Mercury IT Governance Center™

Mercury Demand Management™ Configuration Guide

Version: 7.0

Mercury
379 North Whisman Road
Mountain View, CA 94043
http://www.mercury.com

If you have any comments or suggestions regarding this document, please send email to documentation@mercury.com.

Publication Number: ITGDemandMgmtCG7.0/02

Table of Contents

List of Figures

List of Tables

Chapter

Getting Started with Mercury Demand Management Configuration

In This Chapter:

Introduction to Mercury Demand Management

Mercury Demand Management™ is a Mercury IT Governance Center™ product that automates your business processes. At the core of this functionality is an integrated workflow engine that lets you digitize both simple and complex processes. Demand Management works by capturing requests and processing the requests based upon the processes and business rules created for that type of request. The process behind each request is modeled, automated, enforced, and measured on your best-practice business processes. In addition, a detailed audit trail helps you pinpoint problems quickly and supports regulatory compliance requirements, such as segregation of duties (SOD), at both the role level and the process step level.

End users complete a configurable request form using a standard Web browser. Each type of request type has an associated workflow that specifies the process for reviewing, evaluating, prioritizing, scheduling, and resolving the request. Based on the workflow, the reviewer can assign the request to a person or team for scheduling and delivery. Notifications defined as part of the process can be activated at any step in the process to indicate work is to be done, hasn't been done, is being escalated, or most any other reason. Included with Demand Management is the Web-based Mercury IT Governance Dashboard™, which delivers the right information to anyone with a browser.

This document provides the details on how to configure a Mercury Demand Management system using the Mercury IT Governance Workbench, and includes the information you need to ensure that your requests follow your digitized business processes. Chapter 1, *Getting Started with Mercury Demand Management Configuration,* on page 13 (this chapter) provides an overview of how to configure Mercury Demand Management to support your business processes.

Mercury Demand Management Concepts

- **Contacts.** Contacts are Mercury Demand Management users used as points of reference or information by other Mercury Demand Management entities, such as requests.
- **Notification Templates.** Notification templates are pre-configured notification forms that can be selected and used with the various Mercury Demand Management entities, such as workflows and requests.

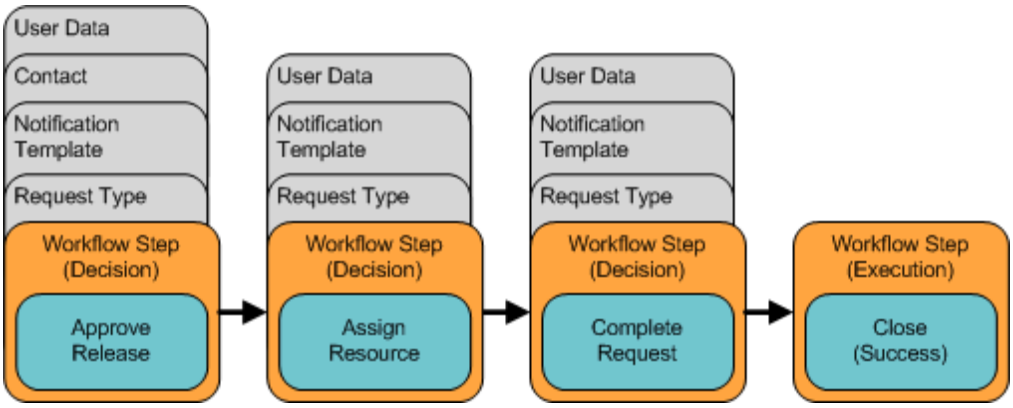
- **Request Resolutions.** Request resolution refers to the creation, processing and closing of requests. A request can be anything from a simple question to a detailed report of a software bug.
- **Requests.** A request is the fundamental work unit of the request resolution piece of Mercury Demand Management. End-users create requests and submit them along a resolution process (workflow). The request contains all information typically required to complete a specific business process.
- **Request Types.** Request types define the structure and logic of requests. Mercury Demand Management includes such predefined system request types as the Bug request type and Enhancement request type.
- **User Data.** Mercury Demand Management entities such as requests 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 custom "user data" fields to capture additional information specific to your business process.
- **Workflows.** 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 demands.

Overview of a Simplified Demand Management Process

Figure 1-1 illustrates a simple four step Demand Management process to approve a release. The first step, **Approve Release,** is a decision step where a user is notified that a decision needs to be made, in this case, should a release be approved. Once manually approved the process moves to the second step.

The second step, **Assign Resource,** is decision workflow step where a resource must be manually assigned to the release. Once a resource is assigned and the step is complete, the process moves to the third step.

Figure 1-1. Mercury IT Governance Center components



The third step, **Complete Request,** is a decision step where the assigned resource completes the request. Once the request is completed, the process moves to the four and final step.

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

Overview of Configuring Demand Management

A Mercury Demand Management system can be configured using the following steps:

Step 1: Gather process requirements

Before configuring a Mercury Demand Management system, you should collect specific information concerning your process, the types of requests required, and your contacts.

For detailed information, see Chapter 2, *Gathering Process Requirements,* on page 19. Appendix A, *Worksheets,* on page 251 also provides a series of worksheets to help gather the information required to build a Demand Management 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 43 and Chapter 4, *Configuring Workflow Components,* on page 105. Appendix A, *Worksheets,* on page 251 also provides a series of worksheets to help gather the information required to configure a workflow.

Step 3: Configure request types

Request types gather and track the information required to perform workflow steps. For information on how to configure request types, see Chapter 5, *Configuring Request Types and Request Header Types,* on page 137. Appendix A, *Worksheets,* on page 251 also provides a series of worksheets to help gather the information required to configure a request type.

Step 4: Configure contacts

Contacts are Mercury Demand Management users used as points of reference or information by other Mercury Demand Management entities, such as requests. For information about how to configure contacts, see Chapter 6, *Configuring Contacts,* on page 215.

Step 5: Configure notification templates

Notification templates are pre-configured notification "formats" used by other Mercury Demand Management entities, such as workflows and request types. Chapter 7, *Configuring Notification Templates,* on page 219 discusses how to configure notification templates.

Step 6: Configure user data fields

User data fields add custom data fields used by other Mercury Demand Management entities, such as workflows and request types. Chapter 8, *Configuring User Data,* on page 231 discusses how to configure these user data fields.

Step 7: Configure your security and access requirements

Part of any process are the permissions required to perform various decisions. Two of the ways in which Mercury IT Governance Center controls access to perform these decisions are by licenses and access grants.

- **Licenses.** Licenses provide users with access to a Mercury IT Governance Center product such as Mercury Demand 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 Mercury IT Governance Center product.

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

When You're Ready to deploy Demand Management: Educate your users

Once your Demand 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 Demand Management system users:

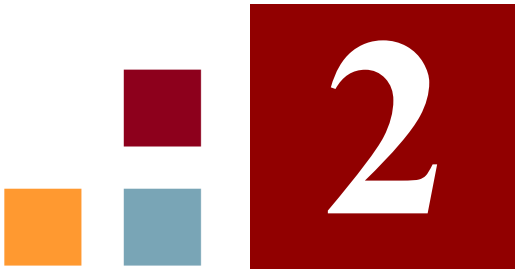
- **Basic Mercury Demand Management training.** Ensure that each user understands how to create, process, and report on requests.
- **Process-specific training.** Ensure that each user understands the new process. Consider holding a formal meeting or publishing documents on the workflow steps and requests.
- **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 release. You can use email notifications that are part of Demand 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 Mercury Demand Management:

- *Mercury Demand 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*
- *Configuring the Standard Interface*
- *Mercury-Supplied Entities Guide* (includes descriptions of all Mercury Demand Management portlets, request types, and workflows)

Chapter



Gathering Process Requirements

In This Chapter:

Overview of Gathering Process Requirements

This chapter discusses the following information that needs to be collected Before configuring a Mercury Demand Management process, you should gather information concerning your process, such as the steps in the workflow, the types of requests required, and contacts. Once this information is collected, you can then begin configuring your Mercury Demand Management process.

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

- **Defining workflows.** What are the steps of your Demand Management process (workflow)? Which steps require manual decisions (reviews and approvals)? Which steps require automatic executions? See *Defining Workflows* on page 21 for detailed information.
- **Defining request types.** What are you requesting? See *Defining Request Types* on page 30 for detailed information.
- **Defining contacts.** What contacts are required? See *Defining Contacts* on page 37 for detailed information.
- **Defining notification templates.** Is the correct notification template in place? Does your process require a new notification template? See *Defining Notification Templates* on page 38 for detailed information.
- **Defining user data fields.** Does your process require additional user information to process correctly? See *Defining User Data Fields* on page 38 for detailed information.
- **Defining security and access.** Who will be allowed to submit requests? Who should receive notifications? Who will be allowed to approve the request at each step? See *Defining Security and Access* on page 39 for detailed information.

Defining Workflows

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

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

- Are you designing a simple approval process with little oversight or supervision?
- Are you designing a business-wide bug-tracking system with a great deal of oversight and supervision?

Once you understand the intent of the business process, you can then begin to define the 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. Mercury Demand 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 automatic time-stamping or automatic request 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 251 for a complete list of decision step considerations.

Table 2-1. Decision Workflow Checklist (page 1 of 2)

	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 request 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 (page 2 of 2)

	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 the workflow step?

Gathering Information for Execution Steps

Execution steps are workflow steps that perform actual work or actions, such as automatic times-stamping or automatic request status changes. *Table 2-2* provides a configuration consideration checklist to help define execution steps. See Appendix A, *Worksheets,* on page 251 for a complete list of execution step considerations.

Table 2-2. Execution Workflow Checklist (page 1 of 2)

	Execution Step Check Item	Example
	What is the name of this workflow step?	■ Create Request ■ Close ■ Set Temp Date
	Will this workflow step execute this command?	■ Cancel request ■ Update request
	What is the execution type?	■ Close ■ Jump ■ Return from Subworkflow
	What is the processing type?	■ Immediate ■ Manual
	What is the source environment (group)?	KINTANA_SERVER
	What is the destination environment (group)?	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

Table 2-2. Execution Workflow Checklist (page 2 of 2)

	Execution Step Check Item	Example
	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 (page 1 of 2)

	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

Table 2-3. Condition Workflow Checklist (page 2 of 2)

	Condition Step Check Item	Example
	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 251 for a complete list of subworkflow step considerations.

Table 2-4. Subworkflow Workflow Checklist

	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

Table 2-4. Subworkflow Workflow Checklist

	Subworkflow Step Check Item	Example
	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 workflow.

Table 2-5. Workflow logical guidelines (page 1 of 3)

	Guideline	Reason
	Workflows	
	Make one or more workflows available to process the request.	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 request 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 request will not close if a transition exists on ‘Success.’

Table 2-5. Workflow logical guidelines (page 2 of 3)

	Guideline	Reason
	Close step in subworkflow closes entire request.	Do not include a Close step in a subworkflow unless you want to close the workflow in the subworkflow.
	All Steps	
	All steps must be enabled.	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.
	'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 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.

Table 2-5. Workflow logical guidelines (page 3 of 3)

	Guideline	Reason
	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.

Workflow and Request Interaction

Request statuses can change as the request moves through its resolution process. Each request status can control its request field attributes, such as whether or not the field is visible or editable. A request status can be tied to a workflow step, which means that when a request reaches a certain workflow step, it acquires a status that determines a field's attributes. The request status at a particular workflow step can also drive field logic during the life of the request.

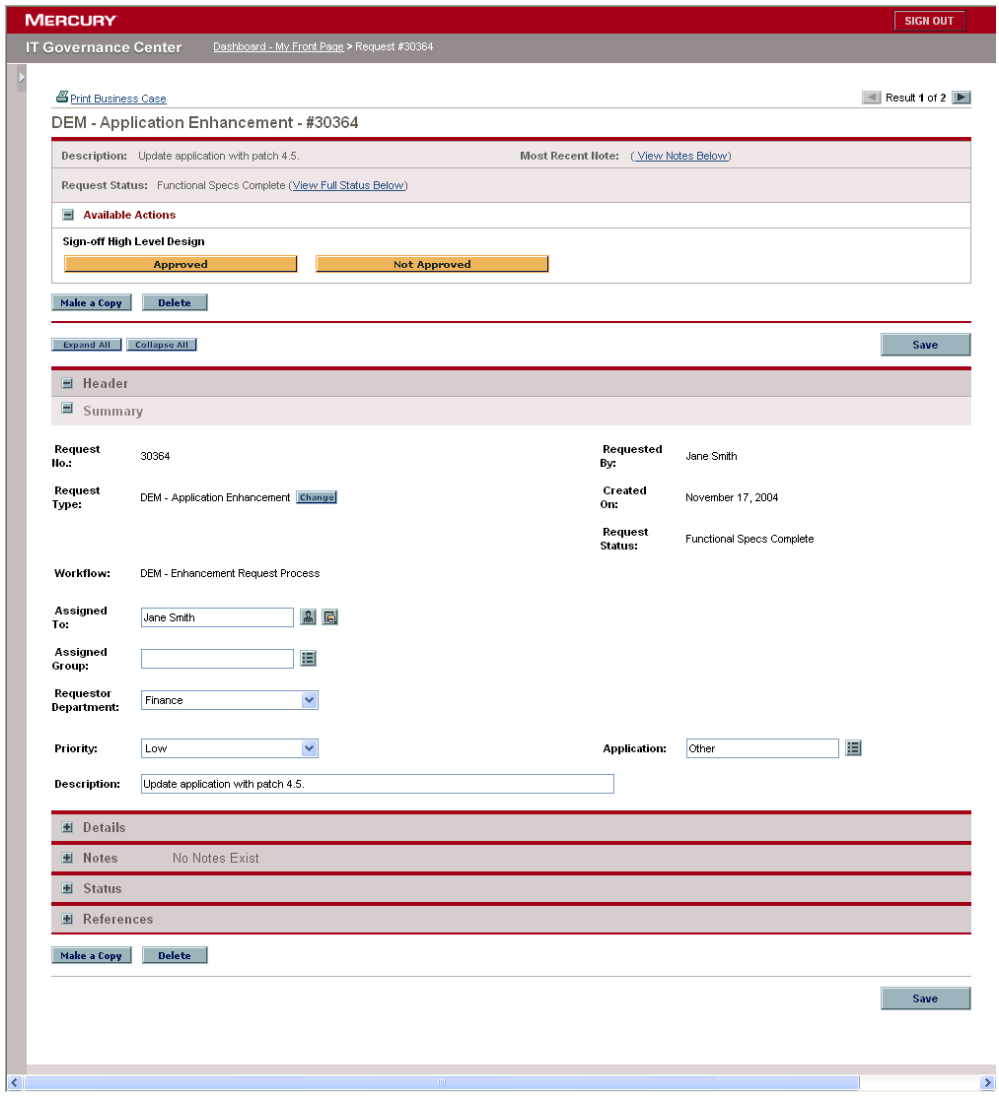
In most cases, a single request type is associated with a single workflow. Information contained in the request (which is defined in the request type) works in conjunction with the workflow process to ensure that the request is correctly processed. While it is possible to use one workflow with many different request types, the level of possible integration between request type and workflow is easier with a one-to-one mapping.

Defining Request Types

Requests correspond to request types. Request types define the structure and logic of requests using fields on the request type. Request types also include a request header type which supplies a standard set of fields to each request type. Additionally, selectable field groups add standard sets of fields to a request type, allowing standardized support for other Mercury products such as Mercury Universal Configuration Management Database, Mercury Application Mapping Monitor, Mercury Program Management, Mercury Project Management, and Mercury Portfolio Management.

Each request requires different information to process it. For example, to resolve a software bug, you might need to know the software unit, product version, problem, and priority. This information is captured using fields on the request type (see *Figure 2-1*).

Figure 2-1. Sample request



When creating a request type, first check to see what standard fields are available for the request (request header types and fields groups). In *Figure 2-1*, the fields included under the **Summary** banner are from the request type's request header type.

For each request type, gather the following information:

- **Request Name.** The name of the request and request type.
- **Request Header Type.** What request header type is attached to this request.
- **Request Fields.** What fields should appear on the request?

- **Request Statuses.** What statuses should the request have, such as **Pending, On Hold, Approved,** and **Canceled?**
- **Status Dependencies.** What fields should have request status dependencies, such as the **Assigned To** field only being required when the status of the request is **Assign.**
- **Notifications.** What notifications should be sent when the value of a selected field changes?
- **User Access.** Who is allowed to edit and submit this request?
- **Workflows.** Which workflows can be attached to this request?

If your request requires new fields, gather the following information for each field:

- **Field name.** Specify a field prompt that helps to ensure that the correct information is captured.
- **Information type.** What type of information needs to be collected? Should the field be a text field, a drop-down list, an auto-complete? A field's validation determines the field's component type, such as text field and drop-down list.
- **Field behavior.** Many aspects of a field's behavior can be controlled including:
  - The field can be configured to be editable, non-editable, required, hidden and so on.
  - The field can be configured to automatically populate based on values in other fields.

See Appendix A, *Worksheets,* on page 251 for more information concerning request types and request type fields.

Request and Workflow Interaction

Request statuses can change as the request moves through its resolution process. Each request status can control its request field attributes, such as whether or not the field is visible or editable. A request status can be tied to a workflow step, which means that when a request reaches a certain workflow step, it acquires a status that determines a field's attributes. The request status at a particular workflow step can also drive field logic during the life of the request.

