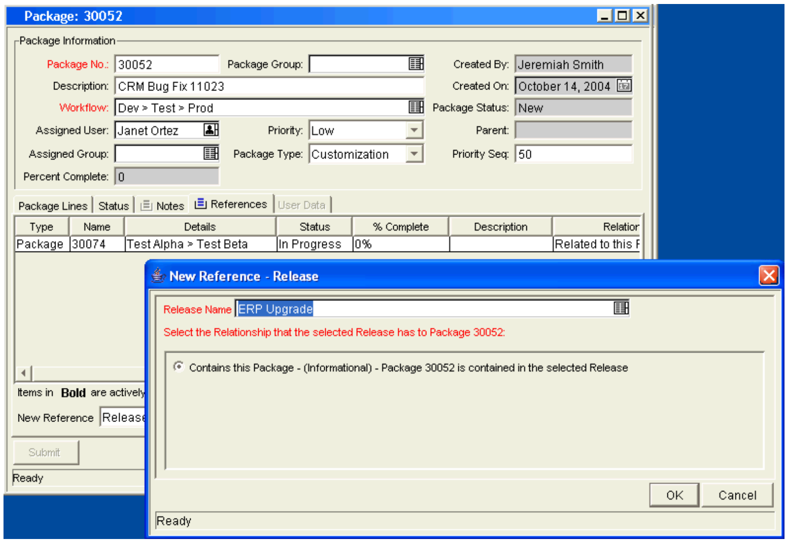
8. Click **Save.**

Adding Packages Through the Package Window

To add packages to a release from the Package window, you reference the release on the package **References** tab.

The References window is shown in *Figure 6-6*.

Figure 6-6. Package added to a release through the package reference



Package No.: 30052

Package Group:

Created By: Jeremiah Smith

Description: CRM Bug Fix 11023

Created On: October 14, 2004

Workflow: Dev > Test > Prod

Package Status: New

Assigned User: Janet Ortez

Priority: Low

Parent:

Assigned Group:

Package Type: Customization

Priority Seq: 50

Percent Complete: 0

Type	Name	Details	Status	% Complete	Description	Relatior
Package	30074	Test Alpha > Test Beta	In Progress	0%		Related to this F

Release Name: ERP Upgrade

Select the Relationship that the selected Release has to Package 30052:

● Contains this Package - (Informational) - Package 30052 is contained in the selected Release

Adding Packages by the Ready for Release Workflow Step

Release management provides a Ready for Release workflow step source which can add significant value to your release process. When a package line reaches the Ready for Release workflow step and is executed, the status of the package line changes to Confirmed. As soon as all of the lines in a package reach this status, the entire package status changes to Ready for Release. The release distribution can then feed back to each of its associated packages so that each package can progress to the next workflow step.

To reject the Ready for Release workflow step, select **Bypass Execution** or **Override Status.** This stops package release, but allows the package to continue through its own workflow.

To act on a Ready for Release workflow step:

1. From the Workbench shortcut bar, select **Deployment Mgmt > Packages.**

   The Package Workbench opens.

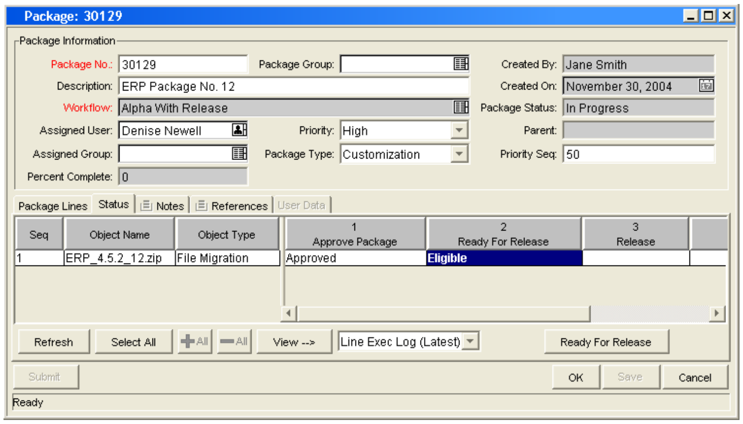
2. Open a package.

   The Package window opens.

3. Click the **Status** tab.

4. Select the workflow step to act on.

   Workflows that are available for action are displayed in bold text. After you select an available workflow step, the **Action** button label changes to the workflow step name.



Package: 30129

Package Information

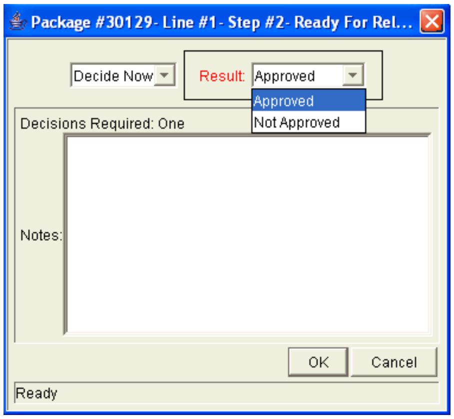
Package No.:	30129	Package Group:		Created By:	Jane Smith
Description:	ERP Package No. 12			Created On:	November 30, 2004
Workflow:	Alpha With Release			Package Status:	In Progress
Assigned User:	Denise Newell	Priority:	High	Parent:	
Assigned Group:		Package Type:	Customization	Priority Seq:	50
Percent Complete:	0				

Seq	Object Name	Object Type	1 Approve Package	2 Ready For Release	3 Release
1	ERP_4.5.2_12.zip	File Migration	Approved	**Eligible**	

Ready

5. Click **Ready for Release.**

   The Package Action window opens.



6. In the **Results** field, select the workflow step result.

   You can also enter any relevant notes in this window.

7. Use the **Add to Release** field to select the release to associate with the package.

   This field may be required, depending on the workflow step configuration.

8. Click **OK.**

9. In the Package window, click **OK.**

   The package is now ready for release.

Adding Packages from Requests

When a request that is included in a release spawns a package, that package is included in the release automatically. This becomes a powerful method for including packages in a release.

For example, a development manager can include a request to fix a software bug in the release. You can configure that request workflow to automatically create a package to migrate changes into production. That package is then included in the release.

Adding Requests to Releases

You can add requests to a release to track information associated with the release. For example, the release manager may want to track which software bugs or enhancements (captured using a request) were implemented during the release timeframe.

Adding Requests Through the Release Window

To add a request to a release from the Release window:

1. From the Workbench shortcut bar, select **Deployment Mgmt > Releases.**

   The Release Workbench opens.

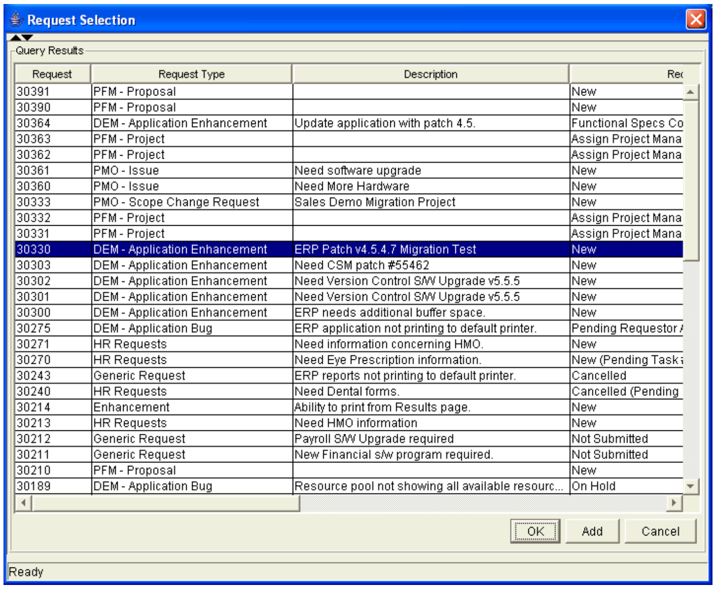
2. Open a release.

   The Release window opens.

3. Click the **Requests** tab.

4. Click **Add.**

   The Request Selection window opens.



Request	Request Type	Description	Re(
30391	PFM - Proposal		New
30390	PFM - Proposal		New
30364	DEM - Application Enhancement	Update application with patch 4.5.	Functional Specs Co
30363	PFM - Project		Assign Project Mana
30362	PFM - Project		Assign Project Mana
30361	PMO - Issue	Need software upgrade	New
30360	PMO - Issue	Need More Hardware	New
30333	PMO - Scope Change Request	Sales Demo Migration Project	New
30332	PFM - Project		Assign Project Mana
30331	PFM - Project		Assign Project Mana
30330	DEM - Application Enhancement	ERP Patch v4.5.4.7 Migration Test	New
30303	DEM - Application Enhancement	Need CSM patch #55462	New
30302	DEM - Application Enhancement	Need Version Control S/W Upgrade v5.5.5	New
30301	DEM - Application Enhancement	Need Version Control S/W Upgrade v5.5.5	New
30300	DEM - Application Enhancement	ERP needs additional buffer space.	New
30275	DEM - Application Bug	ERP application not printing to default printer.	Pending Requestor /
30271	HR Requests	Need information concerning HMO.	New
30270	HR Requests	Need Eye Prescription information.	New (Pending Task
30243	Generic Request	ERP reports not printing to default printer.	Cancelled
30240	HR Requests	Need Dental forms.	Cancelled (Pending
30214	Enhancement	Ability to print from Results page.	New
30213	HR Requests	Need HMO information	New
30212	Generic Request	Payroll S/W Upgrade required	Not Submitted
30211	Generic Request	New Financial s/w program required.	Not Submitted
30210	PFM - Proposal		New
30189	DEM - Application Bug	Resource pool not showing all available resourc...	On Hold

5. Enter search criteria, and then click **List.**

6. In the **Query Results** field, select the request to add to the release.

7. Click **Add.**

   If there are any referenced entities, you are prompted to indicate whether to include or exclude them.

8. Click **Close.**

9. In the Release window, click **Save.**

Adding Requests Through the Requests Window

Users can only add requests to an open release. Once the release reaches the code freeze or closed state, requests can only be added through the Release window by the release manager.

To add a request to a release:

1. From the PPM Dashboard, open a request.

2. In the request, select the References section.

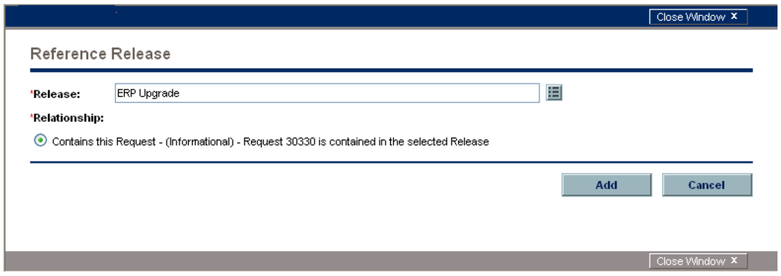
3. In the **New Reference** field, select **Release.**

4. Click **Add.**

   The Reference Release window opens.

5. Use the **Release** field to select a release.

   The **Release** field displays only the releases that the release manager has opened.



Reference Release

*Release: ERP Upgrade

*Relationship:

● Contains this Request - (Informational) - Request 30330 is contained in the selected Release

Verifying Releases

After you assemble a release (packages, requests, and dependencies), verify that the release is properly configured.

To verify your release:

1. From the Workbench shortcut bar, select **Deployment Mgmt > Releases.**

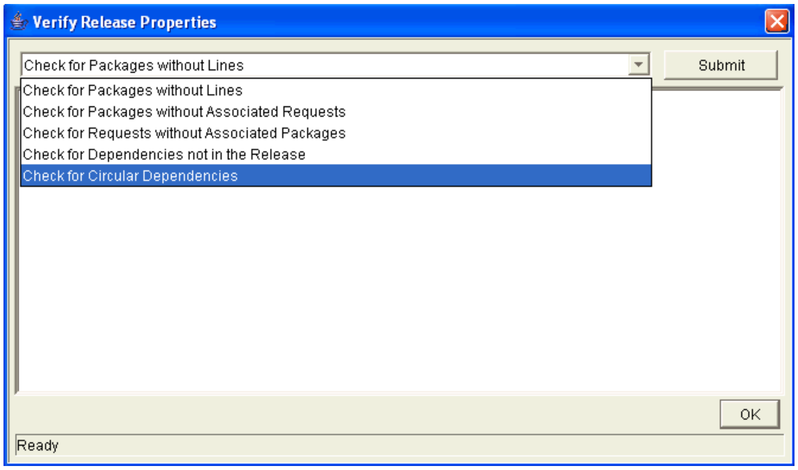
   The Release Workbench opens.

2. Open a release.

   The Release window opens to the **Packages** tab.

3. Click **Verify.**

   The Verify Release Properties window opens.


Verify Release Properties
Check for Packages without Lines
Submit
Check for Packages without Lines
Check for Packages without Associated Requests
Check for Requests without Associated Packages
Check for Dependencies not in the Release
Check for Circular Dependencies
OK
Ready

4. In the list at the top of the window, select one of the following options:

   - **Check for packages without lines**
   - **Check for packages without associated requests**
   - **Check for packages without associated packages**
   - **Check for dependencies not in the release**
   - **Check for circular dependencies**

5. Click **Submit.**

   Any errors are reported in the window.

6. After you examine the results, click **OK.**

Creating Distributions

A distribution is a deployment of a release. In a distribution, the release manager specifies the workflow to control the release process and specifies which release packages to include.

To create a distribution:

1. From the Workbench shortcut bar, select **Deployment Mgmt > Releases.**

   The Release Workbench opens.

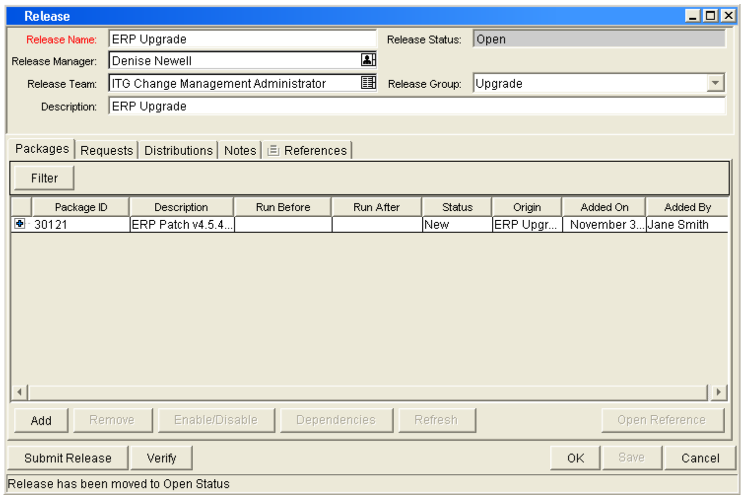
2. Open a release.

   The Release window opens.

3. Click **Open Release.**

   This enables the **Distributions** tab.

4. Click the **Distributions** tab.



Release

Release Name: ERP Upgrade

Release Status: Open

Release Manager: Denise Newell

Release Team: ITG Change Management Administrator

Release Group: Upgrade

Description: ERP Upgrade

	Package ID	Description	Run Before	Run After	Status	Origin	Added On	Added By
⊞	30121	ERP Patch v4.5.4...			New	ERP Upgr...	November 3...	Jane Smith

Release has been moved to Open Status

5. Click **New.**

   The Distribution window opens.

6. Enter the new distribution name and description.

7. Select the workflow for this distribution to follow.

8. Select any packages that you want to disable, and then click **Enable/ Disable.**

   The names of disabled packages are displayed in italic text.

9. Click **Submit Release.**

   The distribution runs through the workflow specified. The release begins to run through the assigned workflow.

Enabling/Disabling Package Lines in a Distribution

To disable a package line in a distribution:

1. From the Workbench shortcut bar, select **Deployment Mgmt > Releases.**

   The Release Workbench opens.

2. Open a release.

   The Release window opens.

3. Click the **Distributions** tab.

4. Click the **Package Status** tab.

5. Locate the package line that you want to disable.

6. To display all of the packages, click **Run Groups.**

   Note that you may have to change your filter to see the package you want.

7. Select the package line to disable.

   You can disable an entire package. Disabling a package line in an active run group (within a package-level subworkflow) cancels the package line.

8. Click **Disable.**

   The name of the disabled package line is displayed in italic text.

If you disable a package line in an active run group, you cannot re-enable the package line until the run group completes. If the disabled package line was not in an active run group, you can re-enable it immediately.

Running Distributions through a Workflow

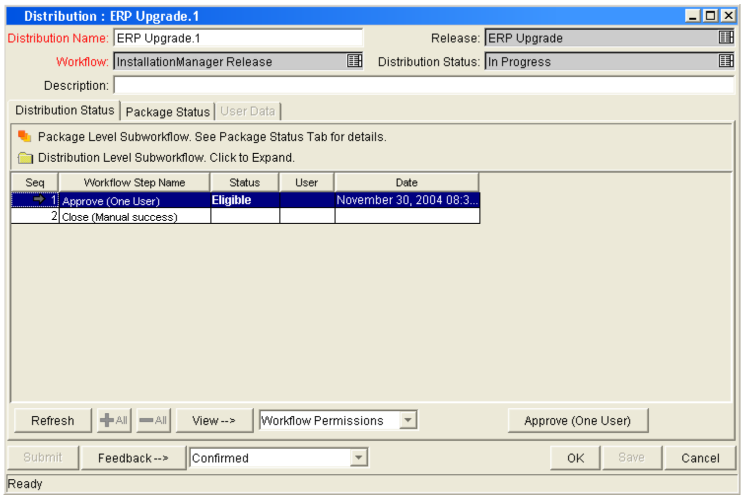
The last step involved in creating a release is running the release through a Deployment Management workflow. This involves running any decisions, commands, token evaluations, or other tasks for the entire distribution.

Processing the distribution requires that you process steps on both the **Distribution Status** and **Package Status** tabs.

Processing Distribution Steps

Active workflow steps appear in bold text. Select the active line and click **Action** to process that workflow step. On the **Distribution Status** tab, you can expand and act on all distribution steps (including distribution-level subworkflow steps). To process package lines, you must use the **Package Status** tab.

Figure 6-7. Distribution Status tab



Distribution : ERP Upgrade.1

Distribution Name: ERP Upgrade.1

Release: ERP Upgrade

Workflow: InstallationManager Release

Distribution Status: In Progress

Description:

Package Level Subworkflow. See Package Status Tab for details.

Distribution Level Subworkflow. Click to Expand.