In most cases, a single request type is associated with a single workflow. Information contained in the request (which is defined in the request type) works in conjunction with the workflow process to ensure that the request is correctly processed. While it is possible to use one workflow with many different request types, the level of possible integration between request type and workflow is easier with a one-to-one mapping.

Request Type Checklist

Table 2-6 provides a configuration consideration checklist to help define your Demand Management system. See Appendix A, *Worksheets,* on page 251 for a complete list of request type considerations.

Table 2-6. Request type configuration checklist (page 1 of 3)

	Request Type Check Item	Configuration Consideration
	Request type considerations.	A request type must be defined for each type of request to be resolved. This includes creating fields that describe the request and decisions and field logic required to process it during resolution.
	Is a request header type associated to the request type?	A request header type must be associated with the request type. If no satisfactory request header type exists, a new one should be created.
	Are fields defined?	■ Fields are required to define the request. ■ Ensure the correct parameters are used to describe the request to be processed. ■ See Chapter 5, *Configuring Request Types and Request Header Types,* on page 137 ■ See *Commands, Tokens, and Validations Guide and Reference*
	Are request rules defined?	■ Rules can be set for the automatic population of fields in the request. ■ For details, see Chapter 5, *Configuring Request Types and Request Header Types,* on page 137.
	Are request status values defined?	■ The statuses the request can take on should be defined and associated with the request type. New statuses can be added to the list if necessary. ■ For details, see Chapter 5, *Configuring Request Types and Request Header Types,* on page 137.

Table 2-6. Request type configuration checklist (page 2 of 3)

	Request Type Check Item	Configuration Consideration
	Are status dependencies set?	■ Request fields can be configured to be hidden, required, read-only, cleared, or reconfirmed based on the status of the request. ■ For details, see Chapter 5, *Configuring Request Types and Request Header Types,* on page 137.
	Is request security set?	■ You can exercise a great deal of control over who participates in request resolution. ■ See Chapter 5, *Configuring Request Types and Request Header Types,* on page 137 ■ See *Security Model Guide and Reference*.
	Is request field security set?	■ Request fields can be configured to be hidden to particular users or security groups. ■ See Chapter 5, *Configuring Request Types and Request Header Types,* on page 137 ■ See *Security Model Guide and Reference*
	Are request notifications set?	■ Notifications can be configured to be sent automatically at various points in your process. ■ See Chapter 5, *Configuring Request Types and Request Header Types,* on page 137.
	Are user data fields defined?	■ User data fields can be used to track specific information across a Workbench entity. ■ See Chapter 5, *Configuring Request Types and Request Header Types,* on page 137.

Table 2-6. Request type configuration checklist (page 3 of 3)

	Request Type Check Item	Configuration Consideration
	Are fields defined for the request type?	■ Fields are required to define the request. Make sure that the correct parameters describe the request. ■ See Chapter 5, *Configuring Request Types and Request Header Types,* on page 137
	Is the request type enabled?	Disabled request types cannot be used.
	Request type and workflow considerations.	■ Decide which request type status values correspond to which workflow steps. ■ Decide which workflow steps to execute any request type commands. ■ Check to make sure that workflow step source validations and request type field validations agree. This is required if a transition is based on a field value (using token, SQL or PL/SQL execution types). ■ Allow the request type use for the workflow (set on the workflow window **Request Types** tab). ■ Allow the workflow to be used by the request type (set on the request type window **Workflows** tab).

Defining Contacts

Contacts are Mercury Demand Management users used as points of reference or information by other Mercury Demand Management entities, such as requests. For information concerning the configuration of contacts, see Chapter 6, *Configuring Contacts,* on page 215.

For each contact, gather the following information:

- First name
- Last name
- Username
- Phone Number
- Email Address
- Company

Contacts Checklist

Table 2-7 provides a configuration consideration checklist to help define your contacts.

Table 2-7. Contacts checklist

	Contacts Check Item	Configuration Consideration
	Is the contact enabled?	Disabled contacts cannot be used.
	Is the contact an IT Governance Center user?	■ Leave the username field empty when the contact is not a IT Governance Center user. ■ The username field is a auto-complete field of IT Governance Center users.

Defining Notification Templates

Notification templates are pre-configured email forms that can be used to quickly construct the body of a message. Notification templates are used by entities of Mercury Demand Management, such as workflows and requests. When configuring a workflow, select the notification template you want to use for each workflow step. Mercury Demand 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 you business process. Chapter 7, *Configuring Notification Templates,* on page 219 provides detailed information concerning the configuration of notification templates.

Notification Templates Checklist

Table 2-8 provides a configuration consideration checklist to help define your notification templates.

Table 2-8. Notification template checklist

	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

Mercury Demand Management entities such as requests 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 8, *Configuring User Data,* on page 231 provides detailed information concerning the configuration of user data fields.

User Data Checklist

Table 2-9 provides a configuration consideration checklist to help define your user data fields.

Table 2-9. User data checklist

	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 Demand Management process are the permissions required to perform various decisions or executions. Two of the ways in which Mercury IT Governance Center controls access to perform decisions and executions are by the following:

- **Licenses.** Licenses provide users with access to a Mercury IT Governance Center products such as Mercury Demand 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 Mercury IT Governance Center product.

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

- License, Demand Management
  - Access Grant, View Requests - Those who can view requests
  - Access Grant, Edit Requests - Those who can edit requests

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

When designing Demand Management 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 Demand Management 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 2-10 provides an example of which security groups can access a Demand Management workflow and at which workflow step.

Table 2-10. Example of workflow security groups

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

Security and User Access Checklist

Table 2-11 provides a configuration consideration checklist to help define your security and user access requirements.

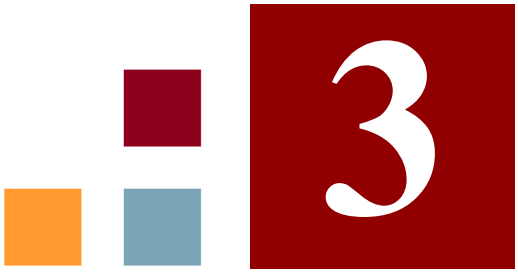
Table 2-11. Security and user access checklist (page 1 of 2)

	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 request creation	All available options for restricting who can create and submit requests have been set.

Table 2-11. Security and user access checklist (page 2 of 2)

	Security and User Access Check Item	Configuration Consideration
	Set security on request processing	All available options for restricting who can process requests have been set.
	Set security on request system configuration	You have specified who can modify the request process. This includes editing the workflow, object type, environment, security groups, and so on.
	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.

Chapter



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 Demand Management workflows.

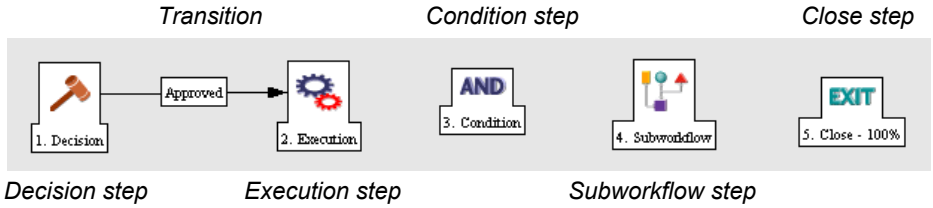
The following is a list of the basic components of a workflow:

- **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 Mercury IT Governance 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 Mercury IT Governance 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

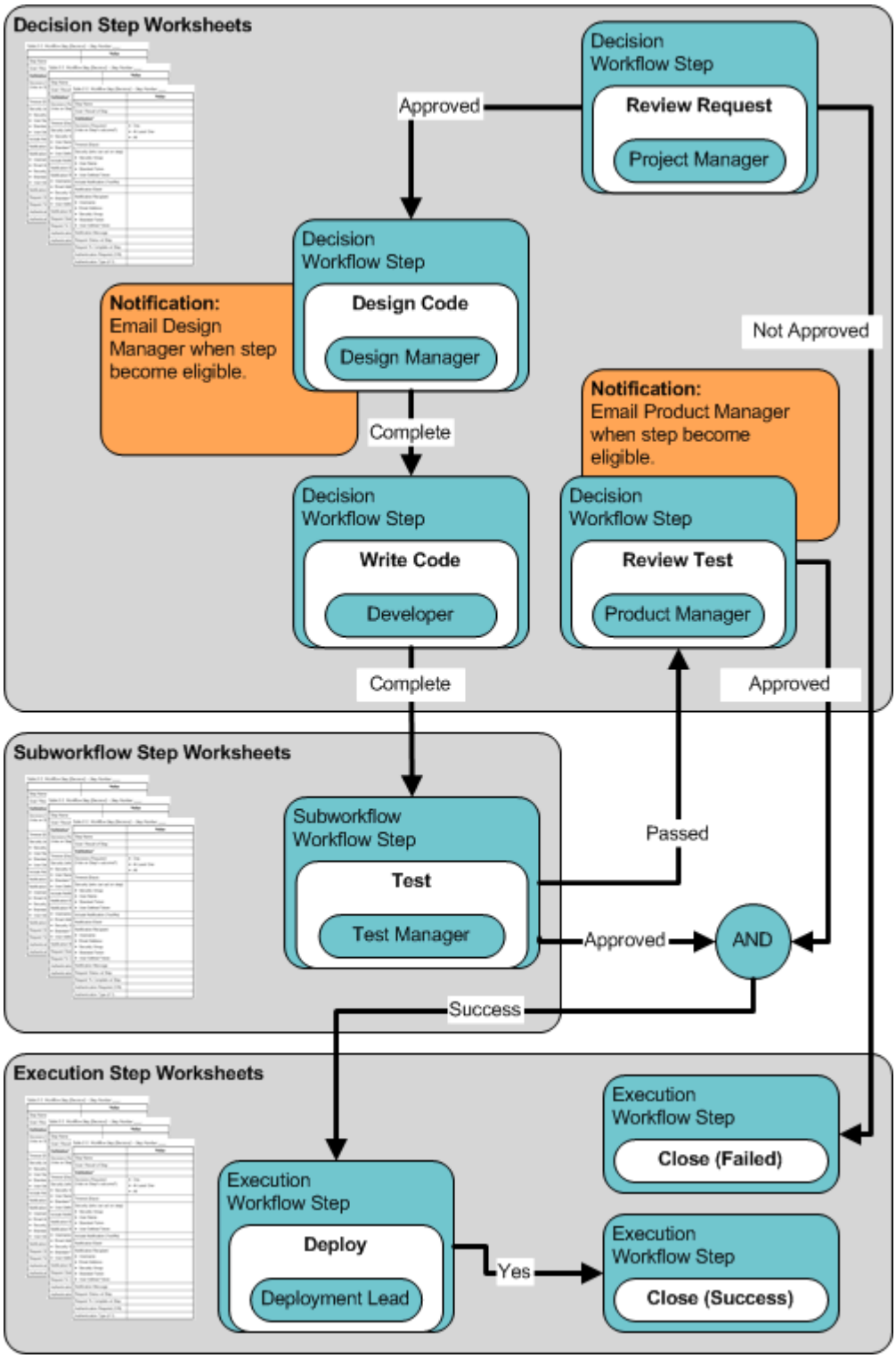
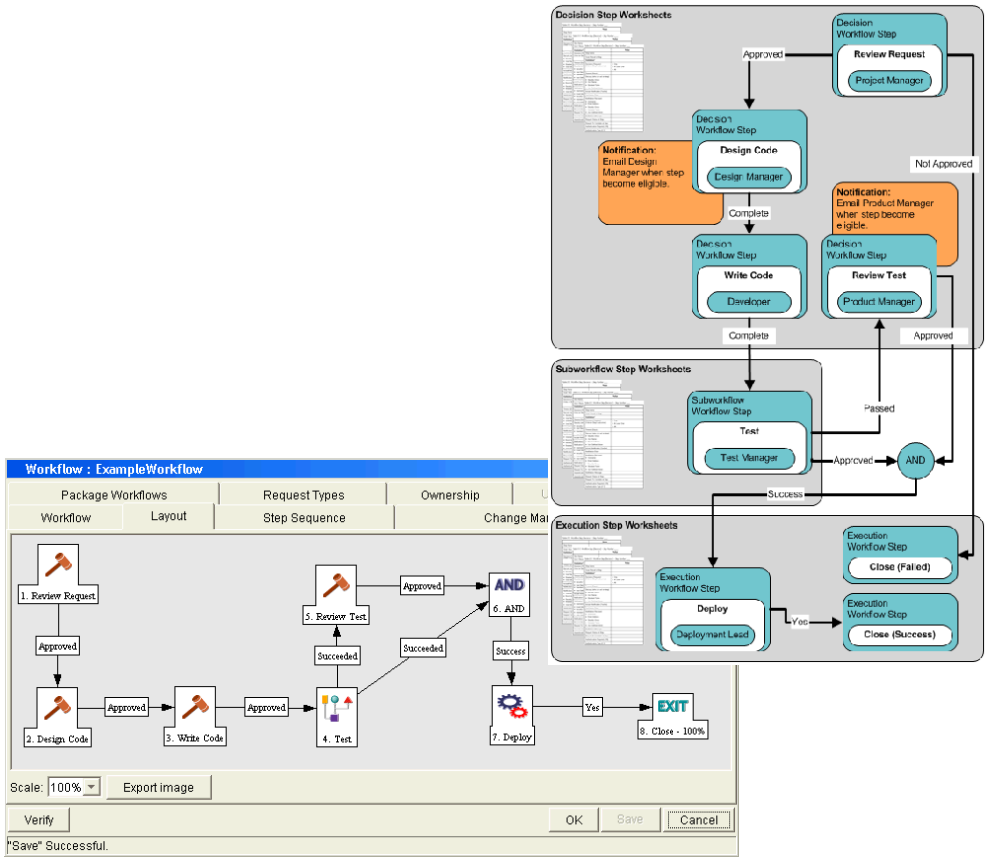


Figure 3-3. Stage 2. Create the workflow



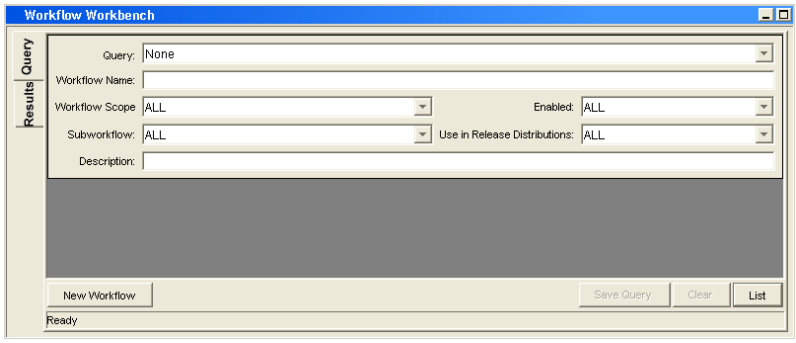
Opening the Workflow Workbench

To open the Workflow Workbench:

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

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

The Workflow Workbench opens.



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

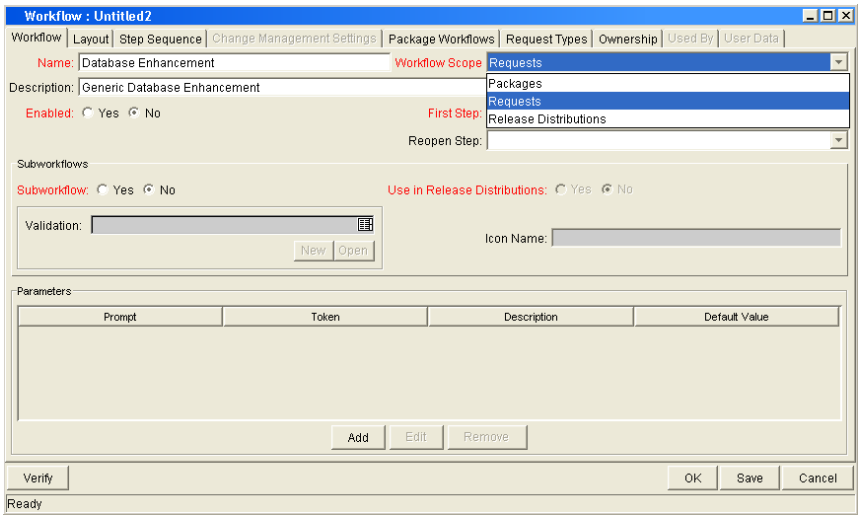
To enter basic workflow information:

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

   The Workflow Workbench opens.

2. Click **New Workflow.**

   The Workflow window opens.