Seq	Workflow Step Name	Status	User	Date
1	Approve (One User)	**Eligible**		November 30, 2004 08:3...
2	Close (Manual success)			

Workflow Permissions

Confirmed

Ready

Processing Package Lines

Package lines can be processed individually or in groups. Package lines that are available for your action appear in bold text. Select an active package line and click **Action** to process that individual workflow step.

Release management provides a convenient interface for processing groups of package lines (in the same workflow step) simultaneously. This is done by viewing and selecting the package statuses.

To select all package lines within a workflow step of a particular status:

1. From the Workbench shortcut bar, select **Deployment Mgmt > Releases.**

   The Release Workbench opens.

2. Open a release.

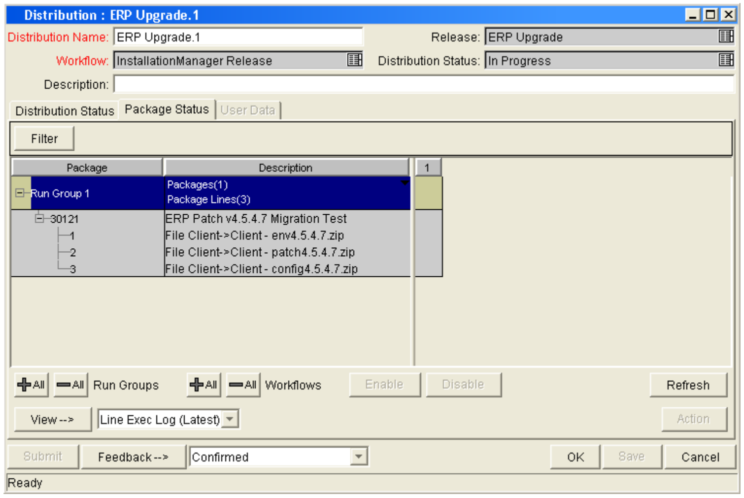
   The Release window opens.

3. Select the **Distributions** tab.

4. Click the **Package Status** tab.

5. Expand all run groups and workflows.

6. To display a status summary of all the package lines in each workflow step, in the Description column, click the plus character (+).



Distribution : ERP Upgrade.1

Distribution Name: ERP Upgrade.1
Release: ERP Upgrade
Workflow: InstallationManager Release
Distribution Status: In Progress
Description:

Package	Description	1
⊟ Run Group 1	Packages(1) Package Lines(3)	
⊟ 30121	ERP Patch v4.5.4.7 Migration Test	
1	File Client->Client - env4.5.4.7.zip	
2	File Client->Client - patch4.5.4.7.zip	
3	File Client->Client - config4.5.4.7.zip	

View --> Line Exec Log (Latest)
Feedback --> Confirmed

Ready

7. Select the summary.

   Again, items that are available for your action appear in bold text. When you select the summary, all package lines in that state are automatically selected. You can deselect individual items using `Ctrl + click`.

8. Click **Action** to process all of the selected package lines.

9. Proceed to the next workflow step in the package process or distribution process (depending on your preconfigured process).

   To view updates in the **Distribution Status** tab, click in the **Distribution Status**, and then click **Refresh.**

Completing Distributions

When the distribution completes and the workflow closes, a value can be returned to the Ready for Release workflow steps. Those packages can then continue to process based on that validation. That value is sent by clicking **Feedback** in the Distribution window.

7 Configuring Environments

In This Chapter:

Overview of Environments

When migrating objects, HP Deployment Management logs onto remote computers in the same way any other user would (using FTP, SCP, SSH, or Telnet). HP Deployment Management can log on using any existing username and password. However, HP recommends that you generate a new user on each computer that HP Deployment Management is to access. This helps to clarify the setup and relieve some administrative burden.

Make sure that you have full access to the <*PPM_Home*> directory on the PPM Server as well as the correct read and write permissions on other required directories. In addition, on Windows® servers, the Administrators group must have read access to the PPM Server home directory. Any Windows server that HP Deployment Management will access should have been configured as described in *System Administration Guide and Reference*.

Environment Connection Protocols

The communication protocol that will be used to connect to the server or client must be specified in the environment. This protocol will be used by commands to connect to source and destination environments in the deployment system. Work with the application administrator to determine which connection protocols are supported at your site for the machines housing the deployment environments.

HP Deployment Management supports the following connection protocols:

- Telnet
- SSH
- SSH2

Environment Transfer Protocols

The transfer protocol that will be used to transfer files to the server or client must be specified in the environment.

HP Deployment Management supports the following transfer protocols:

- FTP
- FTP (active)
- FTP (passive)
- Secure Copy
- Secure Copy 2

Transfer Protocol Configuration Notes

Choose the transfer protocol best suited to the business and technology needs. Consider factors related to security and performance when selecting the transfer protocol. Work with the application administrator to determine which connection protocols are supported for the machines housing the deployment environments. The following list provides some suggestions for when to use the above protocols.

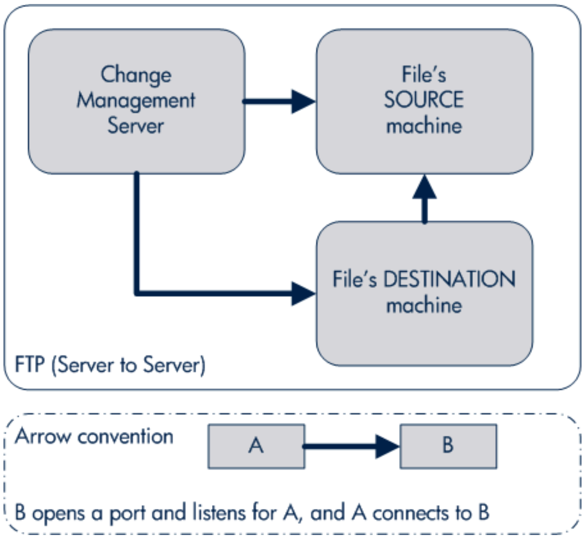
No additional product configuration is required to enable one FTP mode over another. Application administrators need to consider their FTP server configuration (particularly as they relate to security and firewall settings) when selecting an FTP protocol for transferring data.

Selecting the FTP Protocol

The following capabilities must be enabled on the source and destination machines for the following FTP protocol selection to function properly.

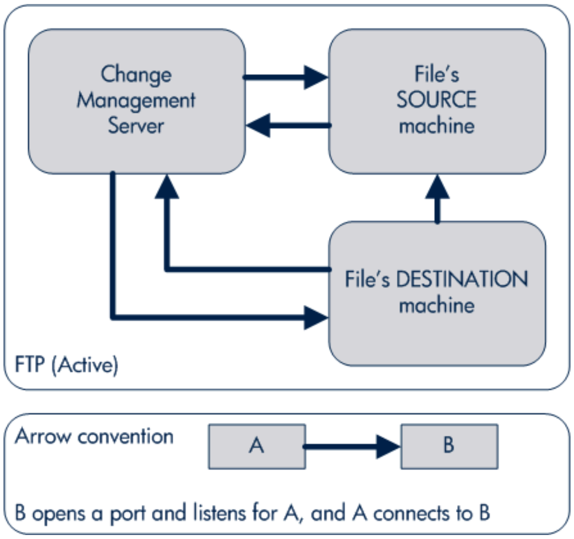
- FTP (server to server). See *Figure 7-1*.
    - Either the source or the destination environment needs to allow outgoing connections to a third party
    - FTP PORT command must be enabled on one of the environments
    - FTP PASV command must be enabled on the other environment

Figure 7-1. FTP (server to server)



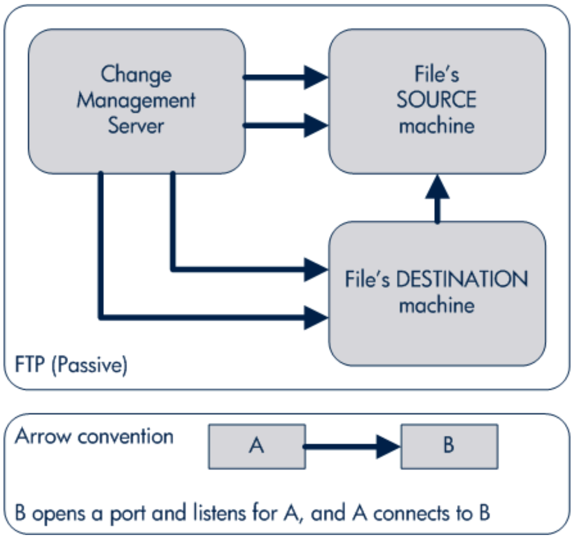
- FTP (active). See *Figure 7-2*.
  - PORT command must be enabled on both the source and destination environments (allows outgoing back to the requestor)

Figure 7-2. FTP (active)


Change Management Server
File's SOURCE machine
File's DESTINATION machine
FTP (Active)
Arrow convention
A
B
B opens a port and listens for A, and A connects to B

- FTP (passive). See *Figure 7-3*.
    - PASV must be enabled on both the source and destination environments. In this configuration, the HP Deployment Management server sends a command to the source or destination instructing that environment to open a port. The HP Deployment Management server then connects to that port.

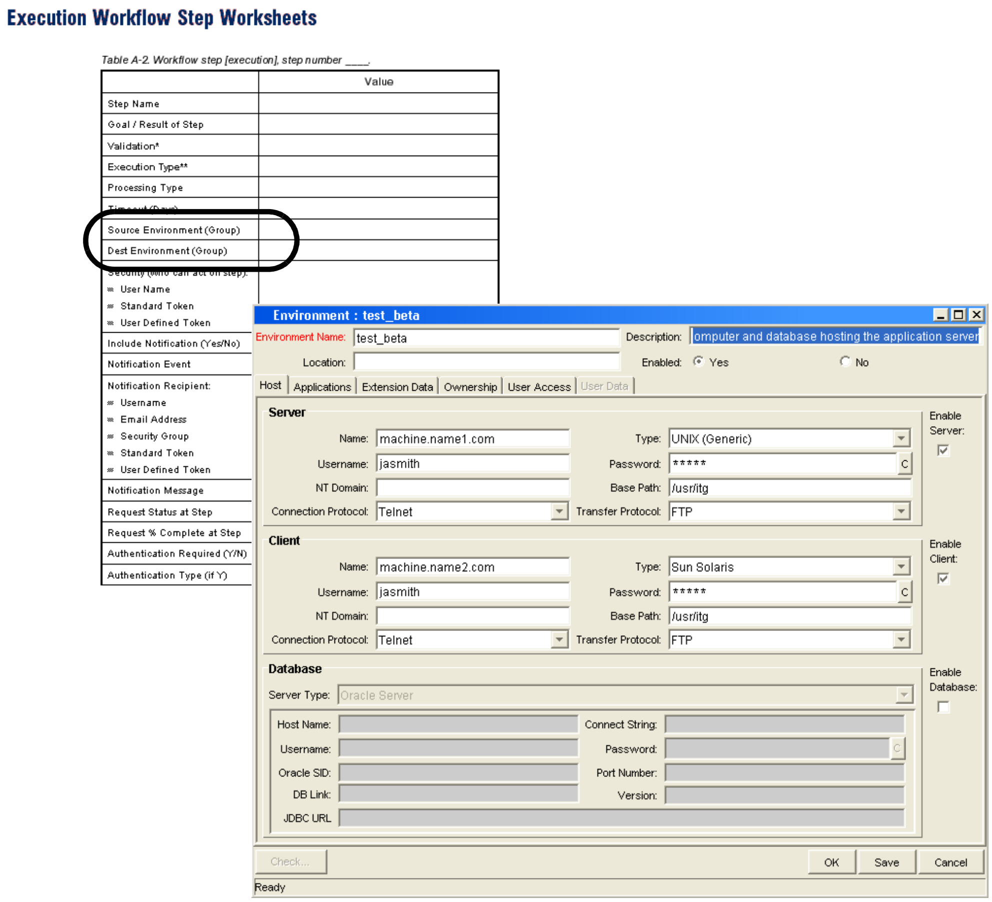
Figure 7-3. FTP (passive)


Change Management Server
File's SOURCE machine
File's DESTINATION machine
FTP (Passive)
Arrow convention
A
B
B opens a port and listens for A, and A connects to B

Overview of Configuring Environments

Environments are configured in the Environments window. Some of the information entered on the Environments window can be gathered from the appropriate Workflow Step Worksheets (see *Figure 7-4*).

Figure 7-4. Worksheet and Environments window



Execution Workflow Step Worksheets

Table A-2. Workflow step [execution], step number ____.

	Value
Step Name	
Goal / Result of Step	
Validation*	
Execution Type**	
Processing Type	
Timeout (Days)	
Source Environment (Group)	
Dest Environment (Group)	
Security (Who can act on step): User Name Standard Token User Defined Token	
Include Notification (Yes/No)	
Notification Event	
Notification Recipient: Username Email Address Security Group Standard Token User Defined Token	
Notification Message	
Request Status at Step	
Request % Complete at Step	
Authentication Required (Y/N)	
Authentication Type (if Y)	

Environment : test_beta

Environment Name: test_beta
Description: omputer and database hosting the application server
Location:
Enabled: ● Yes ○ No

Host | Applications | Extension Data | Ownership | User Access | User Data

Server

Name:	machine.name1.com	Type:	UNIX (Generic)
Username:	jasmith	Password:	*****
NT Domain:		Base Path:	/usr/itg
Connection Protocol:	Telnet	Transfer Protocol:	FTP

Enable Server: ☑

Client

Name:	machine.name2.com	Type:	Sun Solaris
Username:	jasmith	Password:	*****
NT Domain:		Base Path:	/usr/itg
Connection Protocol:	Telnet	Transfer Protocol:	FTP

Enable Client: ☑

Database

Server Type: Oracle Server

Host Name:		Connect String:	
Username:		Password:	
Oracle SID:		Port Number:	
DB Link:		Version:	
JDBC URL			

Enable Database: □

The following lists the major section found in the Environments window:

- **General information.** General information includes basic information concerning the environment, such as the environment name and description. See *Configuring General Information for Environments* on page 200.

- **Host.** The **Host** tab defines basic information about the client, server, and database for the environment. The fields for the client and server sections are identical. See *Creating Environments* on page 200.

- **Applications.** Every HP Deployment Management environment can contain its own set of applications. See *Using Application Codes Environments* on page 204.

- **Extension Data.** HP Deployment Management Extensions requires specially configured environments. If HP Deployment Management Extensions products are not installed, the **Extension Data** tab is disabled.

- **Ownership.** Configure who can edit the environment. See *Setting Ownership and Participants for Environments* on page 209.

- **User Access.** Configures participants of the environment. Participants can then be given specific access rights to the environment. See *Adding Participants to Environments* on page 210.

- **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.

Opening the Environments Workbench

To open the Environments Workbench:

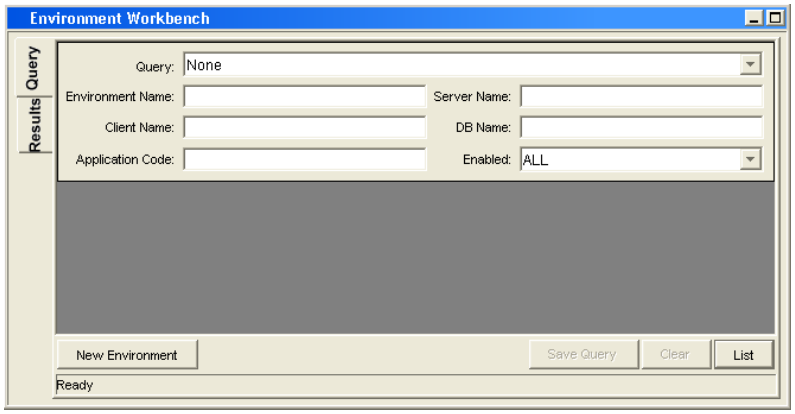
1. Log on to PPM Center.

2. From the menu bar, select **Administration > Open Workbench.**

   The PPM Workbench opens.

3. From the shortcut bar, select **Environments > Environments.**

   The Environments Workbench opens.



Query: None

Enabled: ALL

Ready

Configuring General Information for Environments

To configure the general information of an environment:

1. From the Workbench shortcut bar, select **Environments > Environments.**

   The Environments Workbench opens.

2. Open an environment.

   The Environment window opens.

3. Complete the fields as specified in the following table:

Field Name	Description
Environment Name	The name of the environment.
Location	The location of the environment. For example, In the server room.
Description	A brief description of how the environment is being used.
Enabled	To make the environment available to the system, select **Yes.**

4. Click **OK.**

Creating Environments

To define a new environment:

1. From the Workbench shortcut bar, select **Environments > Environments.**

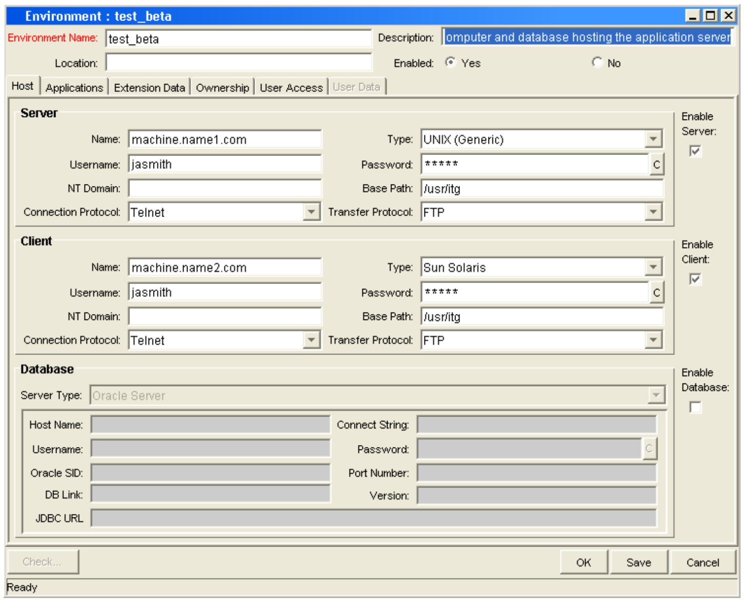
   The Environments Workbench opens.

2. Open an environment.

   The Environment window opens to the **Hosts** tab. The **Hosts** tab has the following three sections:

   - **Server**
   - **Client**
   - **Database**

t


Environment : test_beta
Host | Applications | Extension Data | Ownership | User Access | User Data
Check...
OK
Save
Cancel

Environment Name: test_beta
Description: omputer and database hosting the application server
Location:
Enabled: ● Yes ○ No

Server

Name: machine.name1.com
Type: UNIX (Generic)
Username: jasmith
Password: *****
NT Domain:
Base Path: /usr/itg
Connection Protocol: Telnet
Transfer Protocol: FTP
Enable Server: ☑

Client

Name: machine.name2.com
Type: Sun Solaris
Username: jasmith
Password: *****
NT Domain:
Base Path: /usr/itg
Connection Protocol: Telnet
Transfer Protocol: FTP
Enable Client: ☑

Database

Server Type: Oracle Server
Enable Database: □
Host Name:
Connect String:
Username:
Password:
Oracle SID:
Port Number:
DB Link:
Version:
JDBC URL

Ready

3. Complete the fields in the **Server** section as specified in the following table:

Field Name	Description
Name	The DNS name or IP address of the computer.
Type	A list of supported server/client operating systems. Should be set to the operating system for the computer defined in the **Name** field.
Username	The username that HP Deployment Management uses to log on to the server/client to transfer files or execute commands. if that user account was generated on the local machine, this is most likely “HP PPM.”
Password	The password for the given username. The password is hidden. You can change it by clicking the button to the right of the field.
NT Domain	The domain name to use if this is a Window Server.
Base Path	The path for applications on this computer. In many instances, this is the home directory of the defined username. When HP Deployment Management transfers a file or executes a command on this server, it logs in and changes directories to this base path before proceeding. Use forward slash characters (/) as directory separators, even for Windows systems.

Field Name	Description
Connection Protocol	The protocol to be used for server or client connections. SSH is a secure connections protocol, whereas Telnet is not. In order to use SSH as your connection protocol, you must first set up SSH on the destination machine.
Transfer Protocol	The protocol to be used for moving files between various clients and servers. SCP is a secure transfer protocol, whereas FTP is not. In order to use SCP as your transfer protocol, you must first set up SCP on the destination machine.
Enable Server	Make the server information available to the system. Select the checkbox to make the server information available to the system.

4. Complete the fields in the **Client** section as specified in the following table:

Field Name	Description
Name	The DNS name or IP address of the computer.
Type	A list of supported server/client operating systems. Should be set to the operating system for the computer defined in the **Name** field.
Username	The username that HP Deployment Management uses to log onto the server/client in order to transfer files or execute commands. This will usually be "HP PPM" if that user account has been generated on this computer.
Password	The password for the given username. The password is hidden, and can be changed by clicking the button to the right of the field.
NT Domain	The domain name to use if this is a Window Server.
Base Path	The path for applications on this computer. In many instances, this is the home directory of the defined username. When HP Deployment Management transfers a file or executes a command on this server, it logs in and changes directories to this base path before proceeding. Note that the directory separators should utilize forward slashes ('/'), even for Windows systems.

Field Name	Description
Connection Protocol	The protocol to be used for server or client connections. SSH is a secure connections protocol, whereas Telnet is not. To use SSH as your connection protocol, you must first set up SSH on the destination machine.
Transfer Protocol	The protocol to be used for moving files between various clients and servers. SCP is a secure transfer protocol, whereas FTP is not. In order to use SCP as your transfer protocol, you must first set up SCP on the destination machine.
Enable Client	Make the client information available to the system. Select the checkbox to make the client information available to the system.

5. In the Environment window, in the **Database** section, enter the database server type in the **Server Type** field.

   - If **Oracle Server** is selected, complete the fields as specified in the following table:

Field Name	Description
Host Name	The DNS name or IP Address of the system running the Oracle® database instance.
Connect String	The Oracle SQL*NET connection string used to connect to this database from a remote system. This is usually the same as the Oracle SID.
User Name	The username for the schema in the database that will be used by HP Deployment Management to make remote database connections.
Password	The password for the given username. The password is hidden, and can be changed by clicking the button to the right of the field.
Oracle SID	The SID of the database. This is used in conjunction with the Port Number for Environment Comparison reporting.
Port Number	The port number of the TNS Listener database. Used for Environment comparison reporting.
DB Link	Reference to a remote Oracle database, if applicable.
Version	The Oracle version number for this database. Used for reporting purposes.
Enable Database	Make the database information available to the system. Select the checkbox to make the database information available to the system.

- If **SQL Server** is selected, complete the fields as specified in the following table:

Field Name	Description
Server Name	The DNS name or IP Address of the system running the SQL database.
DB Name	The DB name used by HP Deployment Management when connecting to the SQL database.
User Login	The login ID used by HP Deployment Management when connecting to the SQL database.
Password	The password for the given login ID.
Port Number	The port number of the database. Used for Environment comparison reporting.
Version	The SQL version number for this database. Used for reporting purposes.
Enable Database	Make the database information available to the system. Select the checkbox to make the database information available to the system.

6. Click **OK.**

Using Application Codes Environments

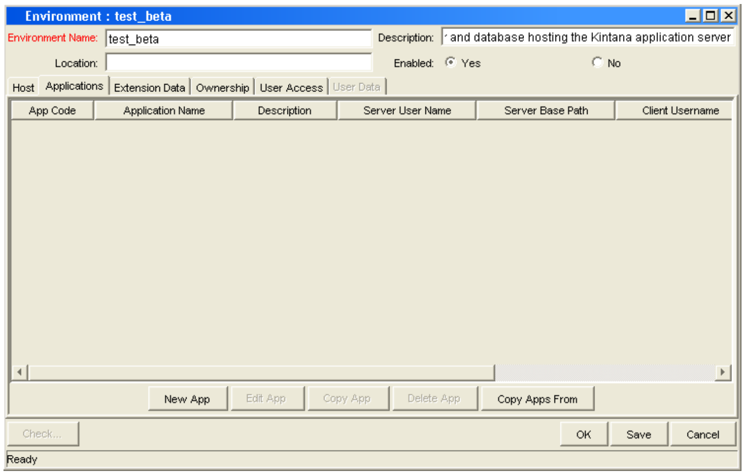
Complex environments are often segmented into subsections called environment applications. The environment information consists of the default set of attributes for an environment. It is rare, however, that an actual environment could be described simply by this set of defaults. For example, files belonging to different applications may reside at different paths and may be owned by different users. SQL scripts may need to be run against a different schema than the one defined on the **Host** tab.

When adding a line to a package, there is the option of choosing the application code to specify the application that the migration object belongs to. When that object is subsequently migrated, the application-specific environment items are referenced in place of the default environment items. As a general rule, any application specific environment item that has no value is substituted by the corresponding environment value.

Environment User Data fields are inherited by each application code and are displayed on the application code's **User Data** tab. **Application Code User Data** fields behave like other application code fields (such as host name and base path), in that blank field values indicate that the application code has the same value as its parent environment. Therefore, required **Environment User Data** fields are not also required at the application code level.

Each environment can contain its own set of applications. The **Applications** tab in the Environment window is shown in *Figure 7-5*.

Figure 7-5. Applications tab



Environment Name: test_beta

Description: ' and database hosting the Kintana application server

Location:

Enabled: ● Yes ○ No

App Code	Application Name	Description	Server User Name	Server Base Path	Client Username

The **Applications** tab consists of the various fields and buttons shown in the *Figure 7-5*. Application fields do not always have to be populated. Click **New App** to open the New Application Code window, where a new application code can be defined.

To define the application codes:

1. From the Workbench shortcut bar, select **Environments > Environments.**

   The Environments Workbench opens.

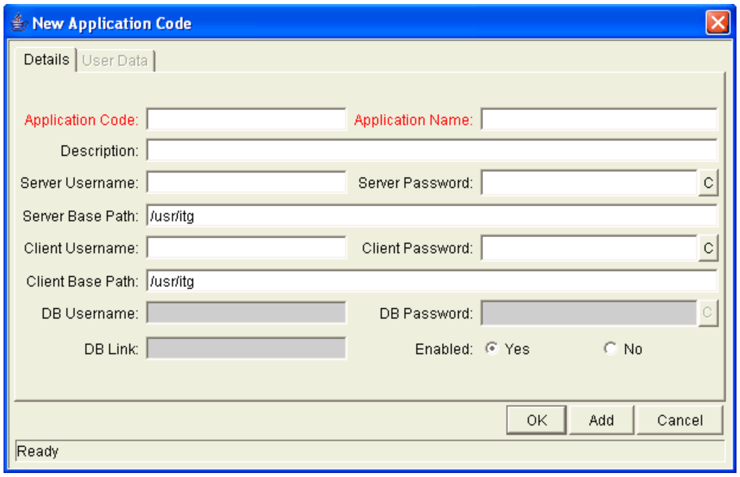
2. Open an environment.

   The Environment window opens.

3. Click the **Applications** tab.

4. Click **New App.**

   The New Application Code window opens.



Server Base Path: /usr/itg

Client Base Path: /usr/itg

Enabled: ● Yes ○ No

5. Complete the fields in the New Application Code window as specified in the following table:

Field Name	Description
App Code	Short name abbreviation for the application.
Application Name	Long name for the application.
Description	A description of the application.
Server User Name	User name that HP Deployment Management should log on as when transferring files or running commands on the environment server for this application, if it is different from the default server username.
Server Password	Password for logging on to the server, if different from the default server password. This field is encrypted.
Server Base Path	Base path for the application on the server machine.
Client User Name	User name that HP Deployment Management should log in as when transferring files or running commands on the environment client for this application, if different from the default client name.
Client Password	Password for logging on to the client, if different from the default client password. This field is encrypted.
Client Base Path	Base path for the application on the client machine.
DB Username	Database username for the application. It is used when running database level commands (such as SQL scripts) for this application.

Field Name	Description
DB Password	Database password for the application. It is used when running database level commands (such as SQL scripts) for this application. This field is encrypted.
DB Link	Name of the database link for this application, if different from the default environment database link.
Enabled	Indicates whether this application environment is enabled.

6. Click **OK.**

7. From the **Applications** tab, click **OK.**

Copying Application Codes from Other Environments

When generating a new environment, all or some of the applications attached to an existing environment can be copied to the new environment to speed up the setup process.

To copy the application codes:

1. From the Workbench shortcut bar, select **Environments > Environments.**

   The Environments Workbench opens.

2. Open an environment.

   The Environment window opens.

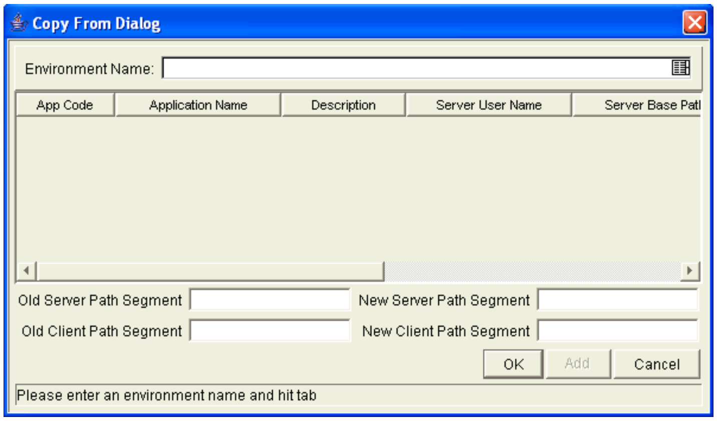
3. Click the **Applications** tab.

4. Click **New App.**

   The New Application Code window opens.

5. Click **Copy Apps From.**

   The Copy From Dialog window opens.



Please enter an environment name and hit tab

6. In the **Environment Name** field, select the environment from which to copy the applications.

   The application code names are listed.

7. To change the base path segment of all the application codes selected, enter the old server and client base path segment and the new server and client base path segment in their respective fields.

8. Select all the applications to copy.

9. Click **Add.**

   These fields have, by default, the server or client base path of the environment from which the applications are being copied. For example, suppose that two applications have been selected. The server base paths for these two applications are `/u2/apps/isi` and `/u2/apps/demo_107`. To change u2 to u3, enter u2 in Old Server Base Path and u3 in New Server Base Path before you copy these applications. Every occurrence of the old server and client base path segment is changed to the new base path segment in all the selected applications. The changes will be reflected in the applications in the environment into which they were copied.

10. After copying the application codes, any necessary modifications such as adding additional applications, deleting applications, or editing any of the applications can be made.

11. Click **OK.**

Setting Ownership and Participants for Environments

Ownership groups are defined by adding security groups to the **Ownership** tab. If no ownership groups are associated with the entity, the entity is considered global and any user who can edit environments can edit, copy or delete the environment. For more information about access grants, see the *Security Model Guide and Reference*.

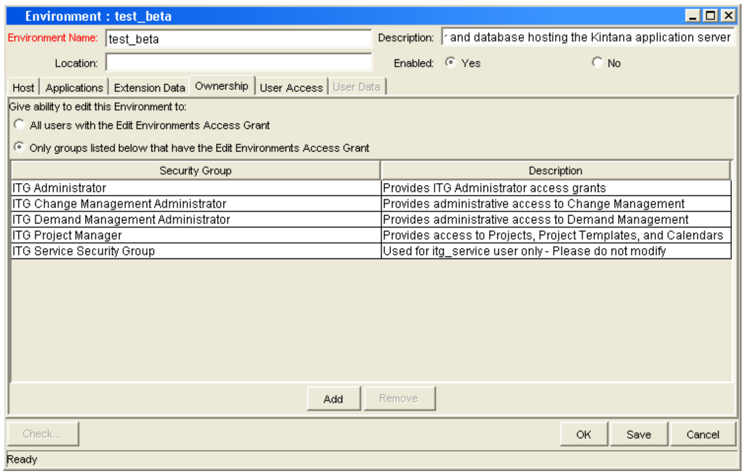
If a security group is disabled or loses its ability to edit environments, that group can no longer edit the entity.

Adding Ownerships to Environments

To add an ownership:

1. From the Workbench shortcut bar, select **Environments > Environments.**
2. Open an environment.

   The Environment window opens.
3. Click the **Ownership** tab.



Environment : test_beta

Environment Name: test_beta

Description: ' and database hosting the Kintana application server

Location:

Enabled: ● Yes ○ No

Give ability to edit this Environment to:

○ All users with the Edit Environments Access Grant

● Only groups listed below that have the Edit Environments Access Grant

Security Group	Description
ITG Administrator	Provides ITG Administrator access grants
ITG Change Management Administrator	Provides administrative access to Change Management
ITG Demand Management Administrator	Provides administrative access to Demand Management
ITG Project Manager	Provides access to Projects, Project Templates, and Calendars
ITG Service Security Group	Used for itg_service user only - Please do not modify

4. Select the ownership option.

   - **All users with the Edit Environment Access Grant**
   - **Only groups listed below that have the Edit Environments Access Grant**

   For **Only groups listed below that have the Edit Environments Access Grant:**

   a. Click **Add.**

      The Add Security Groups window opens.

   b. Select the security groups and click **OK.**

   The security groups are added to the **Ownership** tab.

5. From the **Ownership** tab, click **OK.**

Deleting Ownerships from Environments

To delete an ownership:

1. From the Workbench shortcut bar, select **Environments > Environments.**
2. Open an environment.

   The Environments window opens.

3. Click the **Ownership** tab.
4. Select an ownership.
5. Click **Remove.**
6. Click **OK.**

Adding Participants to Environments

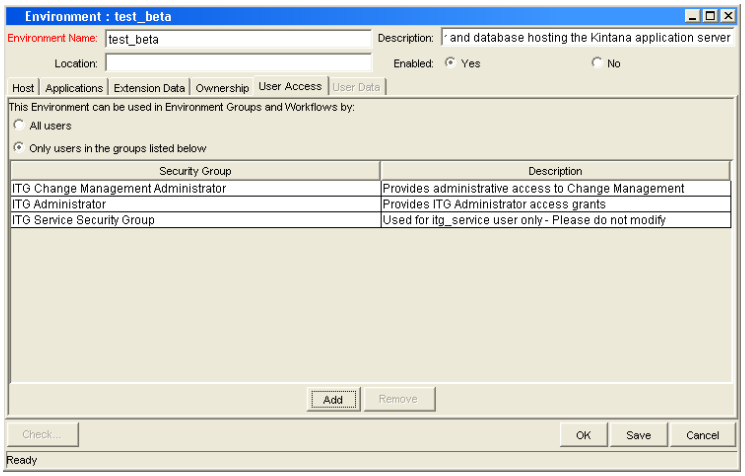
It is possible to control which users can access an environment for use in environment groups and workflows.

To add participants to the environment:

1. From the Workbench shortcut bar, select **Environments > Environments.**
2. Open an environment.

   The Environment window opens.

3. Click the **User Access** tab.



Environment Name: test_beta

Description: ' and database hosting the Kintana application server

Location:

Enabled: ● Yes ○ No

This Environment can be used in Environment Groups and Workflows by:

○ All users

● Only users in the groups listed below

Security Group	Description
ITG Change Management Administrator	Provides administrative access to Change Management
ITG Administrator	Provides ITG Administrator access grants
ITG Service Security Group	Used for itg_service user only - Please do not modify

Ready

4. Select one of the following options:

   - **All Users**
   - **Only Users in the groups listed below**

   If you selected **Only Users in the groups listed below:**

   a. In the **User Access** tab, click **Add.**

      The Add Security Group window opens.

   b. Select the security groups and click **OK.**

      The security groups are added to the **User Access** tab.

5. From the **User Access** tab, click **OK.**

Deleting Participants from Environments

To delete participants from the environment:

1. From the Workbench shortcut bar, select **Environments > Environments.**

   The Environments Workbench opens.

2. Open an environment.

   The Environment window opens.

3. Click the **User Access** tab.

4. Select a participant to delete.

5. Click **Remove.**

6. Click **OK.**

Environment Maintenance and Utilities

This section provides information on validating and maintaining the environment definitions in HP Deployment Management.

Testing Environment Setups

To check the validity of the Environment:

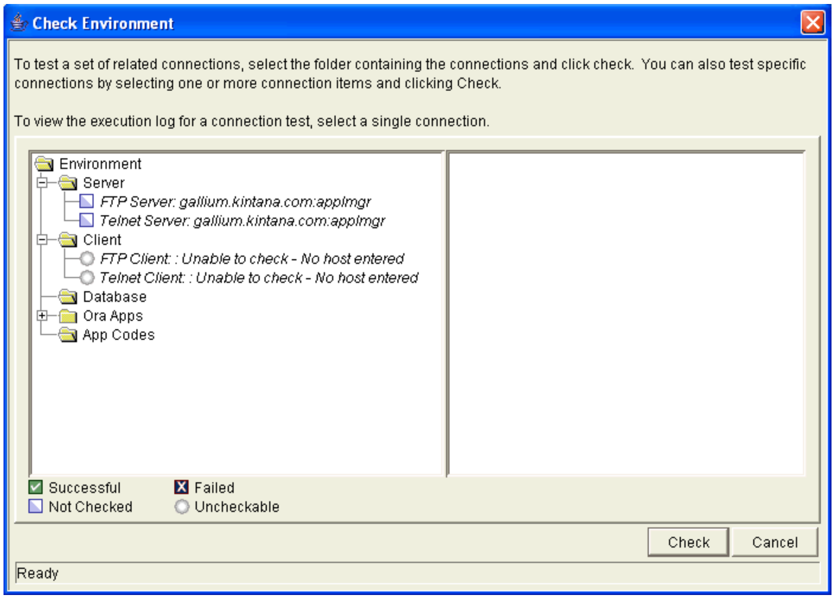
1. From the Workbench shortcut bar, select **Environments > Environments.**

2. Open an environment.

   The Environment window opens.