3. In the **Name** field, type a name for the workflow.
4. In the **Workflow Scope** field, select **Requests.**
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

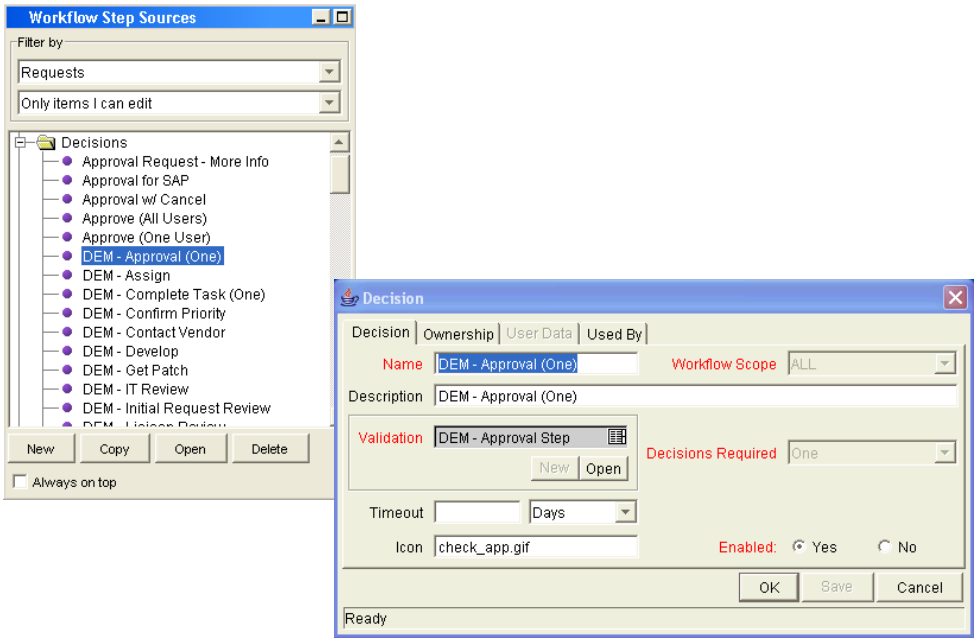
Choosing Workflow Steps

Mercury IT Governance 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

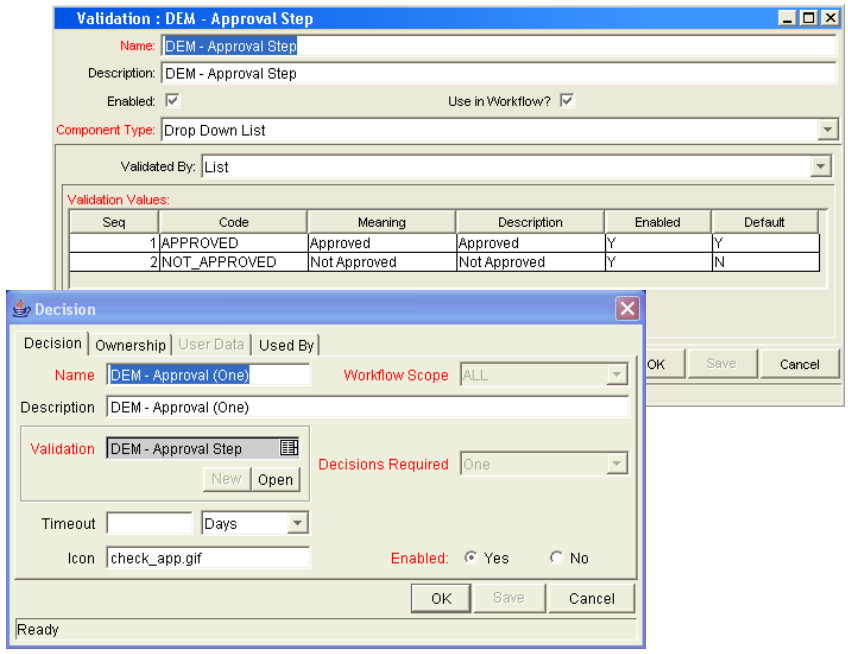
To evaluate a workflow step, determine which of the these workflow folders it corresponds to. Open the Workflow Step Source folder, and then open the workflow steps that most closely suit your needs (*Figure 3-4*).

Figure 3-4. Workflow step source



The validation values are the acceptable values a workflow step can have (see *Configuring Validations for Workflow Steps* on page 89). Check these to see if they meet your transition requirements.

Figure 3-5. Workflow step source validation



Overview of Decisions Workflow Steps

Decision workflow steps represent manual activities performed outside of Mercury IT Governance Center. Decision workflow steps include such activities as:

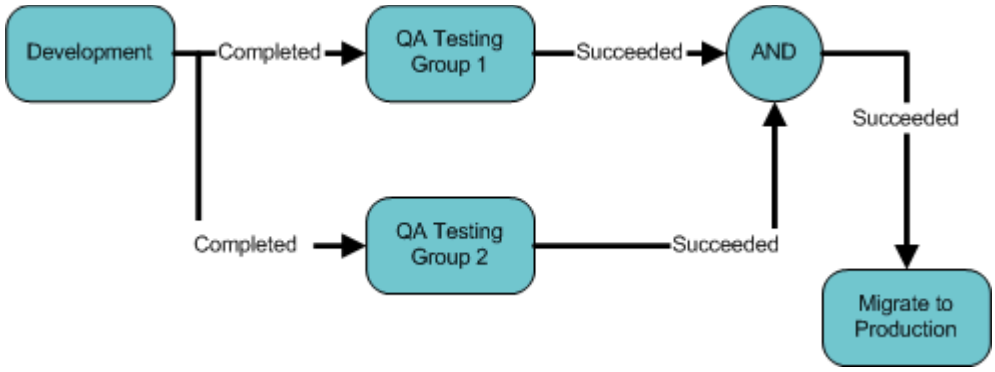
- Decisions made by committees
- Code designs and reviews

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:

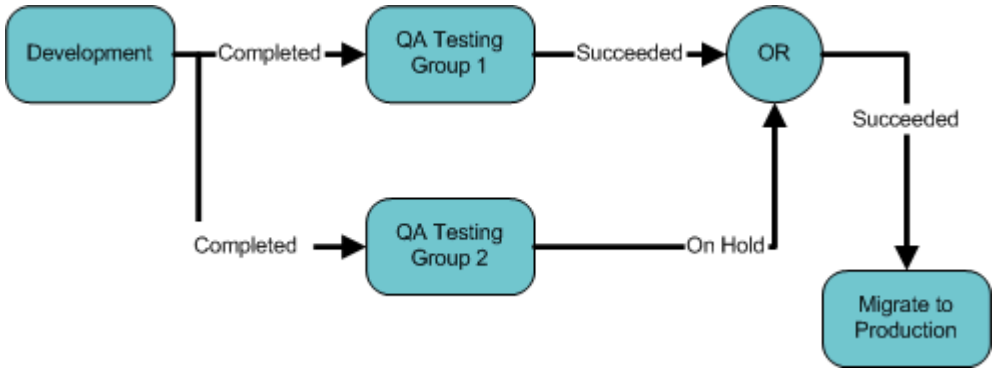
- **AND.** This condition is met only after all workflow steps leading to it reach the specified required status. *Figure 3-6* illustrates an example of the AND condition workflow step.

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



Overview of Execution Workflow Steps

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

- Request jump
- 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

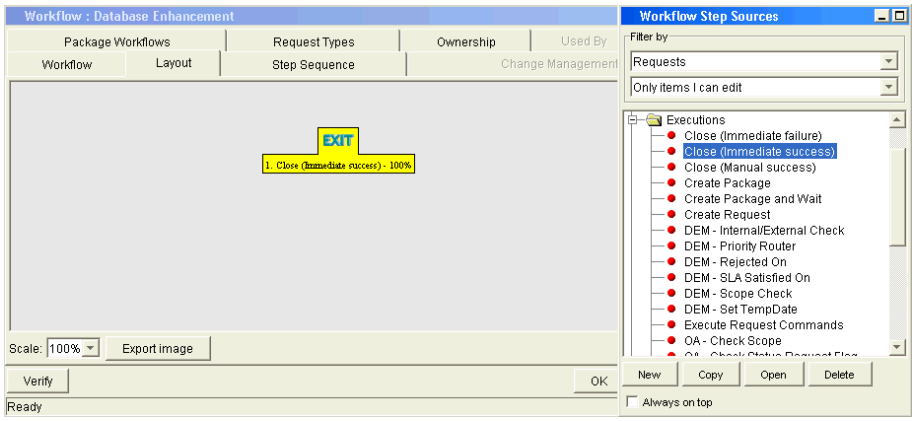
Regardless of how long or short it is, every workflow must include a close workflow step (see *Figure 3-8*). A close workflow step is a type of execution workflow step. You can find it in the Executions folder in the Workflow Step Sources window.

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 3-8*).

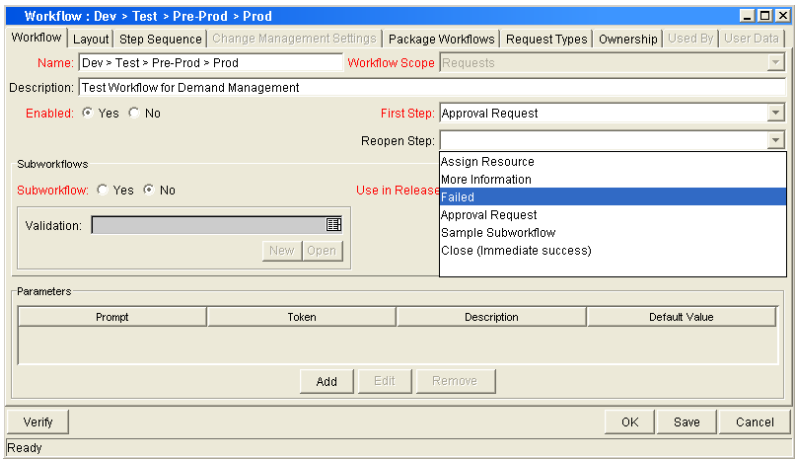
Figure 3-8. Close workflow step



Configuring Reopen Workflow Steps

Users with the required access grants can reopen closed requests. A reopened request begins at a workflow step specified as the reopen workflow step for the workflow.

Figure 3-9. Workflow window reopen step list

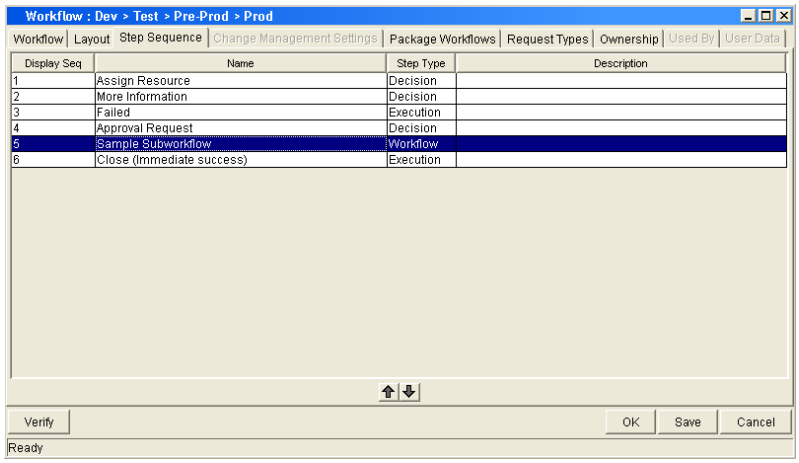


To specify a reopen workflow step for a workflow, click the **Workflow** tab in the Workflow Workbench (see *Figure 3-11* on page 55). In the **Reopen Step** field, select the reopen workflow step.

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.

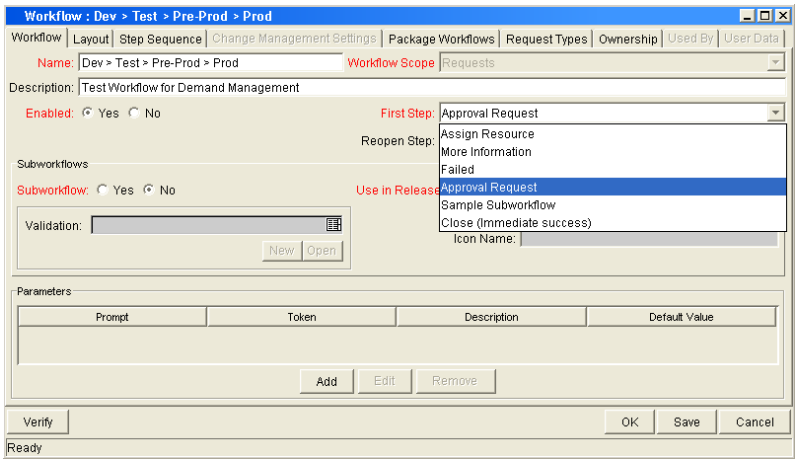
Figure 3-10. Step sequence tab



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 3-11*). In the **First Step** field, select the first step.

Figure 3-11. Workflow tab



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:

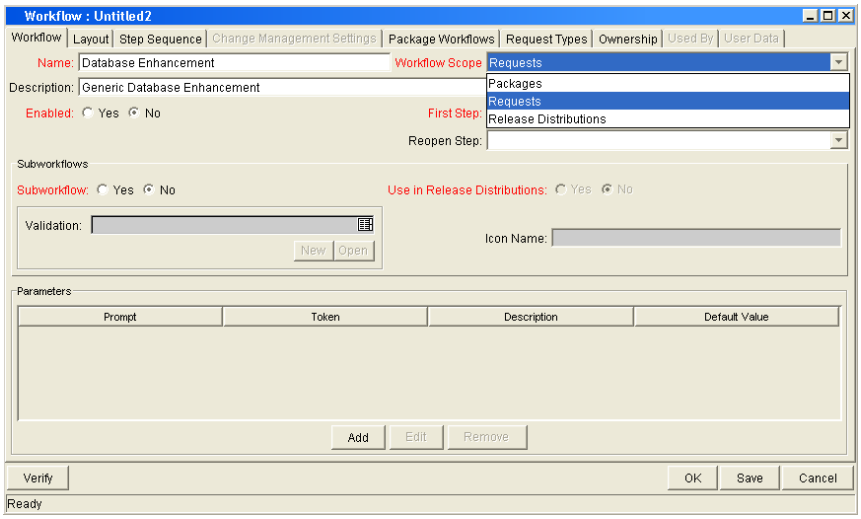
1. From the Workbench shortcut bar, select **Configuration > Workflows.**

The Workflow Workbench opens.

2. Open a workflow.

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

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



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.

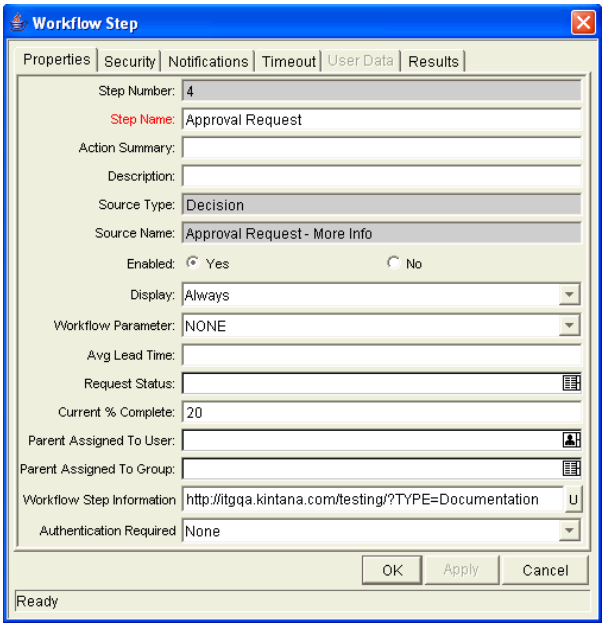
4. Right-click a workflow step.

   A menu window opens.

5. Click **Edit.**

   The Workflow Step window opens.

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



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.

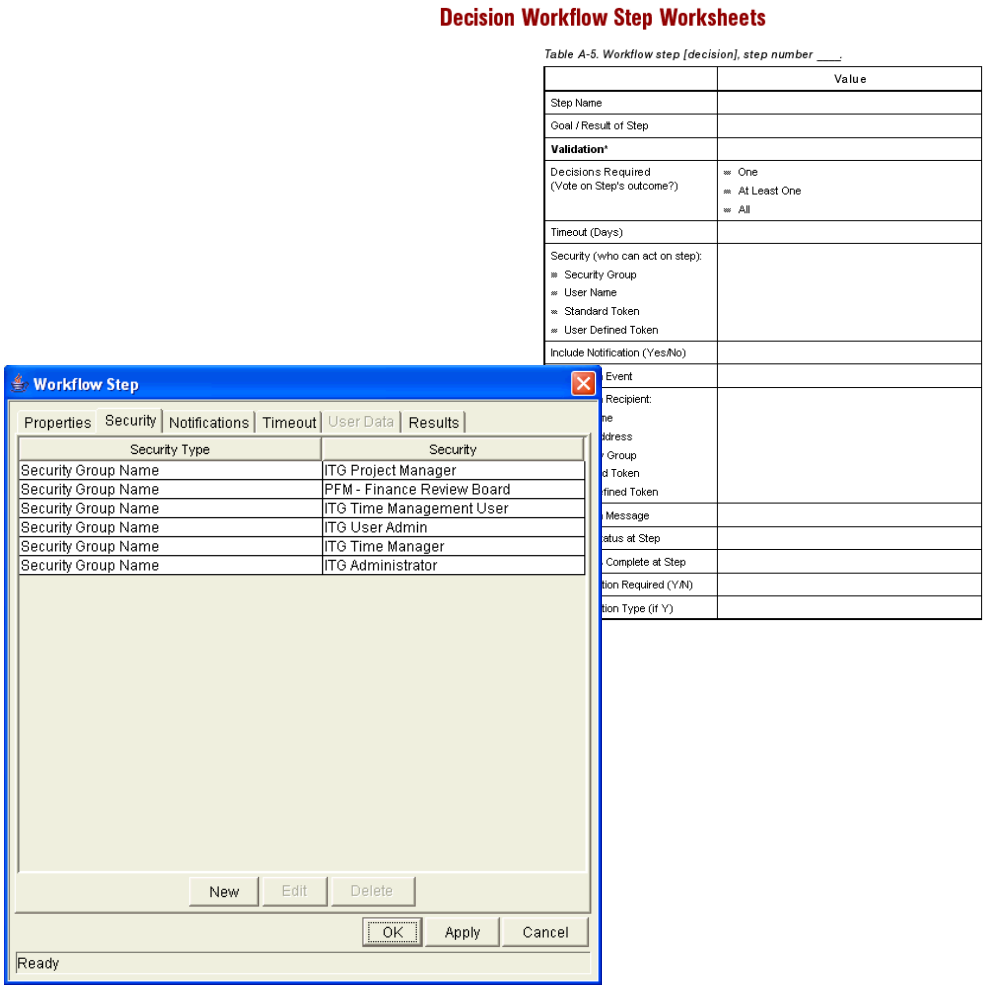
5. Click **Edit.**

   The Workflow Step window opens.

6. Click the **Security** tab.

7. Click **New.**

   The Workflow Step Security window opens.



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.**

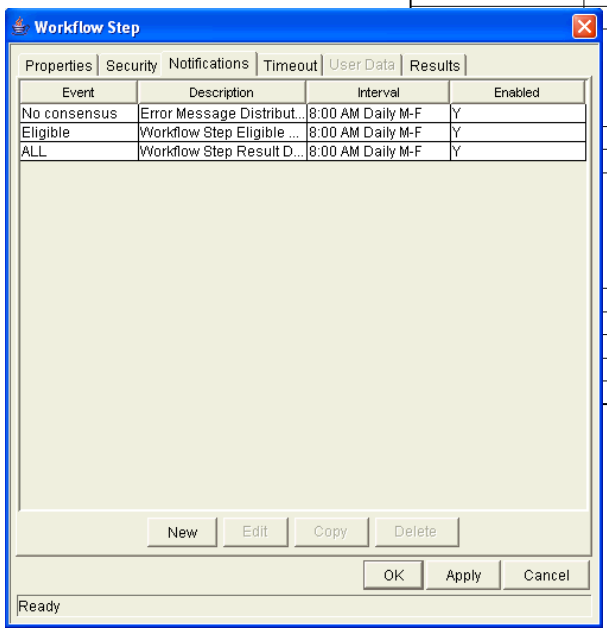
   The Workflow Step window opens.
6. Click the **Notifications** tab.
7. Click **New.**

   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.

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



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 62.

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	This field is 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 62.

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	This field is 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 62.

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 66.
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	This field is 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 62.

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 62. 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

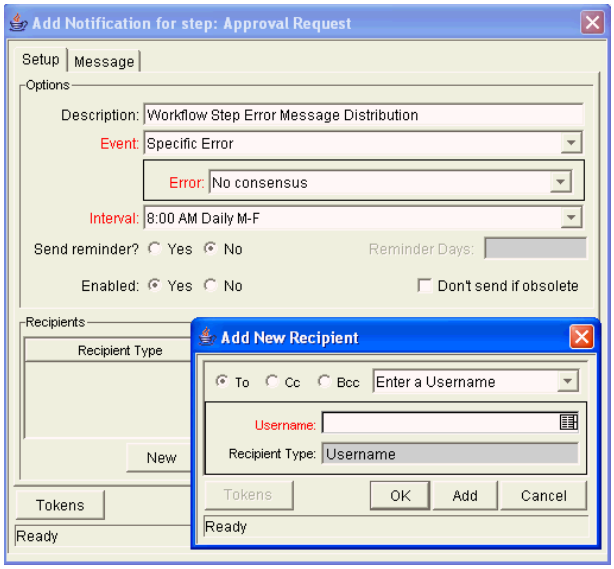
You must specify at least one recipient for a notification. The recipient can be a specific user, all members of a security group, or any email address.

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.



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 Mercury IT Governance Center and update the request's status. Additionally, the notification should include a link (URL) to the referenced request.

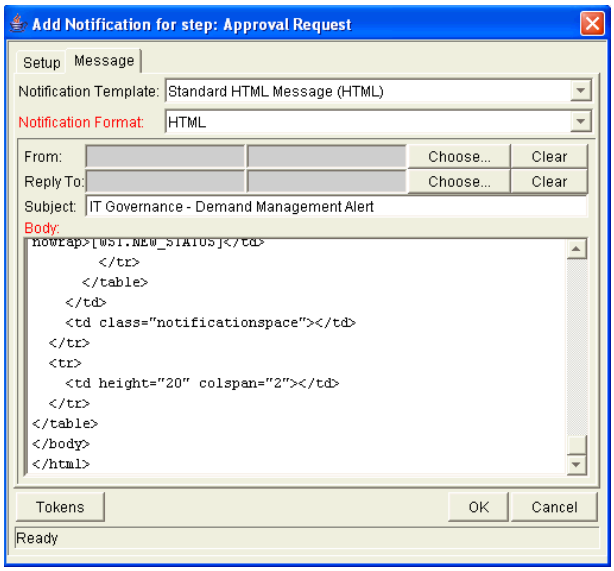
Notifications include the following features to make them easier to configure and use:

- Select from a number of pre-configured 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.



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 74 for a list of these tokens.
   - Token for the URL to the package (Workbench or standard interface). See *Table 3-2* on page 74 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 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 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.

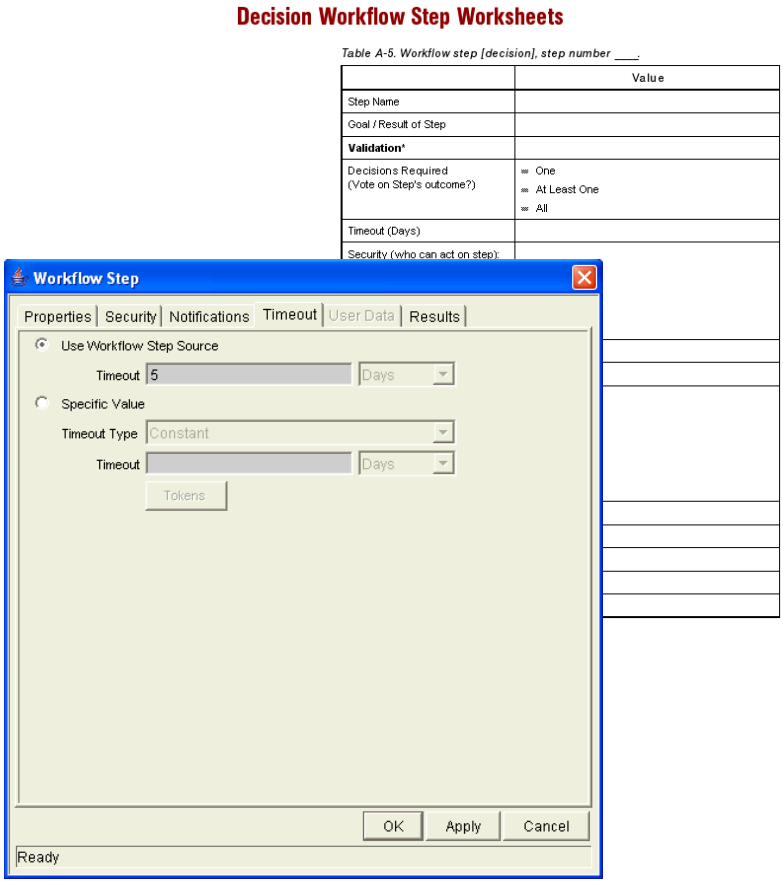
3. Click the **Layout** tab.
4. Right-click a workflow step.

   The workflow step is highlighted. A menu window opens.

5. Click **Edit.**

   The Workflow Step window opens.

6. Click the **Timeout** tab.



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 Event.** The specific event follows this transition. Specific events are based on the workflow step's validation. Used only for Demand Management IT solution.
- **Specific Error.** The specific error follows this transition.
- **Other Errors.** All other errors that do not have transitions set follow this transition.
- **All Errors.** All errors follow this transition.

Adding Transitions Based on Specific Results

To add a Specific Result transition:

1. 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.

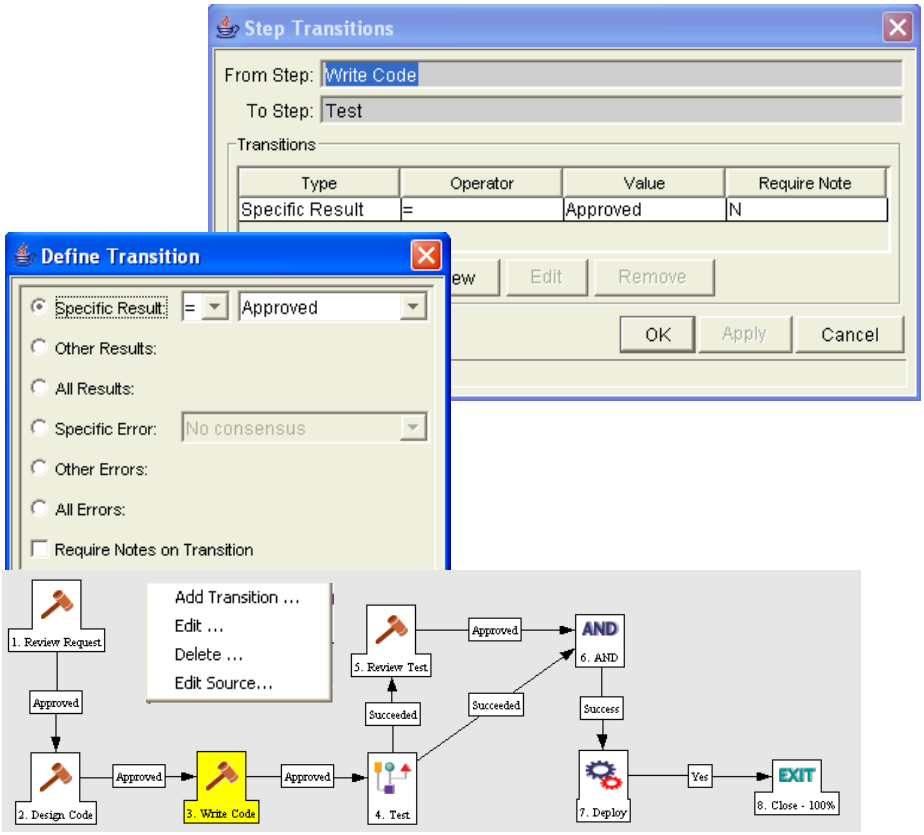
   A line with an arrowhead appears between the workflow steps. The Define Transition and Step Transitions windows opens. The Define Transition window is enabled and has many options on how to define the transition. The most common transition is Specific Results. For information on other transitions definitions, see *Adding Transitions not Based on Specific Results* on page 78.

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.



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 Values in Fields

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

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

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

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

To transition based on the value in a field:

1. 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. In the Workflow Step Source window, copy an existing immediate execution workflow step.

      The Execution window opens.

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

Field Name	Description
Workflow Scope	Requests for request tracking and resolution systems, Packages for deployment systems, Release Distribution for release systems.
Execution Type	Token
Processing Type	Immediate

Field Name	Description
Validation	Selects or creates a validation that includes all of the possible values of the resolved token. For example, if you plan on branching based on the **Priority** field, use the [REQ.PRIORITY_CODE] token and the CRT - Priority - Enabled validation. The validation contains all possible values of the token.
Execution	Enter the token for the value that you would like to transition based on. To find the name of the token, click tokens. The Token Builder opens. Token Builder will help you find the token (for example [REQ.PRIORITY_CODE]), but you must manually enter the name of the token in the **Execution** field.
Enabled	Yes

c. Click **OK.**

5. Add the new immediate execution workflow step to the workflow.
6. Right-click the immediate execution workflow step.

   A menu window opens.
7. Add the transition, as follows.

   a. Select **Add Transition.**

      The menu window closes. The step remains highlighted.

   b. Select the destination workflow step for the transition.

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

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

   d. Click **OK.**

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

      The transition name you specified is added to the transition line.
8. Click **Save.**

Adding Transitions Based on Data in Tables

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

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

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

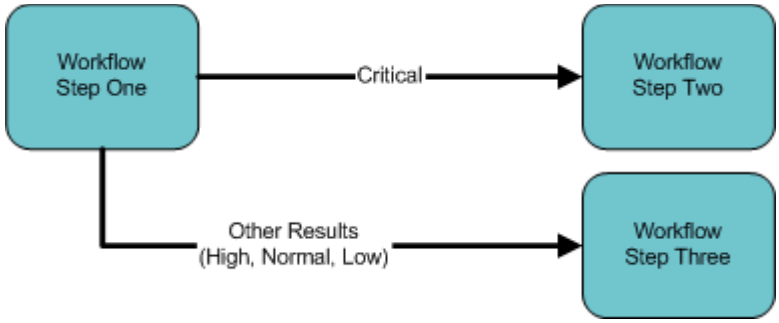
Adding Transitions Based on All But One Specific Value

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

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

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

Figure 3-12. 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 Specific Events

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

Mercury Demand Management supports the following events:

- Assignment
- Schedule Demand
- Reject Demand

A Mercury Demand Management event will not occur if one of the following conditions exist:

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

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

Adding Transitions Based on Errors

You can transition based on a specific error that occurs during an execution step. You can then branch the business process based on likely execution errors such as Timeout, Command Execution, or Invalid Token (see *Table 3-4*). 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 (page 1 of 2)

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.

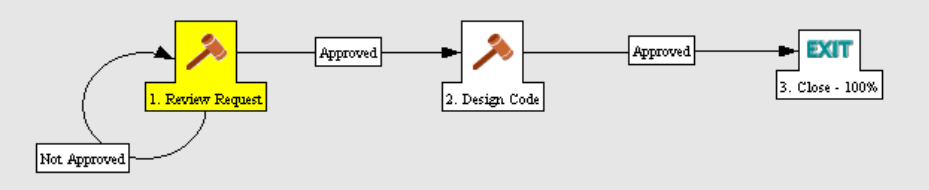
Table 3-4. Workflow transition errors (page 2 of 2)

Transition Option	Meaning
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 3-13*).

Figure 3-13. 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 *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.**

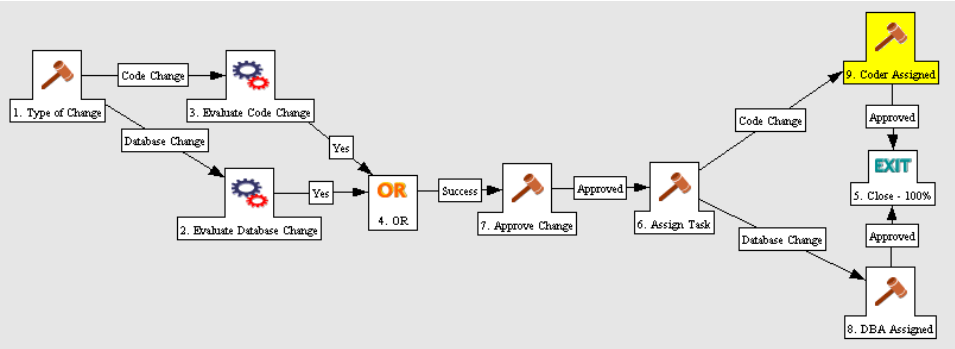
Adding Transitions Based on Previous Workflow Step Results

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 3-14 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 3-14. 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.

Note

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

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

   The Execution window opens.

b. Configure the workflow step.

c. Click **OK.**

6. Add the new execution workflow step to the workflow, as follows:

   a. Add a workflow step to the workflow.

      The Workflow Step window opens.

   b. In the Workflow Step window, on the **Properties** tab, select the workflow parameter from the **Workflow Parameter** field.

   c. Click **OK.**

7. Add the steps and transitions as shown in *Figure 3-14* on page 87.

8. Click **OK.**

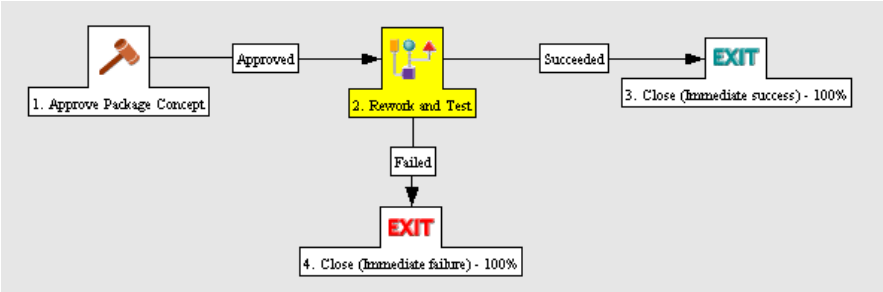
Adding Transitions To and From Subworkflows

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 3-15. Transitioning to and from subworkflows



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

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

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

Configuring Validations for Workflow Steps

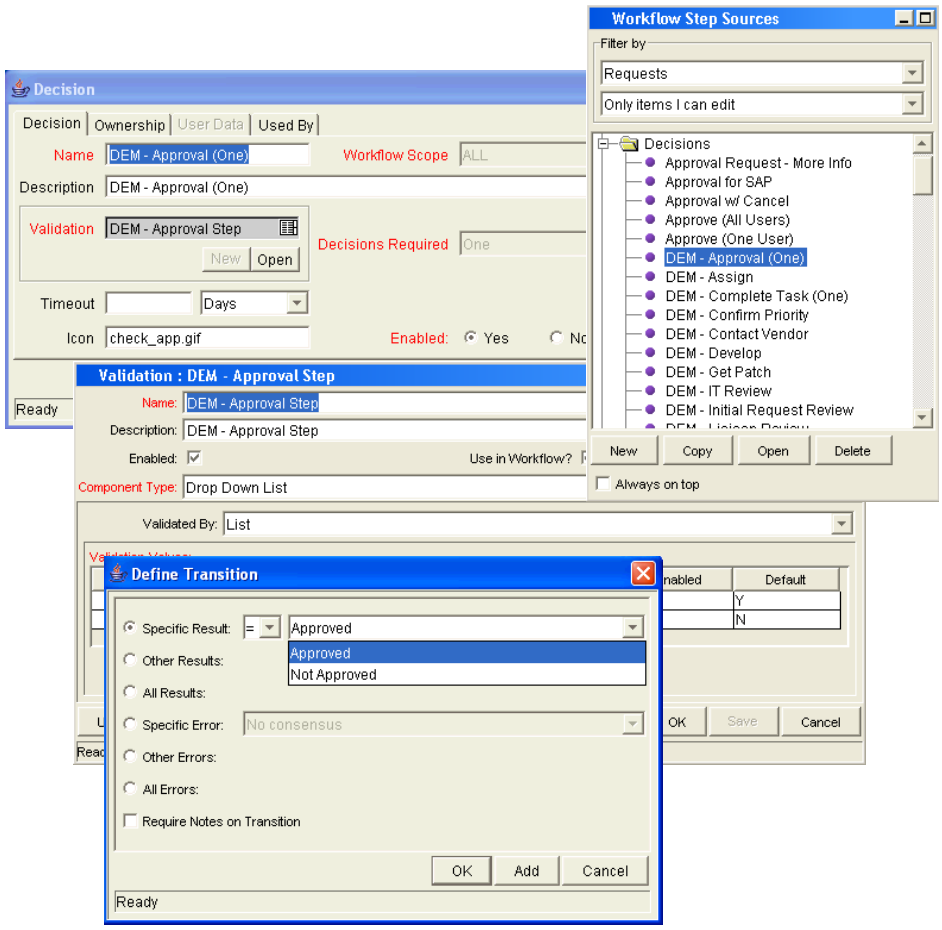
Validations determine the acceptable values for fields. Validations maintain data integrity by ensuring that the correct information is entered in a field before it is saved to the database. For workflow steps, validations ensure the correct transitions are associated with the correct workflow step.