3. Click **Check.**

   The Check Environment window opens.



Check Environment

To test a set of related connections, select the folder containing the connections and click check. You can also test specific connections by selecting one or more connection items and clicking Check.

To view the execution log for a connection test, select a single connection.

Environment
Server
FTP Server: gallium.kintana.com:applmgr
Telnet Server: gallium.kintana.com:applmgr
Client
FTP Client: : Unable to check - No host entered
Telnet Client: : Unable to check - No host entered
Database
Ora Apps
App Codes
☑ Successful
☒ Failed
□ Not Checked
○ Uncheckable
Check
Cancel
Ready

4. Select the environment connections to check.

   Select a folder, such as Server, to check all connections defined for that category. Specific connections can also be tested by selecting the individual checkboxes by the connection item.

5. Click **Check.**

   The system verifies the environment definition. Results from the environment check commands are displayed in the Log File section of the Check Environment window. Use log file output to troubleshoot any connection problems identified during the environment check.

   Environment definition testing includes actions performed during regular code migration, such as opening a Telnet session to the server, opening an FTP session to the server, and connecting to the database. While the environment checker cannot guarantee that all migrations will be successful, it can help catch some of the most common setup problems.

   While the check process can take a significant amount of time, it is recommended that any new environment is checked once all the data for it is entered. Additionally, it is good practice to periodically check all environments to catch any obvious problems, such as changed passwords or disabled accounts.

6. In the Check Environment window, click **Cancel.**

7. In the Environment window, click **Cancel.**

Mass Updates of Base Paths

It is possible to update server and client base path segments in the environment and all of its applications at the same time. This functionality is useful to relocate a particular environment and all of its applications onto a new disk or partition.

To perform a mass update of the base paths:

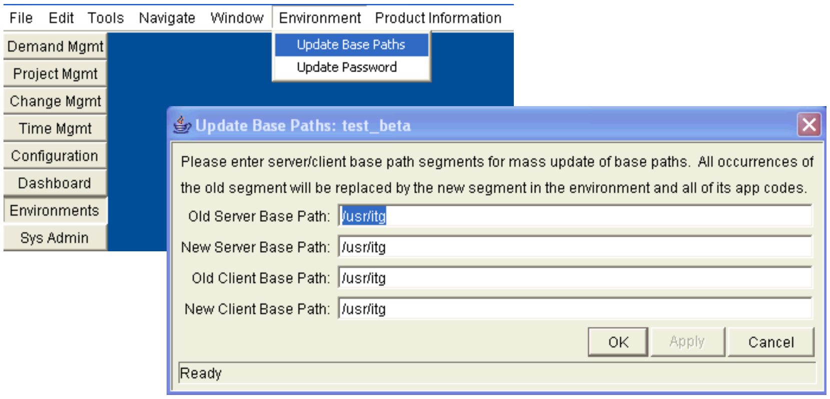
1. From the Workbench shortcut bar, select **Environments > Environments.**

2. Open an environment.

   The Environment window opens.

3. From the menu, select **Environment > Update Base Paths.**

   The Update Base Path window opens.



Update Base Paths: test_beta

Please enter server/client base path segments for mass update of base paths. All occurrences of the old segment will be replaced by the new segment in the environment and all of its app codes.

Old Server Base Path: /usr/itg

New Server Base Path: /usr/itg

Old Client Base Path: /usr/itg

New Client Base Path: /usr/itg

Ready

4. Enter the old server/client base path segment and the new server/client base path segment in the fields provided.

   The default value in the old server/client base path field is the environment's current server/client base path segment.

5. Click **Apply** or **OK.**

   Every occurrence of the old server/client base path segment will be replaced by the new server/client base path segment in the Environment and all of its applications.

   For example, suppose that two applications have been selected. The server base paths for these two applications are `/u2/apps/isi` and `/u2/apps/demo_107`. To change u2 to u3, type u2 in Old Server Base Path and u3 in New Server Base Path before you copy these applications. Every occurrence of the old server/client base path segment is then changed to the new base path segment in all selected applications. The changes are reflected in the applications in the Environment into which you copy them.

   The Environment window opens.

6. Click **OK.**

Environment Password Management Utility

A single user can have access to multiple password protected environments. It is often convenient to use a single username and password for all of the environments that a single user encounters. If the user decided to change their password or if that user's job functions were transferred to another user, it would take hours to update the environment passwords in each environment window.

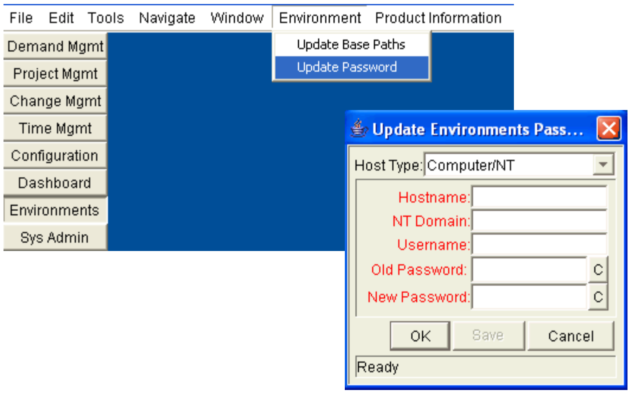
The Environment Password Management Utility enables a user to update their password in all of the environments located on a single host, simultaneously. This utility only updates passwords with matching parameters, such as Hostname, Username, Old Password, and Connect String. These updates can be made using the Environments interface.

To change the environment password for a user:

1. From the Workbench shortcut bar, select **Environments > Environments.**
2. Open an environment.

   The Environment window opens.
3. From the menu, select **Environments > Update Password.**

   The Update Environments Password window opens.


Host Type: Computer/NT

Ready

4. In the **Host Type** field, select the host type.

   The required fields change to match requirements of the selected host type.

5. Complete the fields in the Update Environments Password window as specified in the following table:

Field Name	Description
Host Name	The DNS name or IP Address of the system running the Oracle database instance.
NT Domain	The domain name to use if this is a Window Server.
Username	The username for the schema in the database that will be used by HP Deployment Management to make remote database connections.
Old Password	The old password for the given username. The password is hidden, and can be changed by clicking the button to the right of the field.
New Password	The new password for the given username. The password is hidden, and can be changed by clicking the button to the right of the field.

6. Click **OK.**

   The Environment window opens.

7. Click **OK.**

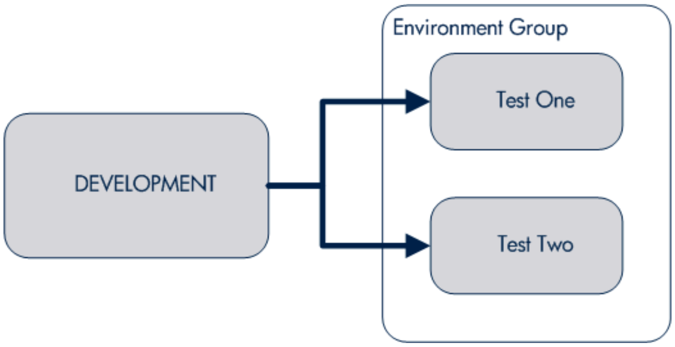
Overview of Environment Groups

Environment groups define a set of environments which can be referenced as the source or destination for object migrations. Environment groups are defined and edited using the Environment Group Workbench.

Use environment groups where you want to execute a workflow step on multiple environments. For example, it may be necessary to migrate an object to multiple testing environments for different targeted tests. These multiple environments can be referenced together as one environment group (*Figure 7-6*).

➤ The ksc_connect command works with environment groups in the “At least one” mode. When the ksc_connect command attempts to connect to the first server in the environment group, and in case it fails to connect, then the ksc_connect command tries to connect to the second server in the environment group, and then the third, and so on. When the ksc_connect command does succeed in connecting to a server, the remaining commands in the object type are run before the command again tries to connect to the next server in the environment group. If the ksc_connect command fails to connect to all servers in the environment group, the command exits with a failure.

Figure 7-6. Environment group



Overview of Configuring Environment Groups

Environment groups are configured in the Environment Group window. The following lists the main areas found in the Environment Group window:

- **General information.** General information includes basic information concerning the environment group, such as the environment group name and description. For more information, see *Opening the Environment Group Workbench* on page 218.
- **Environments.** Use the **Environments** tab to add existing HP Deployment Management environments to an environment group.
- **Host.** The **Host** tab lists basic information about the client, server, and database for the associated environments. For more information, see *Creating Environment Groups* on page 220.
- **Applications Codes.** The **Applications Codes** tab lists basic information about the application codes linked to the associated environments.
- **Serial Execution Order.** The **Serial Execution Order** tab displays the order in which environments are acted on. For more information, see *Setting the Order of Executions* on page 221.

- **Ownership.** Configure who can edit the environment group. For more information, see *Setting Ownership and Participants for Environment Groups* on page 222.
- **User Access.** Configures participants of the environment group. Participants can then be given specific access rights to the environment group. For more information, see *Adding Participants to Environment Groups* on page 223.

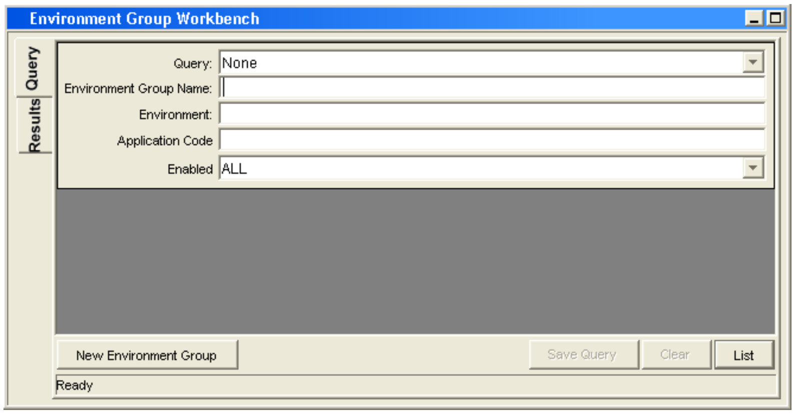
Opening the Environment Group Workbench

To open the Environment Group Workbench:

1. Log on to PPM Center.
2. From the menu bar, select **Administration > Open Workbench.**

   The PPM Workbench opens.
3. From the shortcut bar, select **Environments > Environment Groups.**

   The Environment Group Workbench opens.



Query: None
Environment Group Name:
Environment:
Application Code
Enabled ALL

Ready

Configuring General Information for Environment Groups

To configure the general information of an environment group:

1. From the Workbench shortcut bar, select **Environments > Environment Groups.**

   The Environment Group Workbench opens.

2. Open an environment group.

   The Environment Group window opens.

3. Enter the information specified in the following table:

Field Name	Description
Environment Group Name	The name of the environment group.
Enabled	Makes the environment available to the system. Select **Yes** to make the environment available to the system.
Description	A brief description of how the environment group is being used.
Execution Order	Indicates whether the environments associated with an environment group are executed in parallel order (all at once) or in a specific serial order. Enables the **Serial Execution Order** tab when **Serial** is selected.
Source Environment (No App Code)	Indicates the default environment selected when the environment group is used as the source environment in an execution workflow step with no application code specified.

4. To save the changes to the environment, do one of the following:

   Click **OK** to save the changes and close the Environment Group window. Click **Save** to save the changes and leave the Environment Group window open. Click **Cancel** to lose the changes and close the Environment Group window.

Creating Environment Groups

To define a new environment group:

1. From the Workbench shortcut bar, select **Environments > Environment Groups.**

   The Environment Group Workbench opens.

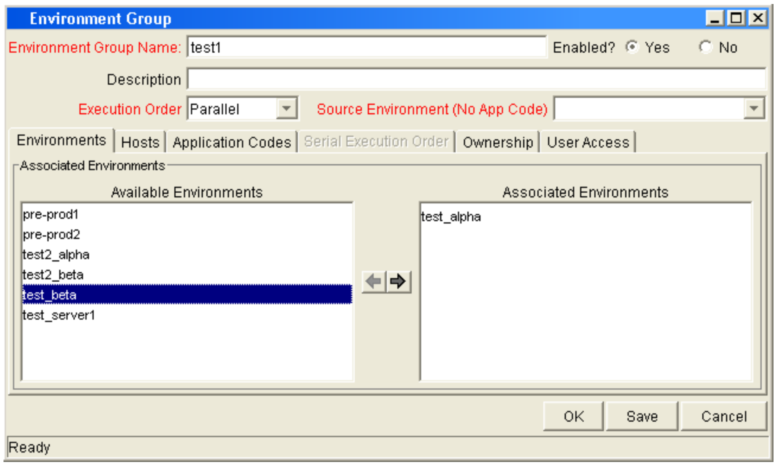
2. Open an environment group.

   The Environment Group window opens to the **Environments** tab.

3. In the **Available Environments** field, select an environment name.

4. Click the right pointer.

   The selected environment is moved to the **Associated Environments** field.



Environment Group Name: test1 Enabled? ● Yes ○ No

Description

Execution Order: Parallel Source Environment (No App Code)

Environments | Hosts | Application Codes | Serial Execution Order | Ownership | User Access

Associated Environments

Available Environments: pre-prod1, pre-prod2, test2_alpha, test2_beta, test_beta, test_server1

Associated Environments: test_alpha

5. In the **Environments** tab, click **OK.**

   The changes to the environment group are saved.

Setting the Order of Executions

The environments are executed in sequential order until all executions have completed. Each environment execution waits for the previous environment execution to complete (success or fail) before beginning.

To set the environment execution order:

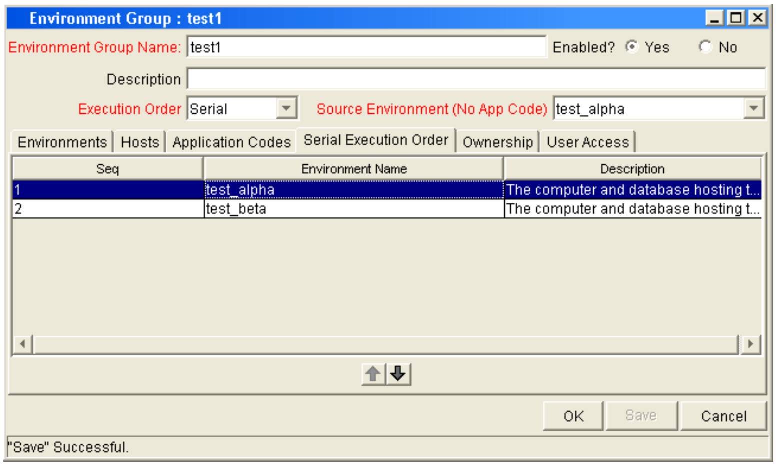
1. From the Workbench shortcut bar, select **Environments > Environment Groups.**

   The Environment Group Workbench opens.

2. Open an environment group.

   The Environment Group window opens.

3. Click the **Serial Application Order** tab.



Environment Group Name: test1 Enabled? ● Yes ○ No

Description

Execution Order Serial Source Environment (No App Code) test_alpha

Seq	Environment Name	Description
1	test_alpha	The computer and database hosting t...
2	test_beta	The computer and database hosting t...

"Save" Successful.

4. Select a row to move.
5. To move the selected environment to a new sequence position, use the arrow pointers under the tab.
6. Click **OK.**

Setting Ownership and Participants for Environment Groups

Ownership groups are defined by adding security groups to the **Ownership** tab. If no ownership groups are associated with an environment group, that environment group is treated as global, and any user who can edit environments for the entity can edit, copy or delete the entity. If a security group is disabled or loses the ability to edit environments, its members can no longer edit the environment group. For more information on access grants, see the *Security Model Guide and Reference*.

Adding Ownerships to Environment Groups

Different groups of users can have exclusive control over the environment groups used by their group. These groups are referred to as ownership groups. Members of the ownership group are the only users who can edit, delete or copy the environment group. Each environment group can be assigned multiple ownership groups.

Ownership groups are defined by adding security groups to the **Ownership** tab.

To set the ownership for an environment group:

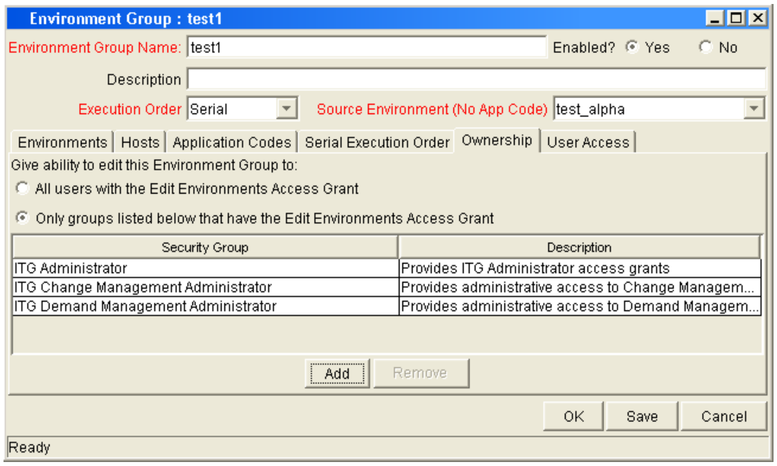
1. From the Workbench shortcut bar, select **Environments > Environment Groups.**

   The Environment Group Workbench opens.

2. Open an environment group.

   The Environment Group window opens.

3. Click the **Ownership** tab.



Environment Group : test1

Environment Group Name: test1

Enabled? ● Yes ○ No

Description

Execution Order: Serial

Source Environment (No App Code): test_alpha

Give ability to edit this Environment Group to:

○ All users with the Edit Environments Access Grant

● Only groups listed below that have the Edit Environments Access Grant

Security Group	Description
ITG Administrator	Provides ITG Administrator access grants
ITG Change Management Administrator	Provides administrative access to Change Managem...
ITG Demand Management Administrator	Provides administrative access to Demand Managem...

Ready

4. Select the ownership option.

   - **All user with the Edit Environments Access Grant**
   - **Only groups listed below that have the Edit Environments Access Grant**

   If **Only groups listed below that have the Edit Environments Access Grant** is selected:

   a. Click **Add.**

      The Add Security Groups window opens.

   b. Select the security groups and click **OK.**

   The security groups are added to the **Ownership** tab.

5. Click **OK.**

Deleting Ownerships from Environment Groups

To delete an ownership:

1. From the Workbench shortcut bar, select **Environments > Environment Groups.**

   The Environment Group Workbench opens.

2. Open an environment group.

   The Environment Group window opens.

3. Click the **Ownership** tab.

4. Select the ownership to remove.

5. Click **Remove.**

6. Click **OK.**

Adding Participants to Environment Groups

To add participants to the environment:

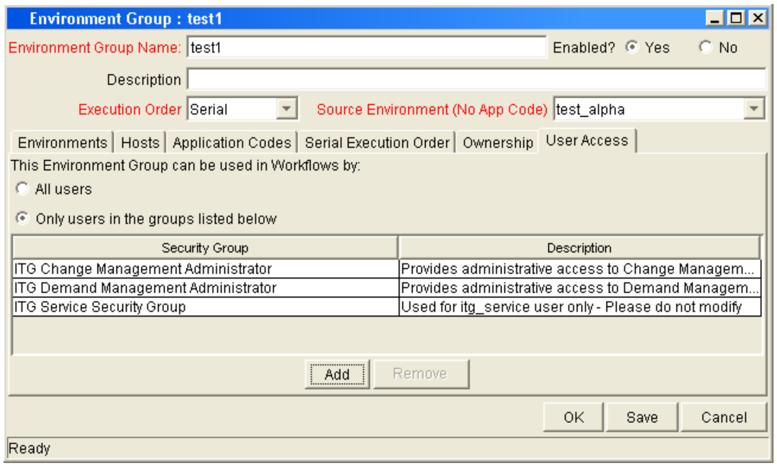
1. From the Workbench shortcut bar, select **Environments > Environment Groups.**

   The Environment Group Workbench opens.

2. Open an environment group.

   The Environment Group window opens.

3. Click the **User Access** tab.



Environment Group : test1

Environment Group Name: test1 Enabled? ● Yes ○ No

Description

Execution Order Serial Source Environment (No App Code) test_alpha

This Environment Group can be used in Workflows by:

○ All users

● Only users in the groups listed below

Security Group	Description
ITG Change Management Administrator	Provides administrative access to Change Managem...
ITG Demand Management Administrator	Provides administrative access to Demand Managem...
ITG Service Security Group	Used for itg_service user only - Please do not modify

Ready

4. Under **This Environment Group can be used in Workflows by** field, select one of the following.

   - To allow all users to edit this environment, select **All Users.**
   - To allow only members of a listed security group to edit the environment, select **Only Users in the groups listed below.**

   If you select **Only Users in the groups listed below:**

   a. Click **Add.**

      The Add Security Group window opens.

   b. Select a security group and click **OK.**

      The **User Access** tab displays the security group name.

5. Click **OK.**

Deleting Participants from Environment Groups

To delete participants from an environment group:

1. From the Workbench shortcut bar, select **Environments > Environment Groups.**

   The Environment Group Workbench opens.

2. Open an environment group.

   The Environment Group window opens.

3. Click the **User Access** tab.

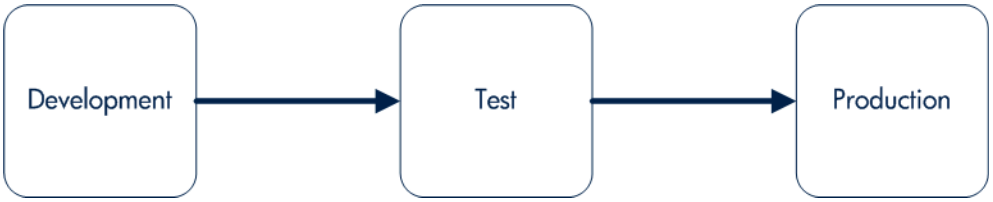
4. Select a participant to delete, and then click **Remove.**

5. Click **OK.**

Overview of Environment Refresh

Deployment Management enables controlled migration of software and application changes to various instances. Deployment Management maintains an audit trail of all this activity, providing visibility into the changes for each environment. There are, however, situations when you need to refresh an environment, essentially replacing that environment with a physical copy of another environment.

Figure 7-7. Example software migration



If a standard software migration is from a Development environment to a Test environment (see *Figure 7-7*) to a Production environment, you may want to periodically copy the Production environment into both the Development and Test environments. This activity would involve copying all objects at the file system and database level from Production into the Development and Test areas, producing mirror images of the Production environment. This refresh function synchronizes the environments, and moves additional production data into development and testing environments.

After using the refresh function, Deployment Management has the ability to update its audit history and its in-process Packages. This update is performed using Deployment Management's Environment Refresh Workbench functionality.

The environment refresh function in Deployment Management does not perform the actual physical refresh of one environment with another. It updates the Deployment Management data to be consistent with the refreshed physical occurrence.

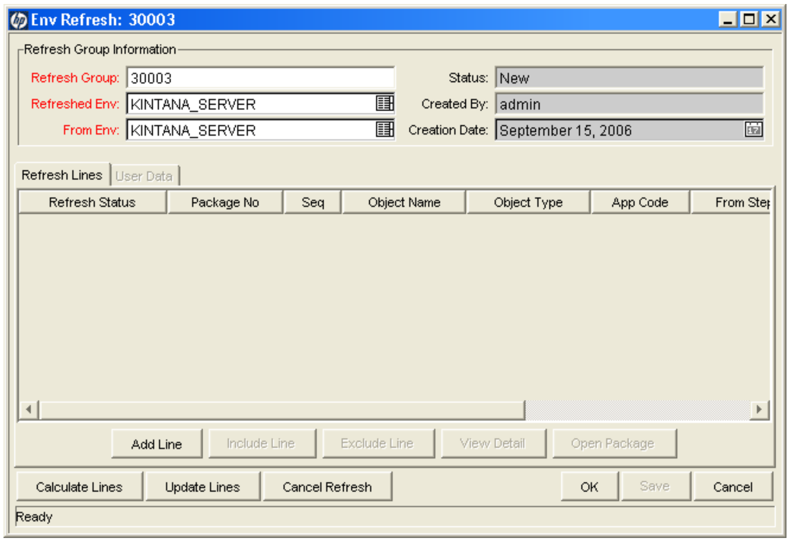
Deployment Management’s Environment Refresh performs the following functions:

- Identifies open package lines that have migrated through the environment and are now inaccurate due to the software changes. In the example described in *Figure 7-7*, the refreshed environments would be Development and Test, while the source environment would be Production. After the physical refresh, these package lines will be inaccurate due to the software changes no longer being present in the refreshed environment.
- Updates the list of affected package lines as necessary, including adding or deleting lines from the list.
- Updates the Deployment Management internal object inventory tables to reflect the physical refresh. This copies the audit history from the source environment to the audit history of the refreshed environment. Since they are now physical matches of each other, their audit history should be the same.
- Updates the open package lines in the refresh list and resets them so they are eligible for migration into the refreshed environment again.

Overview of Configuring Environment Refresh

Environment refresh consists of an Environment Refresh window. Typically, environment refreshes are performed as part of maintaining environments between test systems.

Figure 7-8. Environment refresh window



Refresh Group:	30003	Status:	New
Refreshed Env:	KINTANA_SERVER	Created By:	admin
From Env:	KINTANA_SERVER	Creation Date:	September 15, 2006

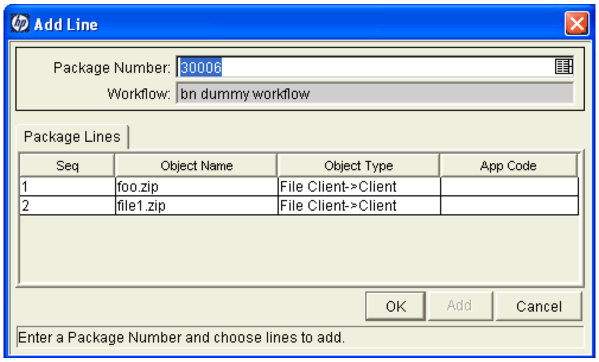
The following lists the major sections found in the Environment Refresh window:

- **General Information.** General information stores the basic information for the refresh group, such as the names of the source and refreshed environment.
- **Refresh Lines tab.** The list of refresh lines includes basic information such as the package number and the line sequence as well as the refresh status of the individual line. In addition to viewing refresh line information, several actions can be performed from the **Refresh Lines** tab:
  - **Add Line.** Manually adds a new refresh line to the list of refresh lines for this refresh group. This command generates the list of lines to be updated (such as a replacement of the calculate lines action), and adds additional lines to the list that might not be picked up by the calculate lines logic.

    When this action is selected, a prompt to add the refresh line information through the Add Line window appears (see *Figure 7-9*). In this window, enter a package number and select one or more package lines.

    This action can only be performed before the refresh group has been processed.

    Figure 7-9. Add Line window

    Add Line

    Package Number: 30006

    Workflow: bn dummy workflow

    Package Lines

Seq	Object Name	Object Type	App Code
1	foo.zip	File Client->Client	
2	file1.zip	File Client->Client	

    Enter a Package Number and choose lines to add.

  - **Include Line and Exclude Line.** Use **Include Line** and **Exclude Line** to change the status of each individual refresh line on the list before the refresh group has been updated. All refresh lines are originally marked with a refresh status of **Pending.** When a refresh group is actually processed, using **Update Lines**, all lines in the list with a status of **Pending** are updated. Any refresh lines with a status of **Exclude** are not processed and are only used for reporting purposes.

    This action can only be performed before the refresh group has been processed.

    - **View Detail.** If a single refresh line is selected in the list, certain details regarding the refresh line can be viewed. Use this action to open up a read-only Line Detail window to view the information.

    - **Open Package.** For the complete details on the refresh line and the parent package, select a single line and click **Open Package** to navigate to the Package window and automatically open up the appropriate package. Then view the complete package as well as perform any eligible package activities.

- **User Data tab.** Each Deployment Management implementation can add custom fields that are common to all refresh groups. These new user data fields essentially become additional refresh group header data. These fields are normally determined during initial system configuration. As part of the configuration, the layout of the fields is determined. This configuration includes:

    - The look and feel of the field (whether the field is a drop down menu or auto-complete or free form text)

    - The logic behind the field (how is the information in the field validated)

    - Graphic characteristics (such as field height and width and placement in the tab)

Opening the Environment Refresh Workbench

To open the Environment Refresh Workbench:

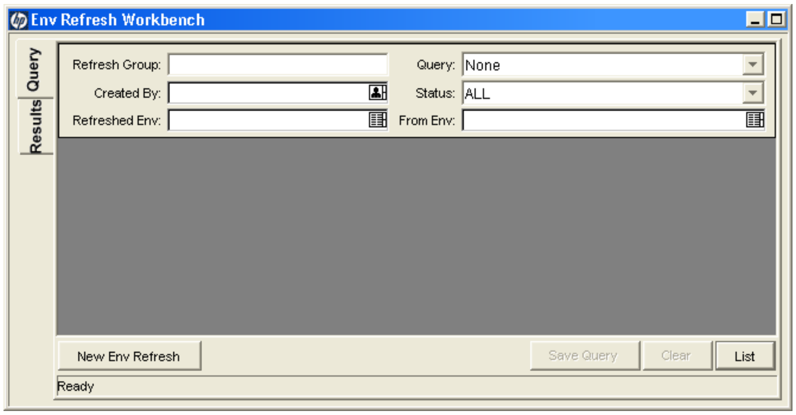
1. Log on to PPM Center.

2. From the menu bar, select **Administration > Open Workbench.**

   The PPM Workbench opens.

3. From the shortcut bar, select **Deployment Mgmt > Environment Refresh.**

   The Environment Refresh Workbench opens.



Refresh Group:

Created By:

Refreshed Env:

Query: None

Status: ALL

From Env:

Configuring General Information for Environment Refreshes

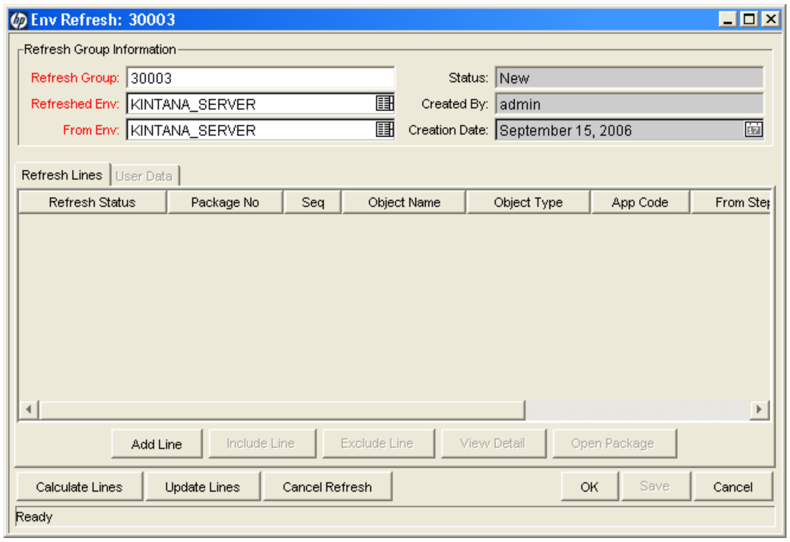
To configure the general information of an environment refresh:

1. From the Workbench shortcut bar, select **Deployment Mgmt > Environment Refresh.**

   The Environment Refresh Workbench opens.

2. Click **New Env Refresh** to open the Environment Refresh window.

   The Environment Refresh window opens.



Refresh Group: 30003

Refreshed Env: KINTANA_SERVER

From Env: KINTANA_SERVER

Status: New

Created By: admin

Creation Date: September 15, 2006

Refresh Status	Package No	Seq	Object Name	Object Type	App Code	From Ste

3. Fill in the following information:

   - **Refresh Group.** Automatically filled-in.
   - **Refreshed Env.** Select the destination environment to be refreshed.

- **From Env.** Select the source environment.

4. Click **Calculate Lines.**

   This action is non-reversible. Once performed, the status of the refresh group is changed to Completed.

   This action queries all open package lines and generates a list of lines that have been migrated to the environment being refreshed but not to the source environment. (After the physical refresh, these package lines will be inaccurate because the software changes are no longer present in the refreshed environment). This action only looks for packages using workflows containing both the source and refreshed environment.

   All package lines matching this criteria will be added to the package lines list for the refresh group.

   If the refresh group already contains a list of package lines, this list will be deleted before the new calculation of eligible lines is performed.

   Clicking **Calculate Lines** can only be performed if the status of the refresh group is New or Open (which means that no updates have been performed). If the original status of the group was New, the status is updated to Open once the package lines are added.

5. Click **Update Lines.**

   This action is non-reversible. Once performed, the status of the refresh group is changed to Completed.

   Once all the appropriate package lines have been added to the lines list and reviewed, this action is used to perform the actual update of the Deployment Management data. This action performs two activities:

   - The Deployment Management internal object inventory tables are updated to reflect the physical refresh. This copies the audit history from the source environment to the audit history of the refreshed environment. Since they are now physical matches of each other, their audit history should be the same.

   - All package lines in the refresh lines list that have a refresh status of Include are reset to be eligible for migration into the refreshed environment again. This is done using special workflow transitions from the currently eligible step to the migration step for the refreshed environment. This allows all the changes that were erased during the physical refresh to be easily migrated back into the environment. After each refresh line is updated, its refresh status is updated to Completed.

6. The refresh status is now completed.

➤ For refresh groups with a status of New and Open, it is possible to cancel the environment refresh. To cancel the environment refresh, click **Cancel Refresh.** This sets the status of the refresh group to Cancel, preventing any further action on the refresh group but allowing the details of the specific refresh group to be queried.

Configuring a Workflow with an Environment Refresh

For this example Development and Test are two IT Governance servers.

To create a workflow environment refresh:

1. From the Workbench shortcut bar, select **Configuration > Workflows.**

   The Workflows Workbench opens.

2. Click **New Workflow** to open a new workflow.

   A Workflow window opens.

3. In the Workflow window, fill in the required information.

4. In the Workflow window, select the **Layout** tab.

5. From the Workflow Step Sources window, drag a Deployment Management execution step onto the **Layout** tab.

   The Workflow Step window opens.

6. In the Workflow Step window, complete the required fields.

   - Type `Development` in the **Source Environment** field.
   - Type `Test` in the **Dest Environment** field.

7. From the Workflow Step Sources window, drag another Deployment Management execution step onto the **Layout** tab.

   The Workflow Step window opens.

8. In the Workflow Step window, complete the required fields.

   - Type `Test` in the **Source Environment** field.
   - Type `Development` in the **Dest Environment** field.

9. Add a transition from the first Deployment Management execution step to the second Deployment Management execution step.

   The Step Transitions window opens.

10. Complete the information in the Step Transitions window.
11. From the Workflow Step Sources window, drag a Deployment Management Close step onto the **Layout** tab.

    The Workflow Step window opens.
12. In the Workflow Step window, complete the required fields.
13. Add a transition from the second Deployment Management execution step to the Deployment Management Close step.

    The Step Transitions window opens.
14. Complete the information in the Step Transitions window.
15. Save and close the workflow.

Configuring a Package with an Environment Refresh

To create a package environment refresh:

1. From the Workbench shortcut bar, select **Deployment Mgmt > Packages.**

   The Package Workbench opens.

2. Click **New Package** to open a new package.

   The Package window opens.

3. Fill in the required Package information.

   - Select the Workflow configured in *Configuring a Workflow with an Environment Refresh*.

4. In the Package window, click **New Line** to add a line to the package.

   The Add Line window opens.

5. In the **Object Type** field, select **File Migration.**

6. In the Add Line window, complete the remaining fields.

7. Click **Submit** and run the package through to completion.

   It might be necessary to manually set the override status to Succeed in the **Status** tab.