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 3-16* 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 3-16. Workflow step sources and validations



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.

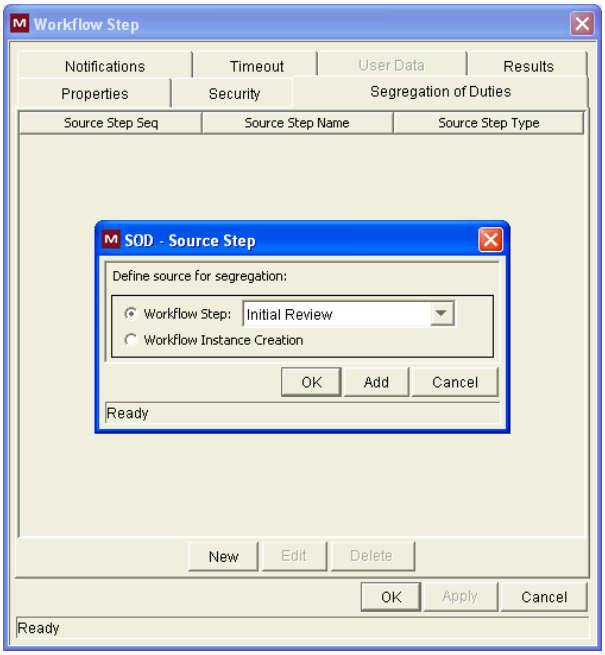
   The Workflow window opens.
3. Right-click a workflow step.

   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.**

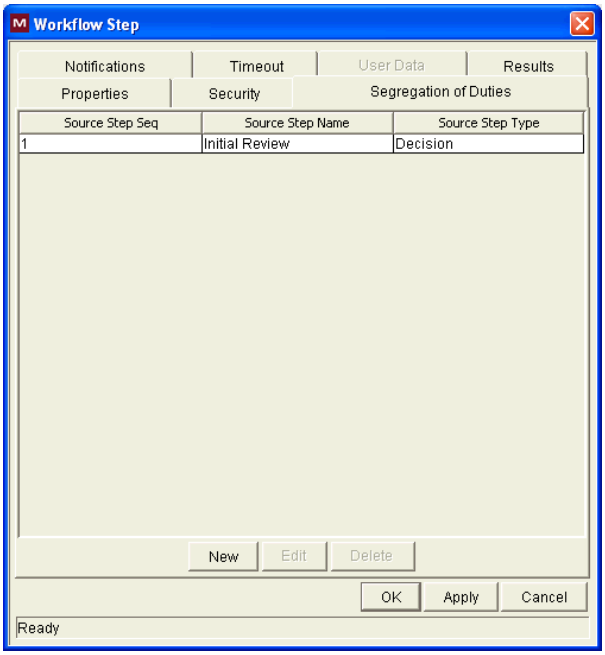
   The SOD - Source Step window opens.



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.



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.

Note

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

Integrating Request Types and Workflows

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

Integrating Request Statuses and Workflows

Request statuses can be linked to their respective workflow steps.

To assign a request status 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 to the **Workflow** tab.

3. Click the **Layout** tab.

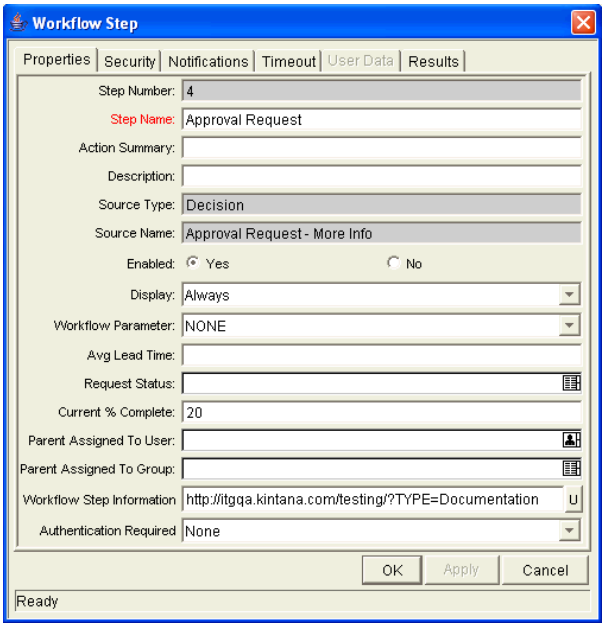
4. Right-click a workflow step.

   A menu window opens.

5. Click **Edit.**

   The Workflow Step window opens.

6. Select the desired request status from the **Request Status** field.



7. Repeat as needed with all necessary workflow steps.
8. From the **Layout** tab, click **OK.**

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

Integrating Request Type Commands and Workflows

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

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

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

- To execute request type commands at a particular workflow step, the workflow step must be configured with the following parameters:
  - Workflow step must be an execution type workflow step
  - Workflow Scope = Requests
  - Execution Type = Built-in Workflow Event
  - Workflow Command = execute_request_commands
- When the request reaches the workflow step (with Workflow Command = execute_request_commands), all commands whose conditions are satisfied will be run in the order they are entered in the request type's command field (in the request type's **Commands** tab).
- The request type can be configured to run only certain commands at a particular step. To do this, specify command conditions.

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.

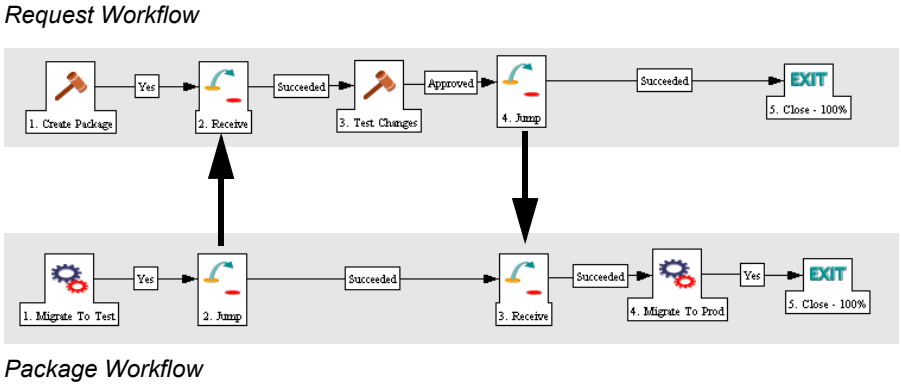
Two built-in workflow events facilitate this cross-product workflow integration. These workflow steps are **wf_jump** and **wf_receive.** Jump workflow step (**wf_jump**) and receive workflow step (**wf_receive**) are used at the points of interaction between workflows. Each jump workflow step must be 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 3-17*.

Figure 3-17. 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 98.

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

   See *Step 2. Generating Jump Step Sources* on page 99.

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

   See *Step 3. Generating Receive Step Sources* on page 101.

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 102.

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

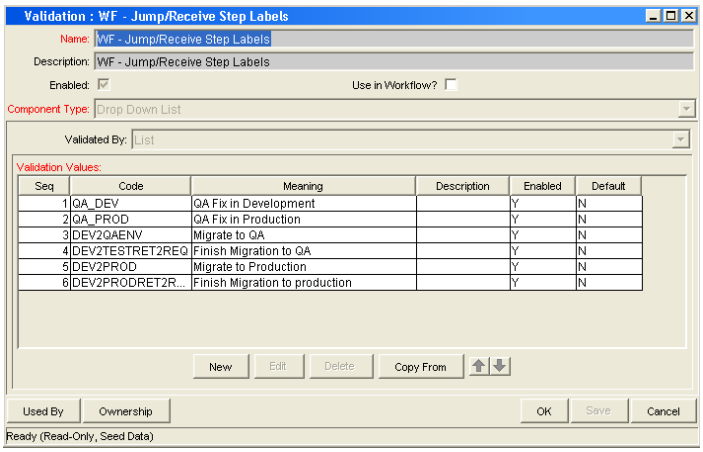
To set up the **WF - Jump/Receive Step Labels** validation:

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

   The Validation Workbench opens.

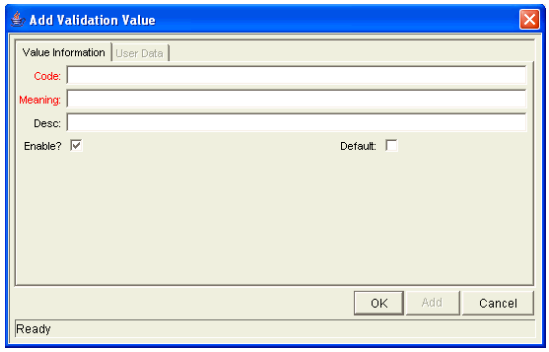
2. In the Validation Workbench, open **WF - Jump/Receive Step Labels.**

   The Validation window opens.



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

   The Add Validation Value window opens.



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 *Commands, Tokens, and Validations Guide and Reference*.

Step 2. Generating Jump Step Sources

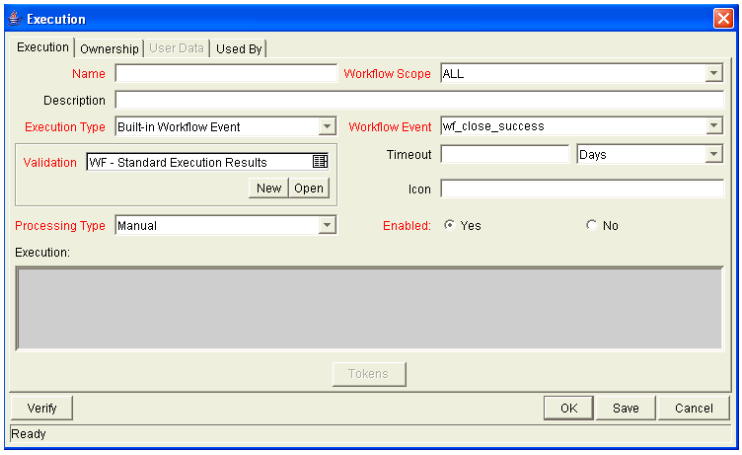
To create a jump step using the wf_jump built-in workflow event:

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.



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.**

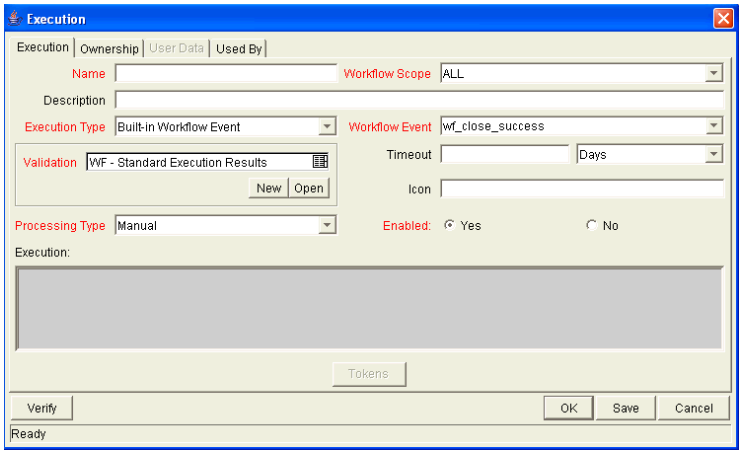
   The Workflow Workbench opens.

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.



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.

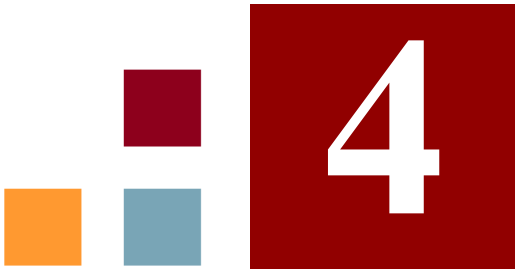
   For example, **Migrate to Production.** This item must be the same for a paired jump and receive workflow step. The Jump/Receive Step Label field is the key communication link between separate workflows. The communicating jump and receive workflow steps must have a matching Jump/Receive Step Label field.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

Chapter



Configuring Workflow Components

In This Chapter:

Overview of Workflow Step Sources

This chapter covers information about Demand Management workflows.

Mercury IT Governance Center includes a number of standard workflow step sources that you can add to a workflow. These sources are pre-configured 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 Mercury IT Governance 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.

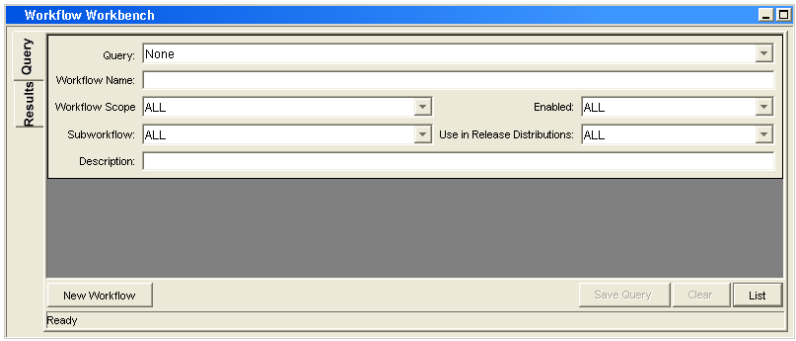
Opening the Workflow Workbench

To open the Workflow Workbench:

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

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

   The Workflow Workbench opens.



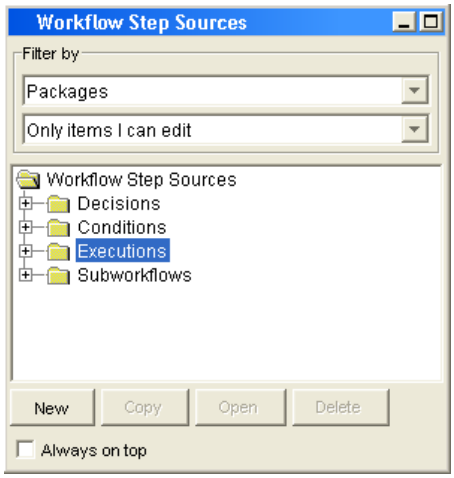
Overview of Creating Workflow Step Sources

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 126). Condition steps cannot be added to, deleted or modified.

To create a new workflow step source:

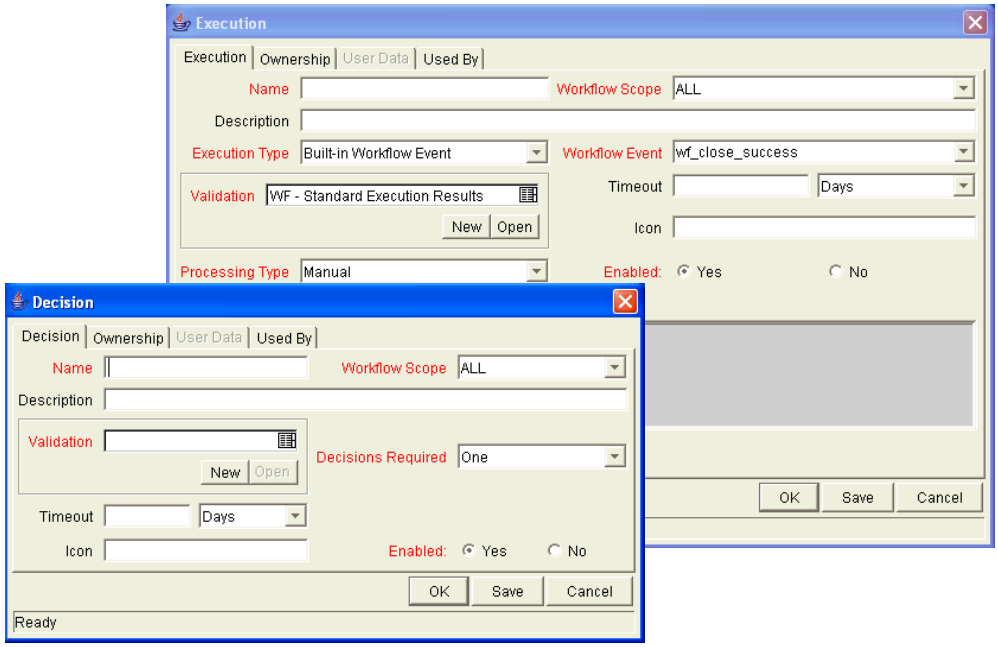
1. From the Workbench shortcut bar, select **Configuration > Workflows.**
2. Open a workflow.