8 Configuring Notification Templates

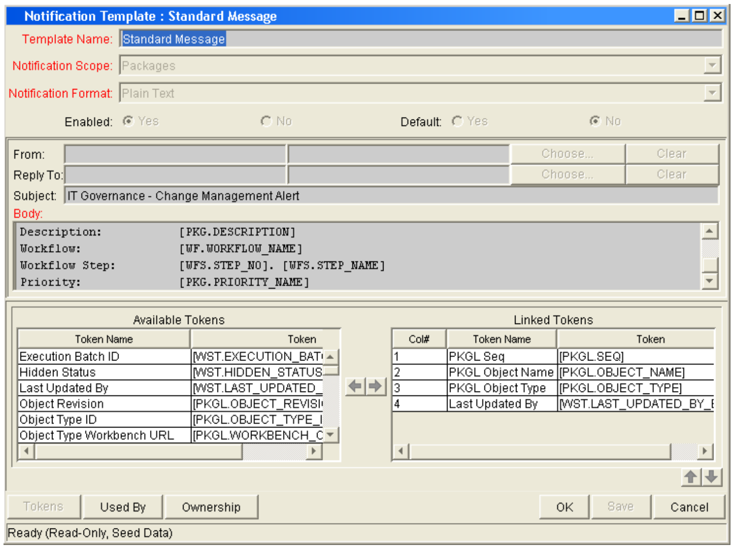
In This Chapter:

Overview of Notification Templates

Notification templates are preconfigured notifications that can be used to quickly construct the body of your message (see *Figure 8-1*). Notification templates are used with the following PPM Center entities:

- Tasks
- Projects
- Requests
- Packages
- Releases
- Workflows
- Reports

Figure 8-1. Notifications Template window



Notification Template : Standard Message

Template Name: Standard Message

Notification Scope: Packages

Notification Format: Plain Text

Enabled: ● Yes ○ No   Default: ○ Yes ● No

From:

Reply To:

Subject: IT Governance - Change Management Alert

Body:

```
Description:            [PKG.DESCRIPTION]
Workflow:               [WF.WORKFLOW_NAME]
Workflow Step:          [WFS.STEP_NO]. [WFS.STEP_NAME]
Priority:               [PKG.PRIORITY_NAME]
```

Available Tokens

Token Name	Token
Execution Batch ID	[WST.EXECUTION_BAT
Hidden Status	[WST.HIDDEN_STATUS
Last Updated By	[WST.LAST_UPDATED_
Object Revision	[PKGL.OBJECT_REVISI
Object Type ID	[PKGL.OBJECT_TYPE_I
Object Type Workbench URL	[PKGL.WORKBENCH_C

Linked Tokens

Col#	Token Name	Token
1	PKGL Seq	[PKGL.SEQ]
2	PKGL Object Name	[PKGL.OBJECT_NAME]
3	PKGL Object Type	[PKGL.OBJECT_TYPE]
4	Last Updated By	[WST.LAST_UPDATED_BY_I

Ready (Read-Only, Seed Data)

Opening the Notification Template Workbench

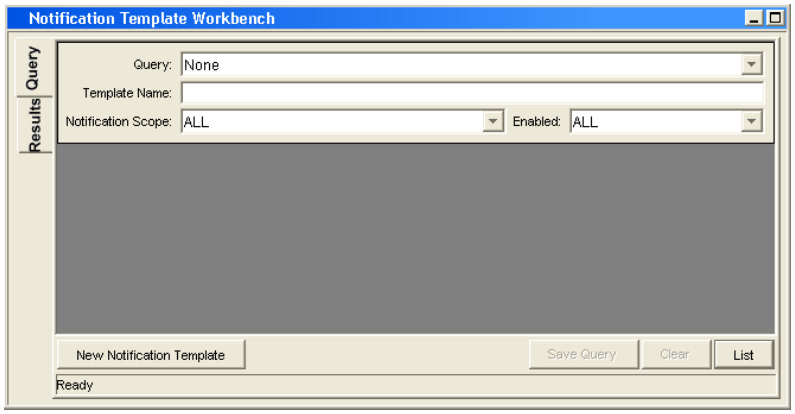
To open the Notification Template Workbench:

7. Log on to PPM Center.
8. From the menu bar, select **Administration > Open Workbench.**

   The PPM Workbench opens.

9. From the shortcut bar, select **Configuration > Notification Templates.**

   The Notification Template Workbench opens.



Query: None

Template Name:

Notification Scope: ALL

Enabled: ALL

Ready

Deleting Notification Templates

You can not delete notification templates that are referenced from an existing notification. To delete a notification template you must first remove these references. Referenced notification templates can be disabled. To see if a notification template is references, see *Checking the Usage of Notification Templates* on page 245.

Creating Notification Templates

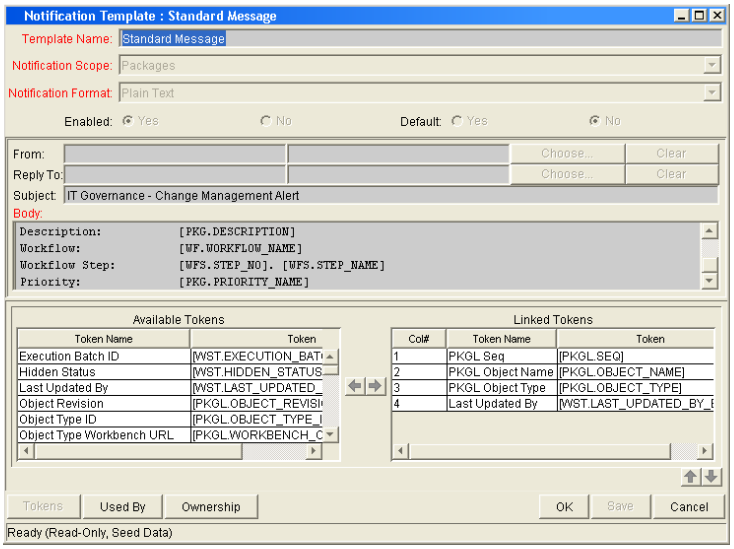
To create a new notification template:

1. From the Workbench shortcut bar, select **Configuration > Notification Templates.**

   The Notification Template Workbench opens.

2. Click **New Notification Template.**

   The Notification Template window opens.



Notification Template : Standard Message

Template Name: Standard Message

Notification Scope: Packages

Notification Format: Plain Text

Enabled: ● Yes ○ No Default: ○ Yes ● No

From:

Reply To:

Subject: IT Governance - Change Management Alert

Body:

```
Description:            [PKG.DESCRIPTION]
Workflow:               [WF.WORKFLOW_NAME]
Workflow Step:          [WFS.STEP_NO]. [WFS.STEP_NAME]
Priority:               [PKG.PRIORITY_NAME]
```

Available Tokens

Token Name	Token
Execution Batch ID	[WST.EXECUTION_BAT
Hidden Status	[WST.HIDDEN_STATUS
Last Updated By	[WST.LAST_UPDATED_
Object Revision	[PKGL.OBJECT_REVISI
Object Type ID	[PKGL.OBJECT_TYPE_I
Object Type Workbench URL	[PKGL.WORKBENCH_C

Linked Tokens

Col#	Token Name	Token
1	PKGL Seq	[PKGL.SEQ]
2	PKGL Object Name	[PKGL.OBJECT_NAME]
3	PKGL Object Type	[PKGL.OBJECT_TYPE]
4	Last Updated By	[WST.LAST_UPDATED_BY_

Ready (Read-Only, Seed Data)

3. Enter the information specified in the following table.

Field Name	Description
Template Name	Enter the name of the new notification template.
Notification Scope	Include the product section where this notification template is to be used. In the list, select one of the following: ■ **Packages** ■ **Projects** ■ **Release Distribution** ■ **Reports** ■ **Request Field Changes** ■ **Requests** ■ **Task Dates** ■ **Task Exceptions** The default notification scope is **Packages.** Selecting another notification scope changes the format of the notification template.
Notification Format	Include the format of the body of the notification. In the list, select one of the following: ● **Plain Text** ● **HTML**
Enabled	Make the notification template available to the system. To make the notification available to the system, select **Yes.**
Default	Make the notification template the default notification template for the system. To make the notification template the default notification template, select **Yes.**

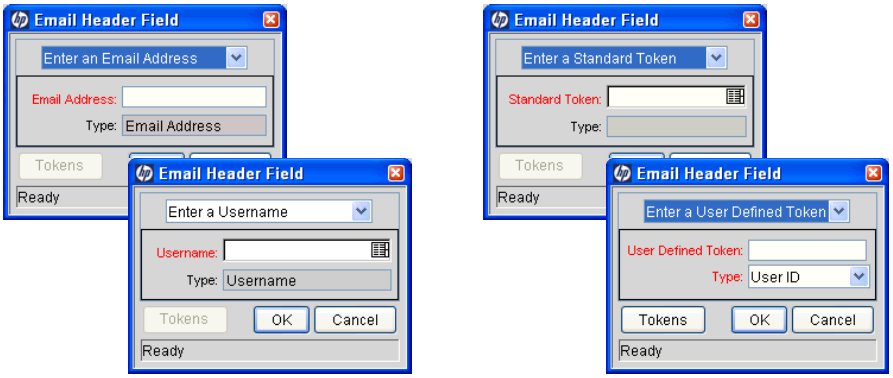
4. Enter a **From** address.

   a. In the Notification Template window, in **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 from the list, then it is necessary to enter an Email Address. If you select **User Defined Token,** click **Tokens** to bring up a complete list of available tokens or type in a specific token.



Enter an Email Address
Email Address:
Type: Email Address

Enter a Username
Username:
Type: Username

Enter a Standard Token
Standard Token:
Type:

Enter a User Defined Token
User Defined Token:
Type: User ID

c. Enter the appropriate information in the required field.

d. If a user defined token has been entered, select the token type that corresponds to the evaluated token value.

e. In the Email Header Field window, click **OK.**

5. In the Notification Template window, enter a **Reply** 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 enter 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.

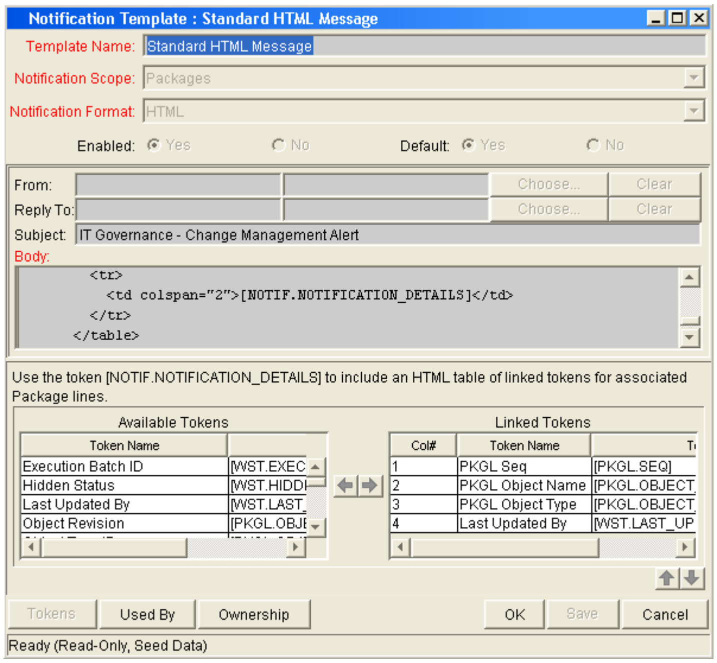
c. Enter the information in the required field.

d. If **User Defined Token** is entered, select the token type that corresponds with the evaluated token value.

e. In the Email Header Field window, click **OK.**

6. In the **Body** field, enter the body of the notification text.

Make sure the format of the body of the notification is the same as specified in **Notification Format.** HTML notifications for HP Deployment Management must include the token '[NOTIF.NOTIFICATION_ DETAILS]' within the <*body*> tags to incorporate linked tokens.



Notification Template : Standard HTML Message

Template Name: Standard HTML Message

Notification Scope: Packages

Notification Format: HTML

Enabled: ● Yes ○ No Default: ● Yes ○ No

From:

Reply To:

Subject: IT Governance - Change Management Alert

Body:

```
        <tr>
          <td colspan="2">[NOTIF.NOTIFICATION_DETAILS]</td>
        </tr>
      </table>
```

Use the token [NOTIF.NOTIFICATION_DETAILS] to include an HTML table of linked tokens for associated Package lines.

Available Tokens

Token Name	
Execution Batch ID	[WST.EXEC
Hidden Status	[WST.HIDD
Last Updated By	[WST.LAST_
Object Revision	[PKGL.OBJE

Linked Tokens

Col#	Token Name	T
1	PKGL Seq	[PKGL.SEQ]
2	PKGL Object Name	[PKGL.OBJECT
3	PKGL Object Type	[PKGL.OBJECT
4	Last Updated By	[WST.LAST_UP

Ready (Read-Only, Seed Data)

7. In the **Body** field, add tokens to the body of the text.

   To add tokens to the body of the notification template:

   a. Click **Tokens.**

      The Token Builder window opens.

   b. Select a token.

   c. In the **Token** field, copy the name of the token and paste the name in the **Body** field.

   d. Click **Close.**

8. 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* on page 242.

9. 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 the entity, the entity is considered global and any user with the edit access grant for the entity can edit, copy or delete it. For more 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.

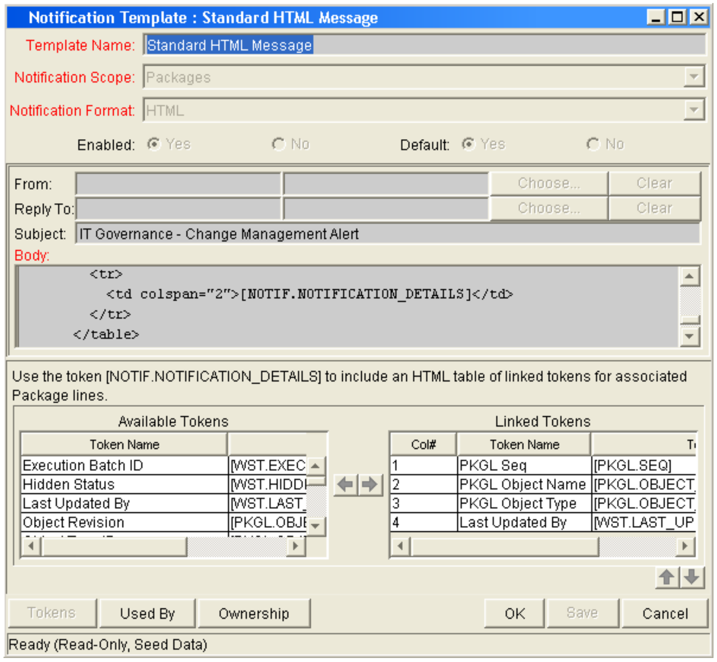
To configure the ownership of a notification template:

1. From the Workbench shortcut bar, select **Configuration > Notification Templates.**

   The Notification Template Workbench opens.

2. Open a notification template.

   The Notification Template window opens.



Notification Template : Standard HTML Message

Template Name: Standard HTML Message

Notification Scope: Packages

Notification Format: HTML

Enabled: ● Yes ○ No Default: ● Yes ○ No

From:

Reply To:

Subject: IT Governance - Change Management Alert

Body:

```
      <tr>
        <td colspan="2">[NOTIF.NOTIFICATION_DETAILS]</td>
      </tr>
    </table>
```

Use the token [NOTIF.NOTIFICATION_DETAILS] to include an HTML table of linked tokens for associated Package lines.

Available Tokens

Token Name	
Execution Batch ID	[WST.EXEC
Hidden Status	[WST.HIDDI
Last Updated By	[WST.LAST_
Object Revision	[PKGL.OBJI

Linked Tokens

Col#	Token Name	T
1	PKGL Seq	[PKGL.SEQ]
2	PKGL Object Name	[PKGL.OBJECT_
3	PKGL Object Type	[PKGL.OBJECT_
4	Last Updated By	[WST.LAST_UP

Ready (Read-Only, Seed Data)

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**

   If **Only groups listed below that have the Edit Notification Template access grant** is selected:

   a. Click **Add.**

      The Add Security Groups window opens.

   b. In the **Security Groups** field, select the security groups.

   c. Click **OK.**

      The **Ownership** tab lists the selected security groups.

5. Click **OK.**

   The changes to the notification template are saved.

Deleting Ownerships from Notification Templates

To delete an ownership:

1. From the Workbench shortcut bar, select **Configuration > Notification Templates.**

   The Notification Template Workbench opens.

2. Open a notification template.

   The Notification Template window opens.

3. Click **Ownership.**

   The Ownership window opens.

4. Select an ownership to remove.

5. Click **Remove.**

6. Click **OK.**

Configuring Notification Intervals

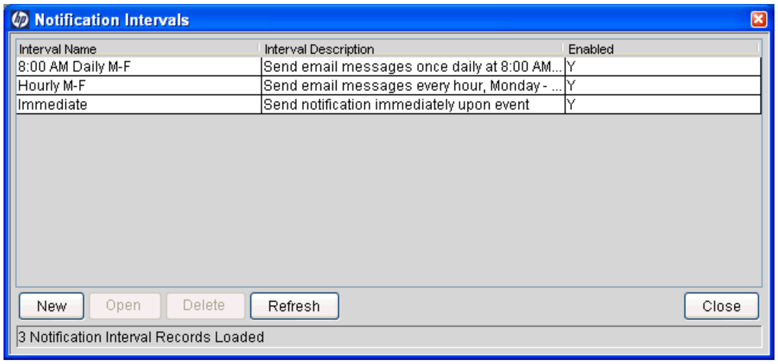
To create a new notification template:

1. From the Workbench shortcut bar, select **Configuration > Notification Templates.**

   The Notification Template Workbench opens.

2. From the menu, select **Notification Templates > Intervals.**

   The Notification Intervals window opens.

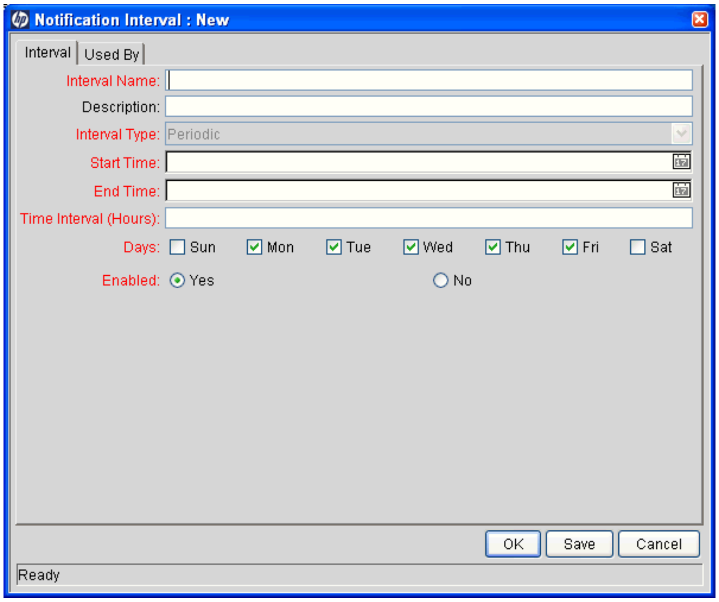


Interval Name	Interval Description	Enabled
8:00 AM Daily M-F	Send email messages once daily at 8:00 AM...	Y
Hourly M-F	Send email messages every hour, Monday - ...	Y
Immediate	Send notification immediately upon event	Y

3 Notification Interval Records Loaded

3. Click **New.**

   The Notification Interval: New window opens to the **Interval** tab.



Interval Name:

Description:

Interval Type: Periodic

Start Time:

End Time:

Time Interval (Hours):

Days: □ Sun ☑ Mon ☑ Tue ☑ Wed ☑ Thu ☑ Fri □ Sat

Enabled: ● Yes ○ No

Ready

4. Enter the information specified in the following table:

Field Name	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** is set, 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 email message.

Checking the Usage of Notification Templates

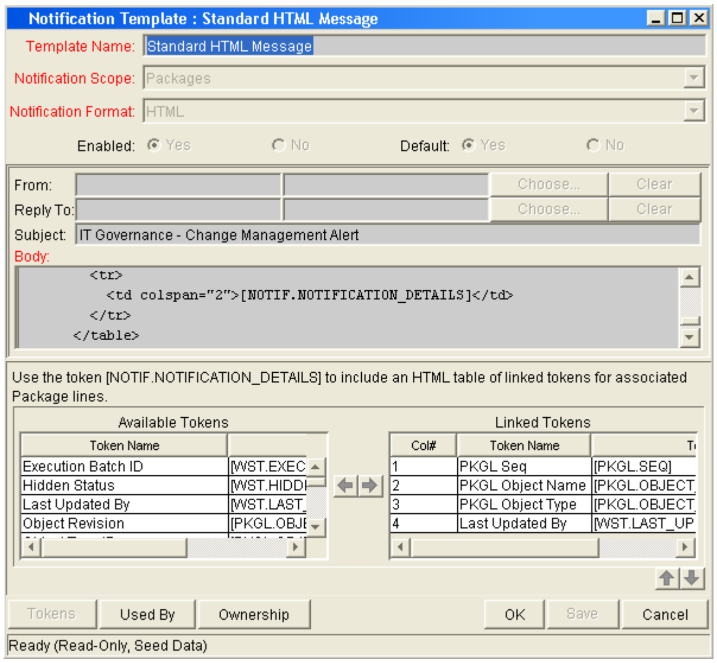
To check the usage of a notification template:

1. From the Workbench shortcut bar, select **Configuration > Notification Templates.**

   The Notification Template Workbench opens.

2. Open notification template.

   The Notification Template window opens.



Notification Template : Standard HTML Message

Template Name: Standard HTML Message

Notification Scope: Packages

Notification Format: HTML

Enabled: ● Yes ○ No Default: ● Yes ○ No

From:

Reply To:

Subject: IT Governance - Change Management Alert

Body:

```
      <tr>
        <td colspan="2">[NOTIF.NOTIFICATION_DETAILS]</td>
      </tr>
    </table>
```

Use the token [NOTIF.NOTIFICATION_DETAILS] to include an HTML table of linked tokens for associated Package lines.

Available Tokens

Token Name	
Execution Batch ID	[WST.EXEC
Hidden Status	[WST.HIDDI
Last Updated By	[WST.LAST_
Object Revision	[PKGL.OBJE

Linked Tokens

Col#	Token Name	T
1	PKGL Seq	[PKGL.SEQ]
2	PKGL Object Name	[PKGL.OBJECT_
3	PKGL Object Type	[PKGL.OBJECT_
4	Last Updated By	[WST.LAST_UP

Ready (Read-Only, Seed Data)

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.**

9 Configuring User Data

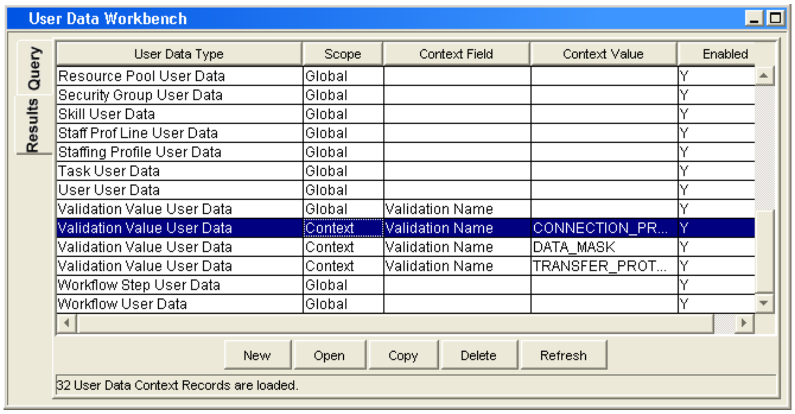
In This Chapter:

Overview of 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, you can use “user data fields” to capture additional information specific to organizations. For example, if you want to include an additional field on every package, you can open **Validation Value User Data** and define the extra field. The field would then be displayed on the **User Data** tab for a validation.

You configure user data types from the User Data Workbench in the User Data Context window. In *Figure 9-1* the **Results** tab in User Data Workbench shows a partial list of the available user data types.

Figure 9-1. User data types



User Data Type	Scope	Context Field	Context Value	Enabled
Resource Pool User Data	Global			Y
Security Group User Data	Global			Y
Skill User Data	Global			Y
Staff Prof Line User Data	Global			Y
Staffing Profile User Data	Global			Y
Task 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	TRANSFER_PROT...	Y
Workflow Step User Data	Global			Y
Workflow User Data	Global			Y

32 User Data Context Records are loaded.

Each user data type consists of the following four components, all of which are required to fully define a user data type. The following lists these components:

- **User Data Type.** This column lists the user data type. PPM Center creates all of these. You cannot create new user data types. However, you can define fields for a user data type.

- **Scope.** This column lists the scope of the user data type field. The two possible values are:

  - **Global.** If the scope is global, the **User Data** tab for every designated entity contains the defined field.

  - **Context.** This is a context-sensitive user data type field. If the scope is context, the defined user data field is added only to the **User Data** tab of entities with specific **Context Field** and **Context Value** definitions.

- **Context Field.** This column lists the name of the context-sensitive field. It applies only to user data type fields with context scope. Because only one Context Field value is available for each user data type, 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 the defined entity. You can configure the major attributes of each field, including its graphical presentation, the validation method, and whether it is required.

Referring to User Data

Once a user data field is created, you can refer to it from other parts of the product by its token name, proceeded by the entity abbreviation and the user data (UD) qualifier. For example, Validation Value User Data might have a field name of Class Name, a token value of CLASS_NAME, and a user data qualifier of USER_DATA1.

Migrating User Data

For any configuration entity with user data type fields, the data in the user data type fields is migrated along 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. Check to make sure 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.

Overview of Configuring User Data

This section provides information about the main components of the User Data Context window.

- **General information.** This section displays basic information about the user data, including as the user data type and the user data context value. For more information, see *Configuring General Information for User Data Types* on page 251.
- **Fields.** Used to create additional fields for a user data type. See *Creating User Data Fields* on page 253.
- **Layout.** Once all of the fields are created for a user data type, the layout of those fields can be configured on the **Layout** tab. See *Configuring User Data Layouts* on page 262.

Opening the User Data Workbench

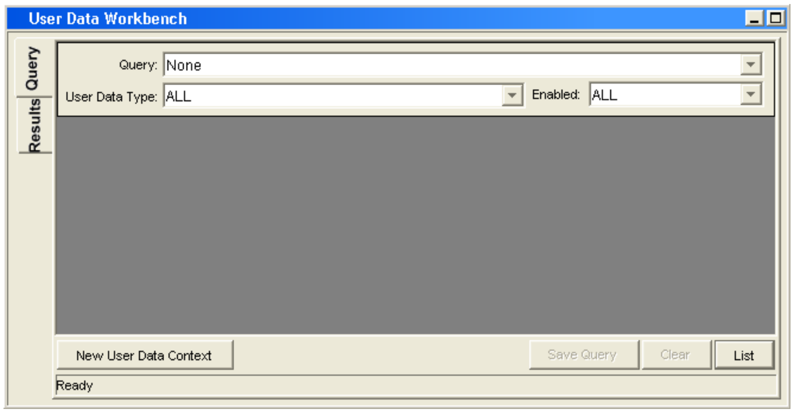
To open the User Data Workbench:

1. Log on to PPM Center.
2. From the menu bar, select **Administration > Open Workbench.**

   The PPM Workbench opens.

3. From the shortcut bar, select **Configuration > User Data.**

   The User Data Workbench opens.



Query: None

User Data Type: ALL

Enabled: ALL

Ready

Configuring General Information for User Data Types

To configure the general information for a user data type:

1. From the Workbench shortcut bar, select **Configuration > User Data.**

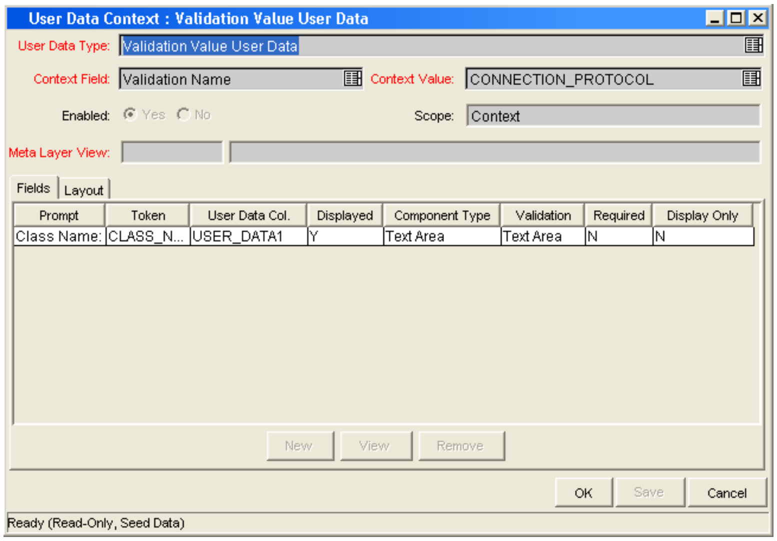
   The User Data Workbench opens.

2. Select an existing user data type or create a new user data type.

   To configure a global user data type, you must open an existing user data type such as Skill User Data, which already exists PPM Center.

   When configuring a context-sensitive user data type, you can select an existing context-sensitive user data type or click **New** to create a context-sensitive user data type.

   The User Data Context window opens.



User Data Context : Validation Value User Data

User Data Type: Validation Value User Data

Context Field: Validation Name

Context Value: CONNECTION_PROTOCOL

Enabled: ● Yes ○ No

Scope: Context

Meta Layer View:

Prompt	Token	User Data Col.	Displayed	Component Type	Validation	Required	Display Only
Class Name:	CLASS_N...	USER_DATA1	Y	Text Area	Text Area	N	N

Ready (Read-Only, Seed Data)

3. Enter the information specified in the following table:

Field Name	Description
User Data Type	Selects the name of the user data type. For global user data types, this field is automatically populated. For context-sensitive user data types, select the context-sensitive user data type from the list. You can choose one of the following context-sensitive user data types: ■ **Package User Data** ■ **Validation Value User Data**
Context Field	The name of the context-sensitive field. This field is disabled for user data types where Scope = Global. This field is automatically filled in for context-sensitive user data. The following lists the **User Data Types** and the **Context Field:** ■ **Package User Data - Priority** ■ **Validation Value User Data - Validation Name**
Context Value	Selects the value for the **Context Field.** This field is disabled for user data types where Scope = Global. For context-sensitive user data types, select the context value from the list. Only one **Context Value** can be defined at a time. For example, you cannot have two context-sensitive user data types with the same Context Field and Context Value (such as Priority = Critical).
Enable	Indicates whether or not the user data type is available to PPM Center.
Scope	The category of user data type. This field is automatically filled in based on the user data type. The possible scopes for a user data type are: ● **Global.** The standard user data type scope. When Scope = Global, every designated entity has the defined field added to the **User Data** tab. ● **Context.** A context-sensitive user data type. When Scope is defined as Context, only those entities with the correct **Context Field** definition and **Context Value** definition receive the defined user data field.
Meta Layer View	Meta layer views relate information specific 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.

4. Do one of the following:

   - To save the changes and close the User Data Context window, click **OK.**
   - To save the changes and leave the window open, click **Save.**

Creating User Data Fields

➤ Not all user data field types have **Dependency** and **Security** tabs.

To create a new user data field:

1. From the Workbench shortcut bar, select **Configuration > User Data.**

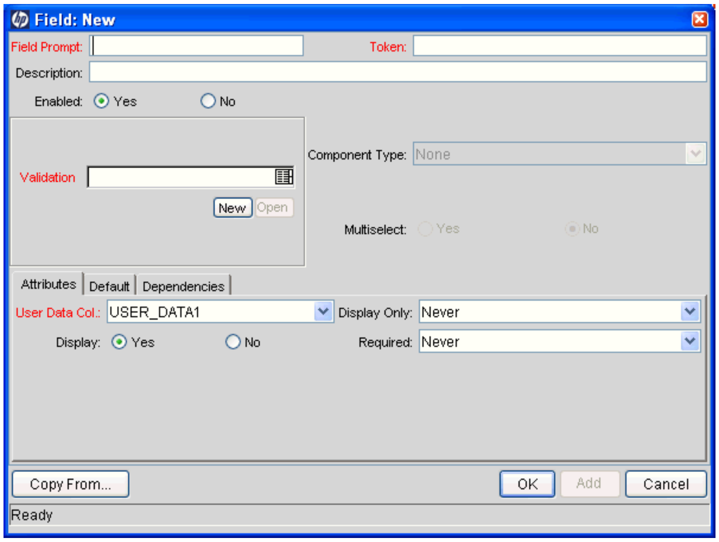
   The User Data Workbench opens.

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 **New.**

   The Field window opens.



Field: New

Enabled: ● Yes ○ No

Component Type: None

Multiselect: ○ Yes ● No

User Data Col.: USER_DATA1

Display Only: Never

Display: ● Yes ○ No

Required: Never

Ready

4. Enter the information specified in the following table:

Field Name	Description
Field Prompt	The prompt visible for the user data field in the request.
Token	An uppercase text string used to identify this field. The token name must be unique for the specific user data. An example of a token name is ASSIGNED_TO_USER_ID.
Description	A description of the user data field.
Enabled	Indicates whether or not the field is turned on for this user data.
Validation	Indicates the validation logic to determine the valid values for this field. This could be a list of user-defined values, a rule that the result has to be a number, and so on.
Component Type	Defines the visual characteristics of the field (list, free form text field, and so on). This is derived from the selected validation. This field is read-only.
Multiselect	Indicates whether or not the field lets users select more than one entry. Only valid for fields with an auto-complete component for the validation.

5. Click the **Attributes** tab.

6. Enter the information specified 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 if the user sees 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.

7. Click the **Defaults** tab.

8. Enter the information specified in the following table:

Field Name	Description
Default Type	Defines if the field will have a default value. Either default the field with a constant value or default it from the value in another user data field.
Visible Value	If a default type of **Constant** is selected, the constant value can be entered here.
Depends On	To default from another field, choose the token name of that field. When using this user data, every time a value is entered or updated in the source field, it will automatically be entered or updated in this destination field.

9. Click the **Dependencies** tab.

10. Enter the information specified in the following table:

Field Name	Description
Clear When ___ Changes	Indicates that the current field should be cleared when the specified field changes.
Display Only When	Indicates that the current field should only be editable when certain logical criteria are satisfied. The field functions with two adjacent fields, a list that contains logical qualifiers, and a text field. To use this functionality, select **Use Dependency Rules** in the **Attributes** tab.
Required When	Indicates that the current field should be required when certain logical criteria are satisfied. The field functions with two adjacent fields, a list that contains logical qualifiers, and a text field. To use this functionality, select **Use Dependency Rules** in the **Attributes** tab.

11. To specify the users who can view and edit this field:

   a. Click the **Security** tab.

   b. Click **Edit.**

      The Edit Field Security window opens.

   c. Enter the information specified in the following table:

Field Name	Description
Visible to all users	Checking this option allows all users to see the field. If this option is not checked, you can set who can see the field. The default is for all users to be able to see a field. If this option is not checked, the Select User/ Security Group that can view this field is enabled. Deselecting the **Visible to all users** or **Editable by all users** checkboxes enables the Select Users/Security Groups that can view this field section of the Edit Field Security window.
Editable by all users	Checking this option allows all users to edit the field. If this option is not checked, you can set who can edit the field. The default is for all user to be able to edit a field. De-selecting the **Visible to all users** or **Editable by all users** checkboxes enables the Select Users/Security Groups that can view this field of the Edit Field Security window.
Enter a Security Group (list)	To select the format for specifying users to grant visibility and edit permission, use the **Enter a Security Group** field. The field displays the formats to choose users. The list dynamically updates the Security Group Validate auto-complete. The choices are: • **Enter a Username.** Select a specific user a to see and/or edit the field. The user must have an email address. • **Enter a Security Group.** Select a specific security group to see and/or edit the field. • **Enter a Standard Token.** Select a standard token to see and/or edit the field. • **Enter a User Defined Token.** Select a user defined token to see and/or edit the field. Selecting the Enter a User Defined Token format enables the **Tokens** button. Selecting an item from the Enter a Security Group list dynamically updates the **Enter a Security Group** field.

Field Name	Description
Security Group	Provides a field for specifying the recipient. If the **Enter a Security Group** field displays: • **Enter a Username,** then the Validate: Username window is returned. • **Enter a Security Group,** then the Validate: Security Group window is returned. • **Enter a Standard Token,** then the Validate: Standard Token window is returned. • **Enter a User Defined Token,** then the Validate: User Defined Token window is returned.

12. Click **OK.**

    The Field window displays the new field.

13. Click **OK.**

Copying a Field Definition

You can streamline the process of adding fields by copying the definitions of existing fields.

To copy a field definition:

1. From the Workbench shortcut bar, select **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 **New.**

   The Field window opens to the **Fields** tab.

4. Click **New.**

   The Field: New window opens.