   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.



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

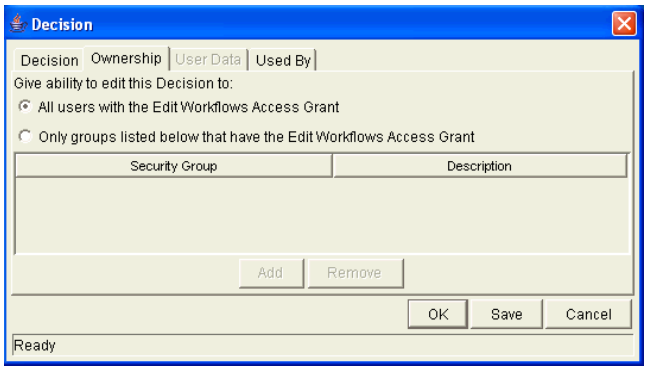
As you Configure a workflow step source, you can specify who can edit the workflow step source.

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.



Note

You use the **Ownership** tab to select the security groups that can edit this workflow step. The default is to allow all security groups who can edit workflows to edit a workflow step source.

5. Select **Only groups listed below that have the Edit Workflows Access Grant.**
6. Click **Add.**

   The Add Security Group window opens.
7. Select a security group.
8. Click **OK.**

   Only users who belong to a listed security group that can edit workflows can now edit this workflow step source.

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

   The new workflow step source is now listed in the Workflow Step Sources window. You can use it in any new or existing workflow with the corresponding workflow scope.

Creating Decision Workflow Step Sources

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)	

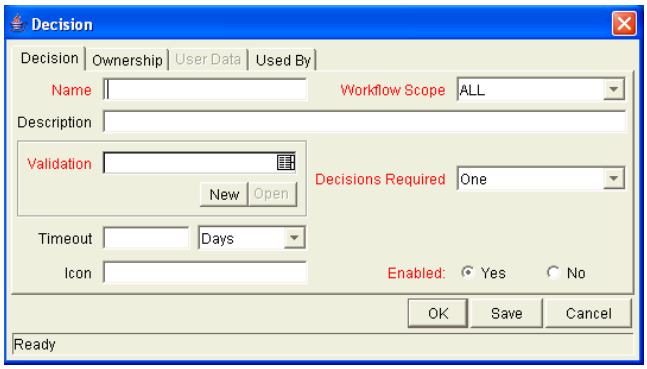
To create a new decision workflow step source:

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.



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 Mercury Demand Management™ request workflows. • **Packages.** For Mercury Deployment Management package workflows. • **Release Distributions.** For Mercury 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.

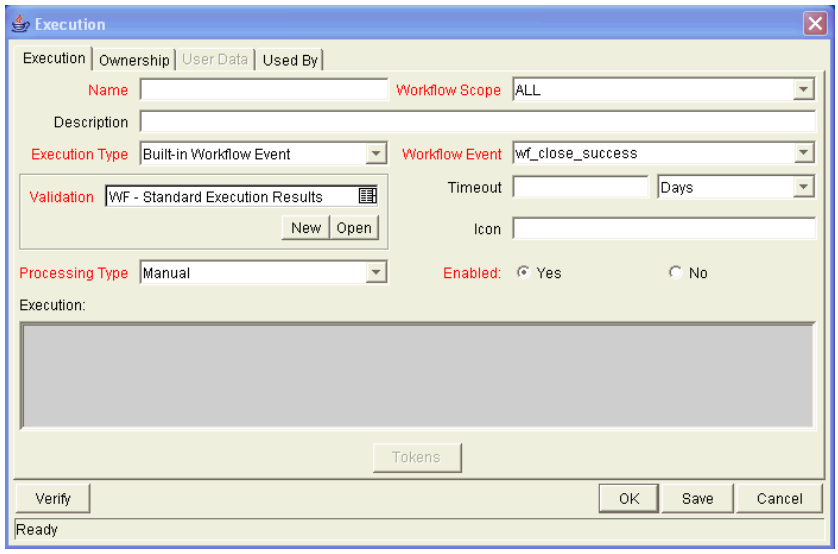
Execution Type**	Value
Built-in Workflow Event: ※ Execute Commands ※ Close ※ Jump / Receive ※ Ready for Release ※ Return from Subworkflow	
PL/SQL Function	
Token	
SQL Statement	
Workflow step commands	

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.



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 Mercury Demand Management request workflows. ● **Packages.** For Mercury Deployment Management package workflows. ● **Release Distributions.** For Mercury 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 a Mercury Deployment Management package. • **rm_ready_for_release.** Generates a Mercury Demand Management request. • **create_package_and_wait.** Generates a Mercury 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.** (Mercury Deployment Management and Mercury Demand Management) Instructs the workflow to proceed to a corresponding Receive Workflow Step in another workflow. • **wf_receive.** (Mercury Deployment Management and Mercury 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.** (Mercury Deployment Management and Mercury 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.
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.

Field Name	Description
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.
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 Executions Types

Execution workflow steps are used to perform specific actions. Mercury Demand 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 Request Type Commands

Certain process steps can require specific commands to be executed. Commands can be added to each request type and the workflow can be configured to execute request type commands at a specific step in the process. Each step runs its own commands to ensure the correct execution for that request type.

Mercury Demand Management includes the execution workflow step source Execute Request Commands 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 Execute Request Commands and changes the field values as defined in *Table 4-1*.

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

Field Name	Description
Name	Enter a descriptive name for the step source.
Workflow Scope	Requests
Execution Type	Built-in Workflow Event
Workflow Event	execute_request_commands
Processing Type	Manual or Immediate
Validation	WF - Standard Execution Results This is the default selection. You can select another existing or create a new validation.
Enabled	Yes
Processing Type	Manual
Page Response	This determines whether the step will complete the execution before reloading the request page for the user (enabling them to make further changes), or whether the request page will reload immediately while the execution is still in progress.

Closing Requests as Success

It is possible to create an execution step that closes a request and marks the request as Success. Each request workflow should resolve with a closed request. All the requests that were closed successfully can then be reported on.

Mercury Demand 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*.

Table 4-2. Execution window values to close requests as success

Field Name	Description
Name	Enter a descriptive name for the step source.
Workflow Scope	Requests
Execution Type	Built-in Workflow Event
Workflow Event	wf_close_success
Processing Type	Manual or Immediate
Validation	WF - Standard Execution Results This is the default selection. You can select another validation or create a new one.
Enabled	Yes

Closing Requests as Failed

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

Mercury Demand 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*.

Table 4-3. Execution window values to close requests as failed (page 1 of 2)

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

Table 4-3. Execution window values to close requests 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

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 4-4*.

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

Field Name	Description
Name	Enter a descriptive name for the step source.
Workflow Scope	Requests
Execution Type	PL/SQL Function
Processing Type	Manual or Immediate
Validation	Selects or creates a validation that includes all of the possible values of the SQL query. You can also create a validation validated by SQL. Use the same SQL from the execution minus the WHERE clause.
Execution	Enter the PL/SQL function.
Enabled	Yes

Executing SQL Statements and Creating Transitions Based on the Results

SQL statement execution workflow steps are used when a workflow 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 as defined in *Table 4-5*.

Table 4-5. Execution window values for executing SQL statements

Field Name	Description
Name	Enter a descriptive name for the step source.
Workflow Scope	Requests
Execution Type	SQL Statement
Processing Type	Manual or Immediate
Validation	Selects or creates a validation that includes all of the possible values of the SQL query. Tip: you can create a validation validated by SQL. Use the same SQL defined for the execution minus the WHERE clause.
Execution	Enter the SQL query.
Enabled	Yes

Evaluating Tokens and Creating Transitions Based on the Results

Mercury 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.

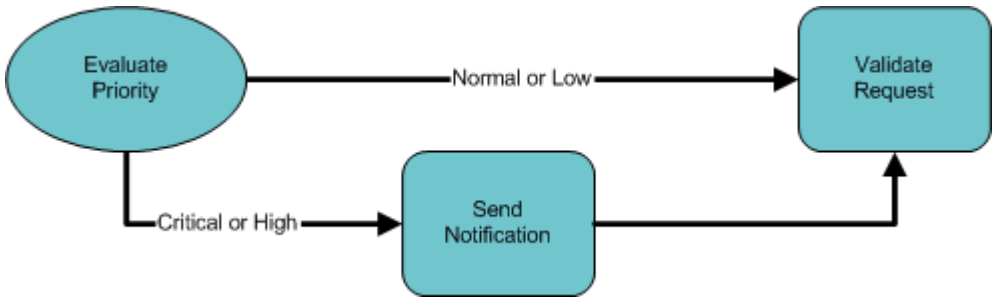
Create a new execution step source with the field values as defined in *Table 4-6*.

Table 4-6. Execution window values for evaluating tokens

Field Name	Description
Name	Enter a descriptive name for the workflow step source.
Workflow Scope	Requests
Execution Type	Token
Processing Type	Manual or Immediate
Validation	Selects or creates a validation that includes all of the possible values of the resolved token. For example, if the token is for the **Priority** field, use the validation for the **Priority** field here as well.
Execution	Enter the token for the value on which the transition is to be based.
Enabled	Yes

For example, IT needs to send an email notification to the Validate and Approve Requests group if the request's priority is High or Critical.

Figure 4-3. Transitioning based on a token



IT decides to use an execution workflow step to automatically evaluate the priority of the request and route it accordingly. If the request's priority is High or Critical, it gets sent to an immediate execution workflow step that sends a notification to the Validate and Approve Requests group before continuing along the workflow. To accomplish this, an execution workflow step source, Evaluate Priority, has been configured with the parameters listed in *Table 4-7*.

Table 4-7. Example of execution window values for evaluating tokens

Field Name	Description
Name	Evaluate Priority
Workflow Scope	Requests
Execution Type	Token
Processing Type	Immediate
Validation	CRT - Priority - Enabled
Execution	[REQ.PRIORITY_CODE]
Enabled	Yes

Executing Multiple System Level Commands

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

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

When either the workflow or the request type commands execute at this step, the commands will 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 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 Demand Management Workflows

You can set up an execution workflow step so that it automatically returns from a subworkflow to its parent Demand Management workflow.

For a request to transition back to the parent workflow, the subworkflow must contain a return step. Transitions leading into the return step must match the validation established for the subworkflow step. You must verify that the validation defined for the subworkflow step is synchronized with the transitions entering the return step.

Mercury Demand Management includes the execution workflow step source Return from Subworkflow that performs this task. Use this step source unless it does not meet the required specifications, such as validation or processing type.

To create the execution step source, make a copy of execution workflow step source Return from Subworkflow and changes the field values as defined in *Table 4-8*.

Table 4-8. Execution window values for subworkflows (page 1 of 2)

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

Table 4-8. Execution window values for subworkflows (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

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:

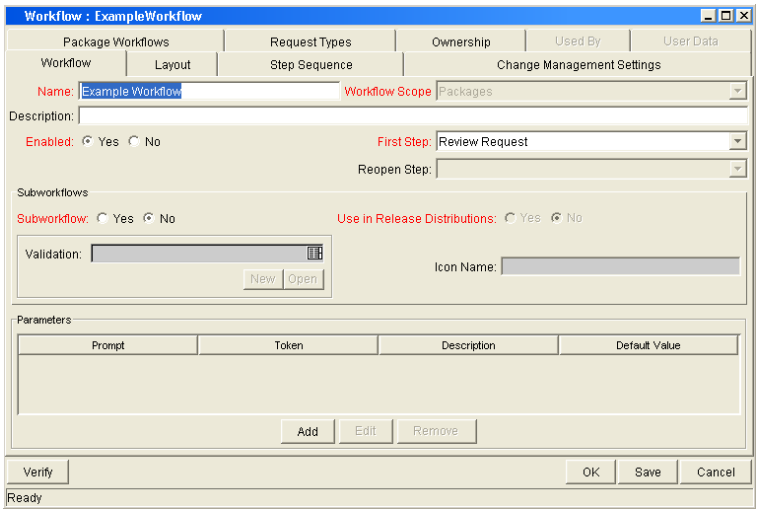
- You can use the WFI.P token prefix to reference workflow parameters.
- You can use workflow parameters in PL/SQL and SQL workflow step executions.

Creating Workflow Parameters

To create a workflow parameter:

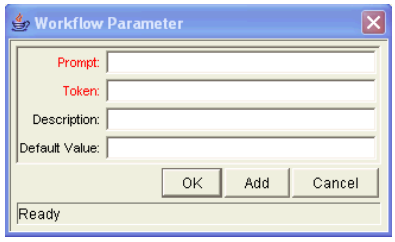
1. From the Workbench shortcut bar, select **Configuration > Workflows.**
2. Open a workflow.

   The Workflow window opens.



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

   The Workflow Parameter window opens.



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

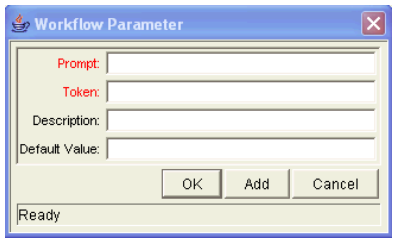
A workflow parameter can be used to generate a counter for the number of times a workflow step enters a state.

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.**

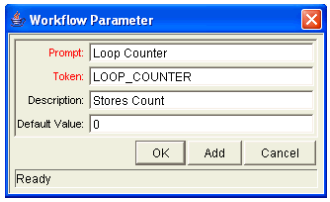
   The Workflow Parameter window opens.



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.**



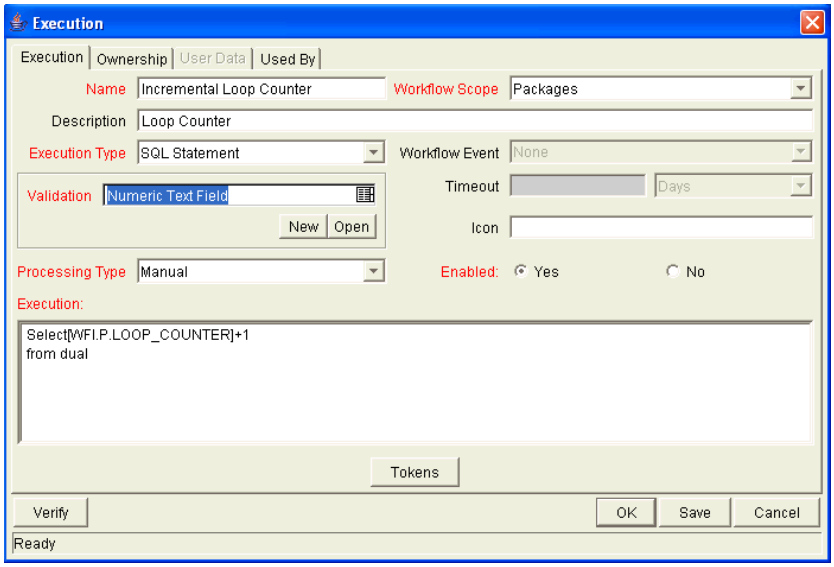
7. From the **Workflow** tab, click **OK.**
8. Create a new immediate SQL execution workflow step.

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.

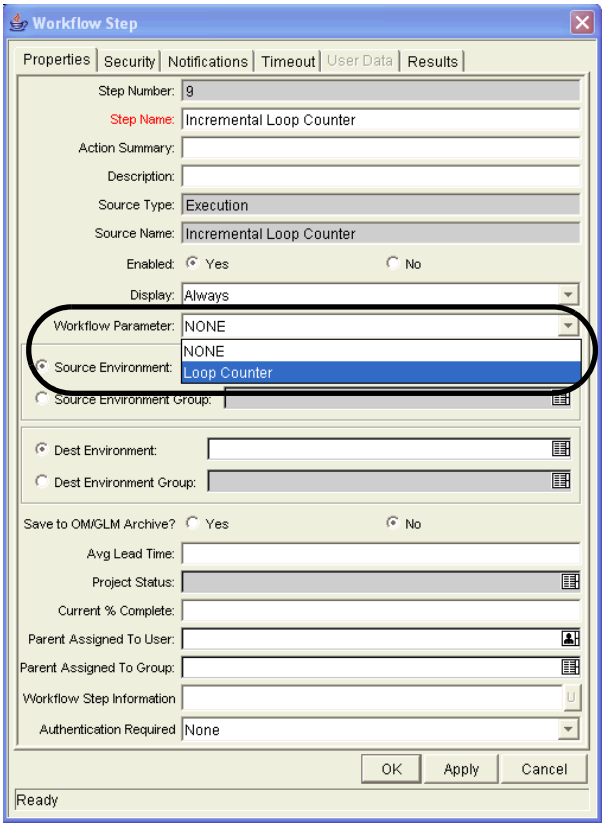
- The result of the SQL execution workflow step returns the result LOOP_COUNTER + 1. This return value is linked back into the parameter when the workflow step is generated on a workflow.
- A validation for a Numeric text field is used. This allows you to use <=, <, >=, and > comparisons in transitions off this step.