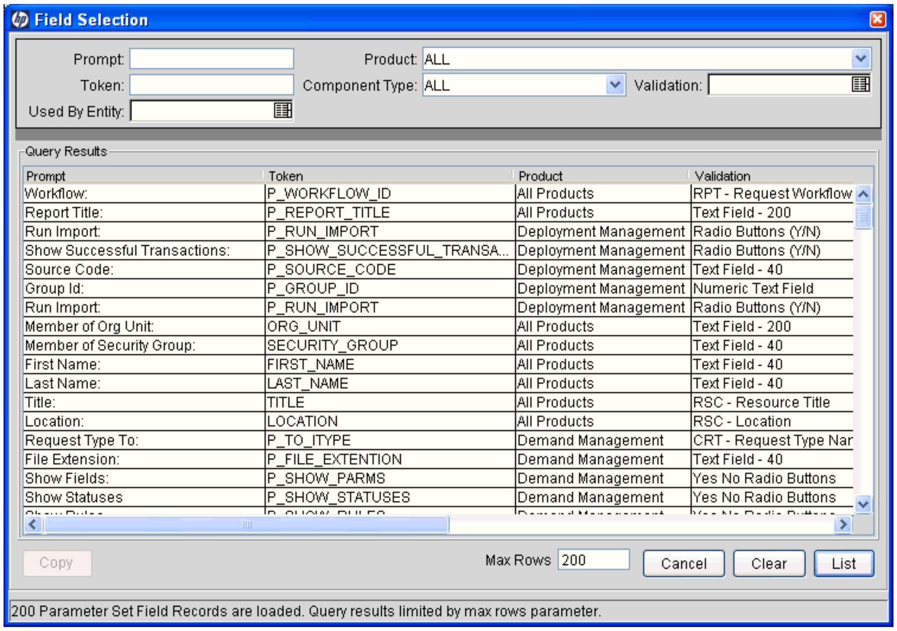
5. Click **Copy From.**

   The Field Selection window opens.

6. Enter the required information.

7. Click **Copy From.**

   The Field Selection window refreshes with fields matching the search criteria.



Field Selection

Prompt:
Token:
Used By Entity:
Product: ALL
Component Type: ALL
Validation:

Query Results

Prompt	Token	Product	Validation
Workflow:	P_WORKFLOW_ID	All Products	RPT - Request Workflow
Report Title:	P_REPORT_TITLE	All Products	Text Field - 200
Run Import:	P_RUN_IMPORT	Deployment Management	Radio Buttons (Y/N)
Show Successful Transactions:	P_SHOW_SUCCESSFUL_TRANSA...	Deployment Management	Radio Buttons (Y/N)
Source Code:	P_SOURCE_CODE	Deployment Management	Text Field - 40
Group Id:	P_GROUP_ID	Deployment Management	Numeric Text Field
Run Import:	P_RUN_IMPORT	Deployment Management	Radio Buttons (Y/N)
Member of Org Unit:	ORG_UNIT	All Products	Text Field - 200
Member of Security Group:	SECURITY_GROUP	All Products	Text Field - 40
First Name:	FIRST_NAME	All Products	Text Field - 40
Last Name:	LAST_NAME	All Products	Text Field - 40
Title:	TITLE	All Products	RSC - Resource Title
Location:	LOCATION	All Products	RSC - Location
Request Type To:	P_TO_ITYPE	Demand Management	CRT - Request Type Nar
File Extension:	P_FILE_EXTENTION	Demand Management	Text Field - 40
Show Fields:	P_SHOW_PARMS	Demand Management	Yes No Radio Buttons
Show Statuses	P_SHOW_STATUSES	Demand Management	Yes No Radio Buttons

Max Rows 200

200 Parameter Set Field Records are loaded. Query results limited by max rows parameter.

8. Select a field to copy.

   You can query fields using several criteria, including the token name or field prompt. You can also perform more complex queries. For example, you could list all fields that reference a specific validation or all fields that a specific entity uses.

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

10. Make any necessary changes, and then click **OK.**

Editing User Data Fields

To edit an existing field:

1. From the Workbench shortcut bar, select **Configuration > User Data.**

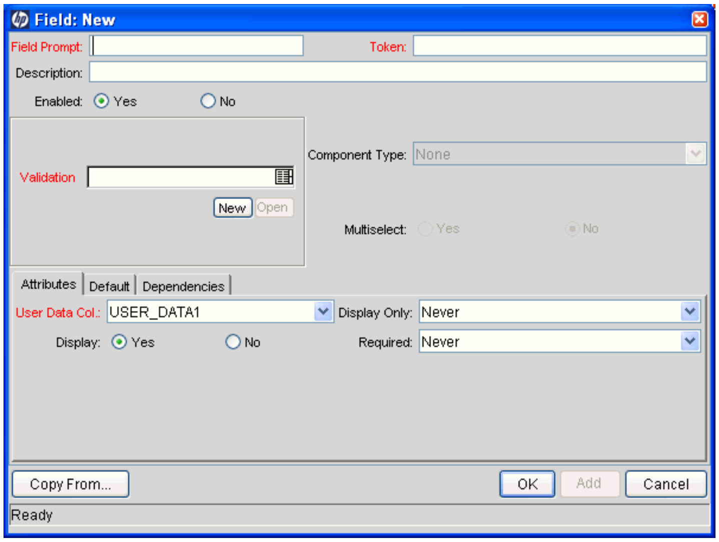
   The User Data Workbench opens.

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 edit, and then click **Edit.**

   The Field window opens.



Enabled: ● Yes ○ No

Component Type: None

Multiselect: ○ Yes ● No

User Data Col.: USER_DATA1

Display Only: Never

Display: ● Yes ○ No

Required: Never

Ready

4. Make the required changes.

   Be sure to include the **Attributes, Default,** and **Dependencies** tabs. For information about these tabs, see *Creating User Data Fields* on page 253.

5. Click **OK.**

6. 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 9-1*.
- Become required after another field meets a logical condition defined in *Table 9-1*.

Table 9-1. Field dependencies

Logical Qualifier	Description
`like`	A `like` condition looks for close matches of the value to the contents of the field chosen.
`not like`	A `not like` condition looks for contents in the selected field that are not close matches to the value field.
`is equal to`	An `is equal to` condition looks for an exact match of the value to the contents of the field chosen.
`is not equal to`	An `is not equal to` condition is true when there are no results exactly matching the value of the field contents.
`is null`	An `is null` condition is true when the field selected is blank.
`is not null`	An `is not null` condition is true when the field selected is not blank.
`is greater than`	An `is greater than` condition looks for a numerical value larger than the value entered in the value field.
`is less than`	An `is less than` condition looks for a numerical value below the value entered in the value field.
`is less than equal to`	An `is less than equal to` condition looks for a numerical value below or the same as the value entered in the value field.
`is greater than equal to`	An `is greater than equal to` condition looks for a numerical value larger than or the same as the value entered in the value field.

To configure a user data field dependency:

1. From the Workbench shortcut bar, select **Configuration > User Data.**

   The User Data Workbench opens.

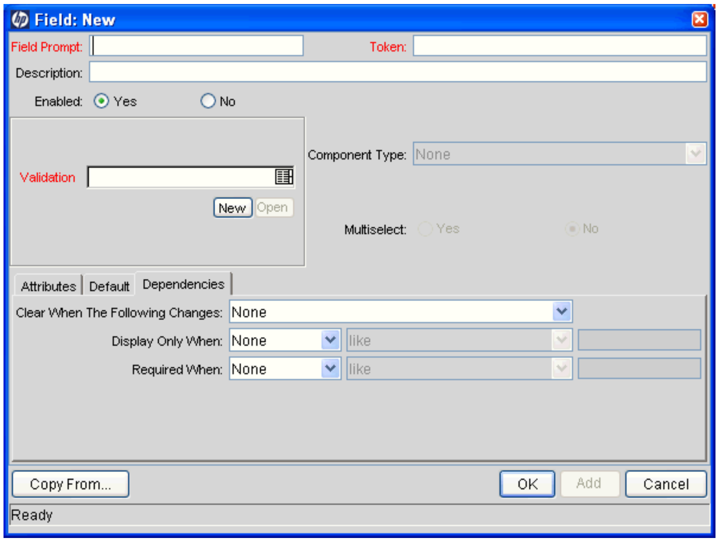
2. Select an existing user data type or create a new user data type.

   The User Data Context window opens to the **Fields** tab.

3. Select the field, and then click **Edit.**

   The Field window opens.

4. Click the **Dependencies** tab.



Field Prompt:

Token:

Description:

Enabled: ● Yes ○ No

Validation

Component Type: None

Multiselect: ○ Yes ● No

Clear When The Following Changes: None

Display Only When: None like

Required When: None like

5. Set the field dependencies, as follows:

   - In the **Clear When** field, select a field name to indicate that the current field is to be cleared if the selected field changes.

   - In the **Display Only When** field, select a field name to display the field only (for example, not editable) if specific logical criteria are satisfied. This field functions with a list that contain logical qualifiers and with another field that dynamically changes to a date field, list, or text field, depending on the validation for the selected field.

   - In the **Required When** field, select a field name to indicate that the field is to be required if certain logical criteria are satisfied. This field functions with a list that contains logical qualifiers and with field that dynamically changes to a date field, list, or text field, depending on the validation for the selected.

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. From the Workbench shortcut bar, select **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.

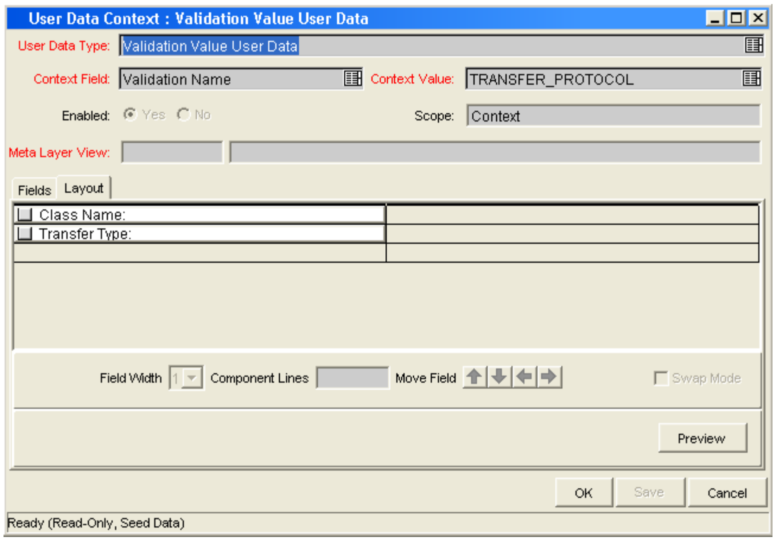
3. Select the field to remove, and then click **Remove.**

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.

Figure 9-2. User Data window Layout tab



User Data Context : Validation Value User Data

User Data Type: Validation Value User Data

Context Field: Validation Name

Context Value: TRANSFER_PROTOCOL

Enabled: ● Yes ○ No

Scope: Context

Meta Layer View:

□ Class Name:

□ Transfer Type:

Field Width 1

Component Lines

□ Swap Mode

Ready (Read-Only, Seed Data)

Changing Column Widths

To change the column width of a field:

1. From the Workbench shortcut bar, select **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. In the **Field Width** field, select either **1** or **2** inches.

   The Layout editor does not let you 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.

   Additionally, for fields of component type Text Area, it is possible to determine the number of lines the text area will 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 attribute is blank and read-only.

6. Click **OK.**

Moving Fields

To move a field or a set of fields:

1. From the Workbench shortcut bar, select **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 a adjacent fields.

   You cannot move a field to where another field exists.

5. Use the arrow pointers to move the fields in the layout builder.
6. Click **OK.**

Swapping Positions of Two Fields

To swap the positions of two fields:

1. From the Workbench shortcut bar, select **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

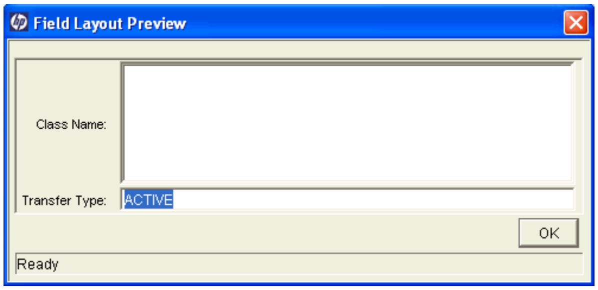
You can check to see what the layout looks like to users.

To preview field layout:

- In the User Data Content window, on the **Layout** tab, click **Preview.**

  A window shows a preview of the fields as they are to be displayed.

Figure 9-3. Preview mode



Class Name:

Transfer Type: ACTIVE

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.

Hidden fields do not affect the layout.

A Worksheets

In This Appendix:

Configuration Workflow Worksheets

Table A-1. Workflow skeleton

#	Step Name	Description	Type	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 ____

Step Field	Value
Step Name	
Goal/Result of Step	
Validation	See *Table A-3*.
Execution Type	See *Table A-4*.
Processing Type	
Timeout (Days)	
Source Environment (Group)	
Dest Environment (Group)	
Security (who can act on step): ■ User Name ■ Standard Token ■ User Defined Token	
Include Notification (Yes/No)	
Notification Event	
Notification Recipient: ■ Username ■ Email Address ■ Security Group ■ Standard Token ■ User Defined Token	
Notification Message	
Request Status at Step	
Request % Complete at Step	
Authentication Required (Y/N)	
Authentication Type (if Y)	

Table A-3. Workflow step [execution], step number ____ validation

Validation Information	Value
Existing Validation?	
New Validation?	
Validation Type: (text field, auto-complete, drop-down 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: ■ 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 ____

Step	Value
Step Name	
Goal/Result of Step	
Validation	See *Table A-6*.
Decisions Required (Vote on Step's outcome?)	■ One ■ At Least One ■ All
Timeout (Days)	
Security (who can act on step): ■ Security Group ■ User Name ■ Standard Token ■ User Defined Token	
Include Notification (Yes/No)	
Notification Event	
Notification Recipient: ■ Username ■ Email Address ■ Security Group ■ Standard Token ■ User Defined Token	
Notification Message	
Request Status at Step	
Request % Complete at Step	
Authentication Required (Y/N)	
Authentication Type (if Y)	

Table A-6. Workflow step [decision], step number ____ validation

Validation Information	Value
Existing Validation?	
New Validation?	
Validation Type: (text field, auto-complete, drop-down list, and so on)	
Validation Definition (list of values or SQL)	

Subworkflow Workflow Step Worksheets

Table A-7. Workflow step [subworkflow], step number ____ (page 1 of 2)

Step	Definition
Step Name	
Goal/Result of Step	
Validation*	
Vote on Step's outcome?	
Timeout (Days)	
Source Environment (Group)	
Dest Environment (Group)	
Security (who can act on step): ■ Security Group ■ User Name ■ Standard Token ■ User Defined Token	
Include Notification (Yes/No)	
Notification Event	

Table A-7. Workflow step [subworkflow], step number ____ (page 2 of 2)

Step	Definition
Notification Recipient: - Username - Email Address - Security Group - Standard Token User Defined Token	
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, drop-down list, and so on)	
Validation Definition (list of values or SQL)	

Object Type Configuration Sheets

Table A-9. Object type information

Field Name	Value
Object Type Name	
Description	

Table A-10. Object type fields

#	Field Name	Description
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		
16		
17		

Table A-11. Object type commands

Command	Definition
Goal of Commands	
Command Steps	
Conditions (When to execute)	

Table A-12. Validation field information

Field Name	Definition
Field Name	
Validation	
Field Behavior	
Attributes (select one):	■ Display ■ Editable ■ Display Only ■ Required
Default Value	
Dependencies	
Clear field when	
Display only when	
Required when	

Table A-13. Field validation information

Validation	Definition
Existing Validation?	
New Validation?	
Validation Type: (text field, auto-complete, drop-down list, and so on.)	
Validation Definition (list of values or SQL)	

Table A-14. Object type field information

Field Name	Definition
Field Name	
Validation	See *Table A-15*.
Field Behavior	
Attributes:	Display Editable Display Only Required
Dependencies	
Clear field when	
Display only when	
Required when	
Default Value	

Table A-15. Field Validation information

Field Validation	Definition
Existing Validation	
New Validation?	
Validation Type (text field, auto-complete, drop-down list, and so on)	
Validation Definition (list of values or SQL)	

Example of Completed Object Type Configuration Sheets

The following tables are an example of a completed set of object type configuration sheets.

Table A-16. Example of object type information

Field Name	Value
Object Type Name	Dev to Test File Migration
Description	Deployment of a file from Dev server to Test server.

Table A-17. Example of object type fields

#	Field Name	Description
1	File Location	The name of the field. See *Table A-18* and *Table A-19*.
2	Sub Path	The name of the field. User this field to select the directory containing the file to be deployed. See *Table A-20* and *Table A-21*.
3	File Name	The name of the file. See *Table A-22* and *Table A-23*.

Table A-18. Example of validation for File Location (page 1 of 2)

Field Name	Definition
Field Name	File Location
Validation	See *Table A-19*.
Field Behavior	
Attributes:	■ Display: Yes, the field is visible on the package line. ■ Editable: Yes, the field can be edited after the package is submitted. ■ Display Only: Never. This field can be edited. If set to Always, you can never edit this field. ■ Required: Always. This field must contain a value.
Default Value	None
Dependencies	

Table A-18. Example of validation for File Location (page 2 of 2)

Field Name	Definition
Clear field when	None
Display only when	None
Required when	None

Table A-19. Field validation for File Location

Validation	Definition
Existing Validation?	DLV - File Location
New Validation?	Not Applicable
Validation Type (text field, auto-complete, drop-down list, and so on)	Drop-down list.
Validation Definition (list of values or SQL)	SQL

Table A-20. Example of validation for Sub Path

Field Name	Definition
Field Name	Sub Path
Validation	See *Table A-21*.
Field Behavior	
Attributes:	■ Display: Yes, the field is visible on the package line. ■ Editable: Yes, the field can be edited after the package is submitted. ■ Display Only: Never. This field can be edited. If set to Always, you can never edit this field. ■ Required: Always. This field must contain a value.
Default Value	None
Dependencies	
Clear field when	None
Display only when	None
Required when	None

Table A-21. Field validation for Sub Path

Validation	Definition
Existing Validation?	Directory Chooser
New Validation?	Not Applicable
Validation Type (text field, auto-complete, drop-down list, and so on)	Directory Chooser
Validation Definition (list of values or SQL)	Default behavior

Table A-22. Example of validation for File Name

Field Name	Definition
Field Name	File Name
Validation	See *Table A-23*.
Field Behavior	
Attributes:	■ Display: Yes, the field is visible on the package line. ■ Editable: Yes, the field can be edited after the package is submitted. ■ Display Only: Never. This field can be edited. If set to Always, you can never edit this field. ■ Required: Always. This field must contain a value.
Default Value	None
Dependencies	
Clear field when	None
Display only when	None
Required when	None

Table A-23. Field validation for File Name

Validation	Definition
Existing Validation?	File Chooser - Full File Name
New Validation?	Not Applicable
Validation Type (text field, auto-complete, drop-down list, and so on)	File Chooser
Validation Definition (list of values or SQL)	Default Behavior

Index