The following illustrates the Execution window for the SQL execution workflow step.



9. Add the workflow step to a workflow and choose the new workflow parameter Loop Counter.

   By choosing Loop Count, the workflow engine is told to assign the result of “select loop counter val + 1” from dual back into the loop counter parameter.



You can now add transitions to and from the new loop counter step. For example, you add the loop counter each time an execution fails. If the execution fails three times, a notification is sent to the user. If the execution fails five times, management is notified.

Modifying Workflows Already In Use

Workflows can be modified while they are going through their workflow steps after a package or request has been initiated. These modifications include adding new workflow steps, as well as changing the transitions, security assignments and notifications from within the workflow.

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 4-9*.

Table 4-9. Rules for modifying production workflows

Entity	Procedure
Transitions Security Notifications Workflow Steps Workflow Parameters	You can change any of these entities or add them to a workflow that is in use.
Transitions Security Notifications Workflow Parameters	You can delete any of these entities from a workflow in use.
Workflow Steps	You cannot delete this entity from a workflow in use, but you can rename it. You can delete transitions coming into or going out of a workflow step to effectively remove it from the workflow.

If a workflow that is in use is changed and saved, the changes take effect immediately. Any changes made to workflow steps are applied to all open package lines, requests, releases, and distributions.

Changes to a workflow can have undesirable effects on requests or packages 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 document *System Administration Guide and Reference*. For help with this procedure, contact your application administrator.


Note

This also applies when migrating a workflow with these types of changes into an instance of the Mercury IT Governance Center.

Migrating a workflow with these types of changes into an instance of the Mercury IT Governance 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 document *System Administration Guide and Reference*. For help with this procedure, contact your application 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 4-9*, 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

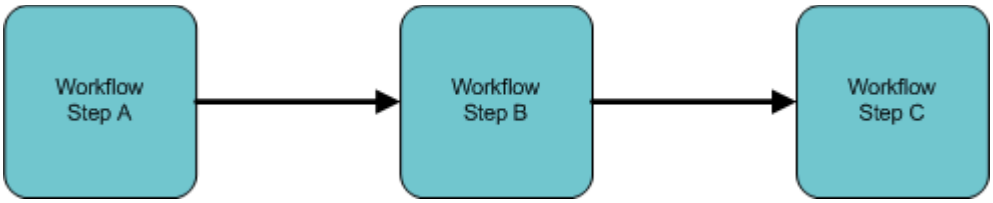
When disabling a workflow step that is currently **Eligible** for user action, the requests or package lines in that step will become stuck. Since the step is now disabled, the user cannot take action on it and will not be able to progress any further through the workflow.

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.

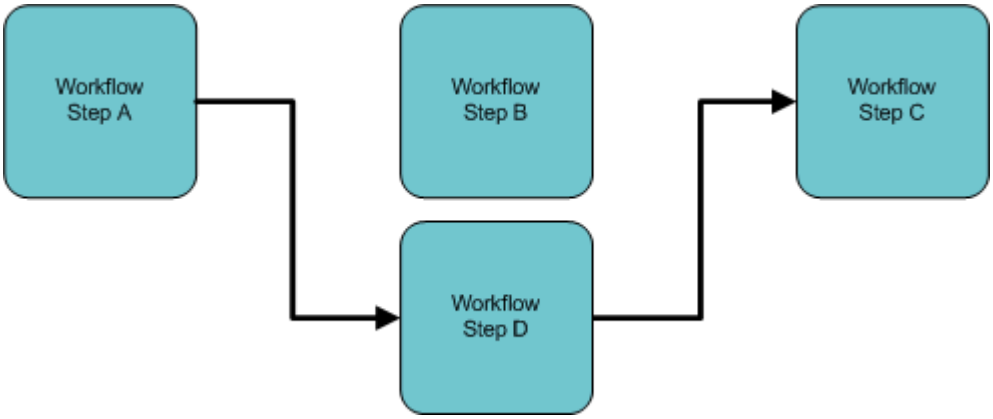
For example, consider a workflow where you wanted to disable workflow step B in the sequence shown in *Figure 4-4*.

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.

Chapter

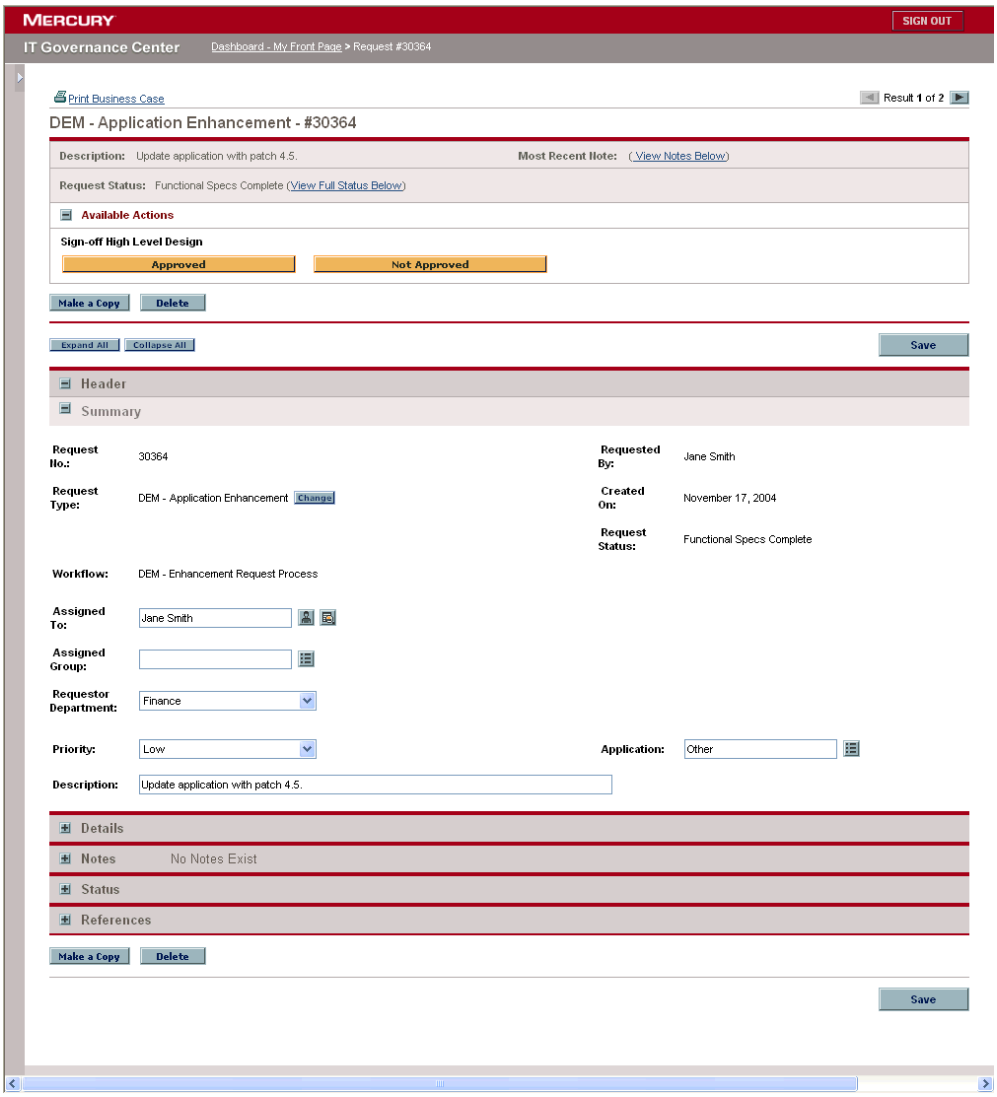
Configuring Request Types and Request Header Types

In This Chapter:

Overview of Request Types

Requests are a fundamental work unit of Mercury IT Governance Center. Users create requests and then submit requests along a resolution process, which is defined in the workflow. The request page contains all information typically required to complete a specific business process (see *Figure 5-1*).

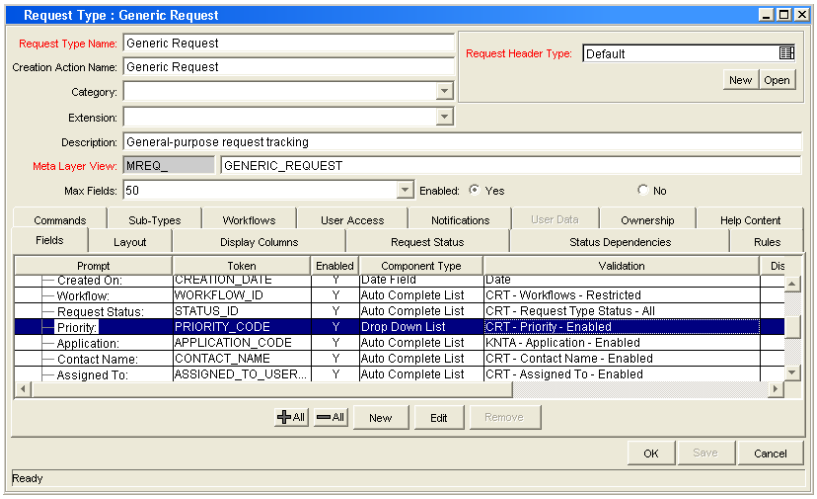
Figure 5-1. Generic request



Each request has an associated request type. Request types determine which fields are included in the request and much of the request-specific logic. Mercury IT Governance Center includes predefined request types, including the Bug request type and the Enhancement request type. Request

types are created and configured in the Request Type window (see *Figure 5-2*).

Figure 5-2. Request Type window



The main components of a request type are as follows:

- **General information.** General information includes basic request type data such as the name and request type category. See *Configuring General Information for Request Types* on page 145.
  - **Request Header Type.** The request header type as a predefined set of basic fields, such as **Priority, Submitted By,** and **Assigned To.**
- **Fields.** Every request type includes a request header type. Each request header type adds a predefined set of fields to the request type. You use the **Fields** tab to create additional fields for the request type. See *Configuring Fields for Request Types* on page 146.
- **Layout.** After you create all of the fields for a request type, you can use the **Layout** tab to configure their layout. For information about how to do this, see *Configuring Layouts for Request Types* on page 157.
- **Display Columns.** Use the **Display Columns** tab to configure the request type columns that can be displayed in a portlet. See *Configuring Displayed Columns for Request Types* on page 163.
- **Request Status.** During its processing, a request can acquire different status values as it progresses along its workflow. These status values can be used to drive field behavior, linking workflow processes to specific information in the request. For information about how to work with request status, see *Configuring Request Statuses for Request Types* on page 165.

- **Status Dependencies.** During its processing, a request can acquire different statuses as it progresses along its workflow. These status values can be used to drive field behavior. For example, a read-only field can become required following changes that effect request status. For more information, see *Configuring Status Dependencies* on page 169.

- **Rules.** You can use request rules to set up the automatic population of request type fields based on dependencies. For more information, see *Configuring Rules for Request Types* on page 173.

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

  For example, you can construct a command to evaluate several parameters, and then set a default value for the field based on those parameters. This provides an advantage over the defaulting features on the **Field** tab, which can only default based on a single parameter stored in the same request type. For more information, see *Configuring Commands for Request Types* on page 180.

- **Sub-Types.** Creates valid sub-types for the request type. For example, a defect request type might have hardware, software, and documentation sub-types. For more information, see *Configuring Sub-Types for Request Types* on page 185.

- **Workflows.** Specify which workflows can work with a request type. For instructions, see *Configuring Request Types to Work with Workflows* on page 188.

- **User Access.** Use the **User Access** tab to set up the participants for a request type. You can then give the participants access rights to the request type, user license and access grant checks still applies on top of these settings, see *Configuring Participants for Request Types* on page 190.

- **Notifications.** Configure emails to be sent if specific fields in the request type are completed. For details, see *Configuring Notifications for Request Types* on page 193.

- **User Data.** Product entities such as packages, workflows, requests and projects include a set of standard fields that capture information about those entities. While these fields are normally sufficient for day-to-day processing, user data fields let you capture additional information specific to each organization. You can specify user data on the **User Data** tab. If there are no user data fields, the **User Data** tab is disabled.
- **Ownership.** Specify who can edit the request type. For details, see *Configuring Ownerships of Request Types* on page 202.
- **Help Content.** Add help content to fields, sections and request types. For details, see *Configuring Help Contents for Request Types* on page 204.

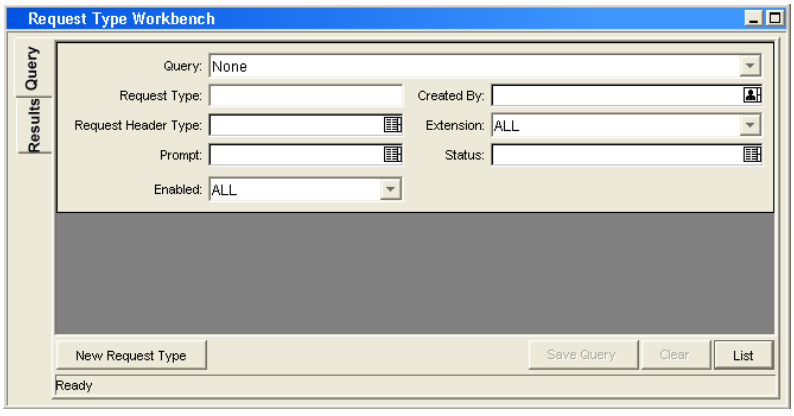
Opening the Request Type Workbench

To open the Request Type Workbench:

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

   The Workbench opens.
3. From the shortcut bar, select **Demand Mgmt > Request Types.**

   The Request Type Workbench opens.



Setting Request Type Defaults

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

To set the default request header type and workflow:

1. From the Workbench shortcut bar, select **Demand Mgmt > Request Types.**

   The Request Type Workbench opens.

2. From the **Query** tab, click **List.**

   The **Results** tab opens and displays all request types.

3. Click **Setup Request Header.**

   The Request Header Setup Dialog window opens.

4. Enter the information specified in the following table:

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

5. Click **OK.**

   The selected workflow and request header type are now defaults.

To change the default number of fields for request types:

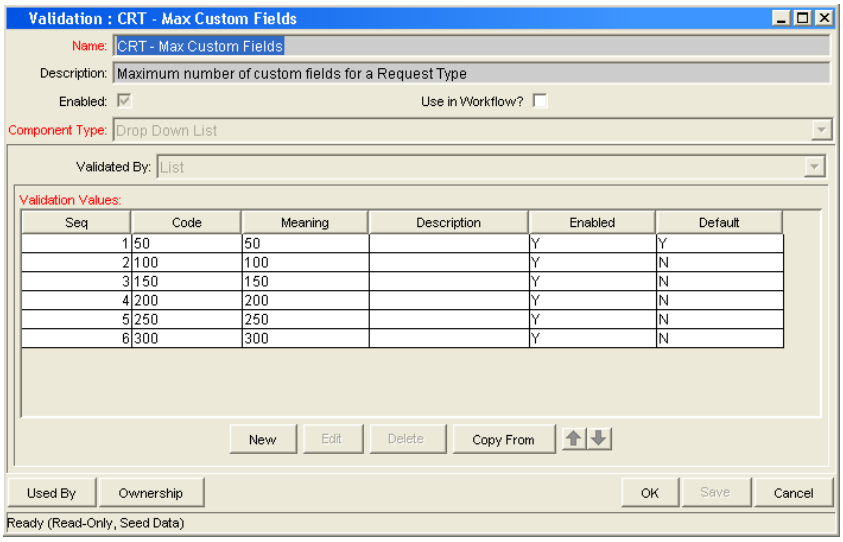
1. From the Workbench shortcut bar, select **Configuration > Validation.**

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

2. Click **List.**

   The **Results** tab opens and displays all validations.

3. Find and then open **CRT- Max Custom Fields.**



4. Click **New.**

   The Add Validation Value window opens.

5. Enter the information specified in the following table:

Field Name	Description
Code	The validation value. Validation values are expressed in increments of 50. The **Code** and **Meaning** fields must display the same value.
Meaning	The validation value. Validation values are expressed in increments of 50. The **Code** and **Meaning** fields must display the same value.
Enable	If set to yes (Y), makes the validation value available to the system.
Default	If set to yes (Y), selects this validation value as the default value.

6. Click **OK.**

   The Validation window lists the new validation.

7. Click **OK.**

Configuring General Information for Request Types

To configure the general information for a request type:

1. From the Workbench shortcut bar, select **Demand Mgmt > Request Types.**

   The Request Type Workbench opens.

2. Open a request type.

   The Request Type window opens.

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

Field Name	Description
Request Type Name	The name of the request type.
Creation Action Name	A description of the request type's function. For example Log a Product Bug. Creation Action Names display on the Create New Request page.
Category	The category containing the request type. Categories are created by an application administrator and are based on the business needs of the organization. Examples of categories which an organization might use are Sales and Support and General Administration. Categories display in the Create New Request window in the standard interface. [Validation = CRT - Request Type Category]
Extension	For release types created for a Mercury Deployment Management extension. Select the extension from the list.
Description	A useful description of how the request type is used.
Meta Layer View	Meta layer views relate information specific Mercury IT Governance 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.
Max Fields	The maximum number of fields the request type can have. See *Setting Request Type Defaults* on page 143.
Enabled	Indicates whether or not the request type is available to Mercury IT Governance Center.
Request Header Type	Selects a request header type to be used with this request type. Select an existing request header type from the auto-complete, or create a new request header type by clicking **New.**

4. Do one of the following:
   - To save the changes and close the Request Type window, click **OK.**
   - To save the changes and leave the window open, click **Save.**

Configuring Fields for Request Types

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

Overview of Request Type Fields

When creating request type fields, there are three general attributes or criteria associated with each request type field. These criteria are:

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

Criteria for Visible Fields

You can specify that a request type field be visible to or hidden from users (see *Table 5-1*).

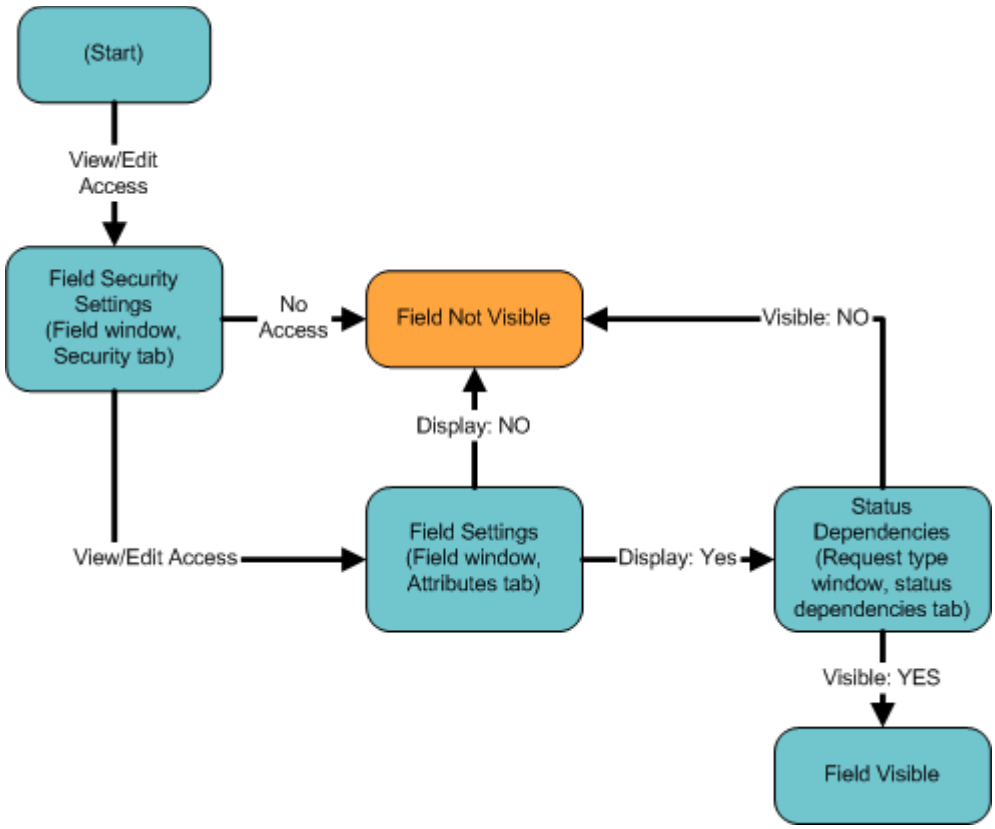
Table 5-1. Criteria for visible fields

Criteria	Description
Field attributes	Use the **Attributes** tab in the Field window to designate a field as always visible or always hidden. For details, see *Creating Fields for Request Types* on page 149.
Request Status	You can specify field visibility based on request status (linked to the workflow step). For details, see *Configuring Request Statuses for Request Types* on page 165.
Field security	You can use the **Security** tab in the Field window to specify field visibility for particular users or security groups. For details, see *Creating Fields for Request Types* on page 149.

Figure 5-3 illustrates how the product determines field visibility for a particular user. This diagram assumes that the user has access to view the

requests, which requires the correct license, access grants, and settings in the Request Type window's **User Access** tab.

Figure 5-3. Field visibility interactions



Criteria for Editable Fields

You can configure request type fields to become read-only (see *Table 5-2*).

Table 5-2. Criteria for editable fields

Criteria	Description
Request Status	You can specify that a field is read-only based on request status. For details, see *Configuring Request Statuses for Request Types* on page 165.
Field security	Use the **Security** tab in the Field window to designate fields as read-only for specific users or security groups. For details, see *Creating Fields for Request Types* on page 149.

Criteria for Default Fields

A field can be configured to automatically update the value in that field (see *Table 5-3*).

Table 5-3. Criteria for default fields

Criteria	Description
Field Defaulting	The value of a single field can be linked to the value of other fields defined for that entity. For example, a request type field can default to a particular manager's username when the value in another field in that request type equals the text Critical. This is controlled from the Field window's **Default** tab. See *Creating Fields for Request Types* on page 149 for additional details.
Request Type Rules	A request type can be configured to automatically populate multiple fields based on the value of one field. For example, if a field has the value bug report, the workflow, contact name, contact phone, and department can be automatically filled. This is controlled from the request type window's **Rules** tab. See *Configuring Rules for Request Types* on page 173 for additional details.
Request Type Commands	Commands can also be used to control certain behavior of request type fields. At specific points (workflow execution steps) in a resolution process, it is possible to select to run the commands stored in the request type. These commands can then manipulate the data inside a request type field. For example, you can construct a command to consider a number of parameters and then default a field based on those parameters. This provides an advantage over the defaulting features in the Field window, which can only default based on a single parameter stored in the same request type. Controlling field values using commands can be useful in the following situations (examples): ■ Store a value from an execution (Note: this can also be done using workflow parameters.) ■ Clearing a field after evaluating a number of parameters. See the *Commands, Tokens, and Validations Guide and Reference* for more information on setting up commands to control field defaulting.

Creating Fields for Request Types

New request type fields are created and configured using the Field window, accessed from the request type window's **Fields** tab.

From the Field window, it is possible to configure:

- Whether the field is displayed
- Whether a field can be edited under different circumstances
- Whether the field defaults to a certain value
- Dependencies to values in other fields in the request type

Because field behavior often depends on other fields in the request type, other request type fields will often have to be created before configuring a field's behavior.

To create a new request type field:

1. From the Workbench shortcut bar, select **Demand Mgmt > Request Types.**

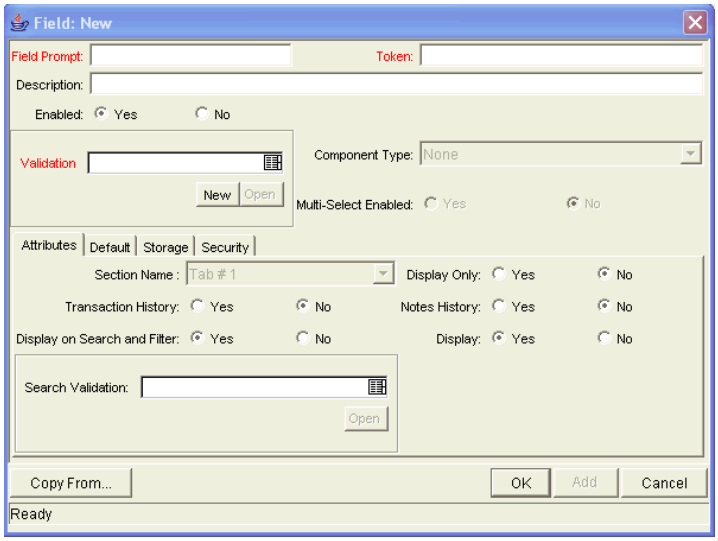
   The Request Type Workbench opens.

2. Open a request type.

   The Request Type window opens.

3. From the **Fields** tab, click **New.**

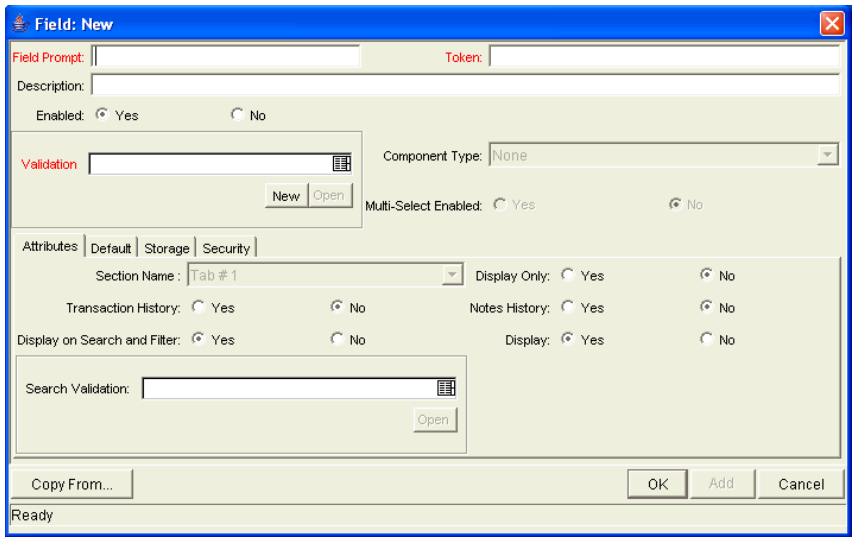
   The Fields window opens to the **Attributes** tab.



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

Field Name	Description
Field Prompt	The prompt visible for the request type field in the request.
Token	An uppercase text string used to identify this field. The token name must be unique for the specific request type. An example of a token name is ASSIGNED_TO_USER_ID.
Description	A description of the request type field.
Enabled	Indicates whether or not the field is turned on for this request type.
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. See *Configuring Fields for Request Types* on page 146 for more details.
Component Type	Defines the visual characteristics of the field (drop-down list, free form text field, and so on). This is derived from the validation chosen. This field cannot be edited.
Multiselect	Indicates whether or not the field allows users to select more than one entry. Only valid for fields with an auto-complete component for the validation.

5. Click the **Attributes** tab.

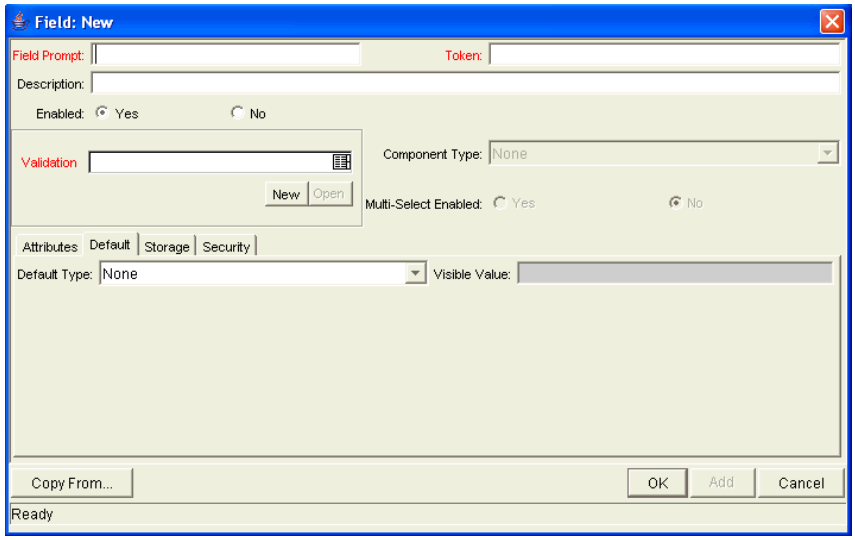


6. Complete the fields in the **Attributes** tab as specified in the following table:

Field Name	Description
Section Name	The section of the request on which the field is displayed.
Display Only	Indicates if the field is only displayed and cannot be updated, even at initial request entry.
Transaction History	Turns transaction auditing on or off for this field. If it is set to **Yes,** whenever this field changes in a request, the change is logged in a transaction history table.
Notes History	Turns notes auditing on or off for this field. If it is set to **Yes,** whenever this field changes in a request, the change is logged in notes for the request.
Display On Search and Filter	Indicates whether or not the field is displayed in Search and Filter pages in the standard interface.
Display	Indicates whether or not the field is seen by requests that use the given request type. If set to **No,** the **Request Type** field is not displayed.

The number of fields in a request type that can have the **Notes History** and **Transaction History** attributes enabled is limited. The total number of fields in a request type that has **Notes History** and **Transaction History** enabled separately or both attributes enabled at the same time cannot exceed forty fields.

7. Click the **Defaults** tab.



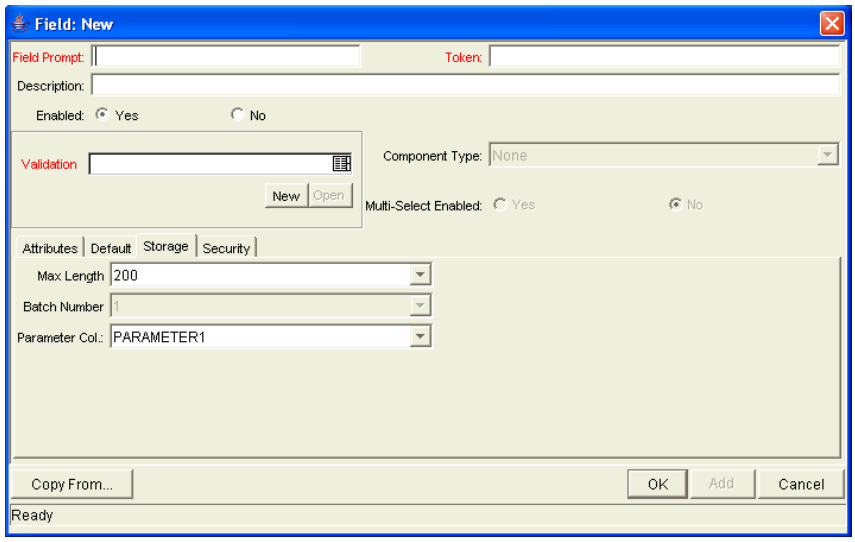
8. Enter the information specified in the following table:

Field Name	Description
Default Type	Defines if the field will have a default value. Either default the field with a constant value, default it from the value in another field, or default to a parameter.
Visible Value	If a default type of **Constant** is selected, you can enter the constant value here.
Depends On	If defaulting from another field, enter the token name of that field. At runtime, when using this request type, 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 **Storage** tab.

   The **Storage** tab automatically places the field into the next available position within the database based on the current field's attributes. By opening the Field window for a specific field found within a request, administrators can use the **Storage** tab to locate a field within the database. This is useful for reporting purposes. If necessary, you can use the **Storage** tab to specify a field location within the database when creating a new field, but the standard method is to allow the interface to automatically position the field for the administrator.

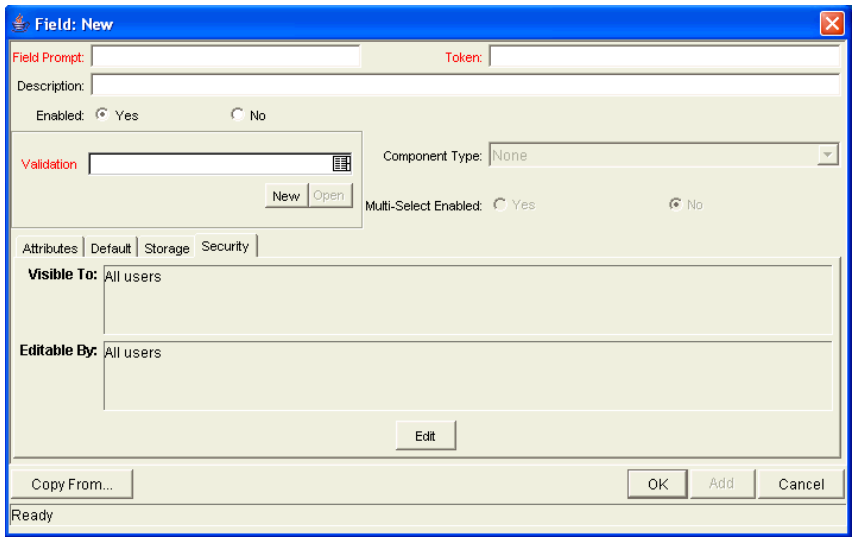
   The **Storage** tab automatically stores the value for a text field of maximum length 4000 in column 41 or higher.



10. Enter the information specified in the following table:

Field Name	Description
Max Length	Indicates the maximum field character length. The possible values are 200 and 4000.
Batch Number	Based on the number of maximum fields. For every 50 fields, one batch is created. 10 of these 50 fields can be more than 200 characters in length. Enabled only when there are more than 50 fields (creating more than one batch).
Parameter Col	Indicates the internal database column that the field value is stored in. These values are then stored in the corresponding column in the request details table for each batch of the given request type. Information can be stored in up to 50 columns using request type, allowing up to 50 fields/batch. No two fields in a request type can use the same column number within the same batch.

11. Click the **Security** tab.



12. Enter the information specified in the following table:

Field Name	Description
Visible To	Lists all users, security groups, and linked tokens to which the field is to be visible.
Editable By	Lists all users, security groups, and linked tokens for which this field is editable.
Edit	Opens the Edit Field Security window, which you use to configure the users, security groups, and linked tokens that can view and/or edit this field. For details, see *Creating Fields for Request Types* on page 149.

a. In the Fields window, on the **Security** tab, click **Edit.**

   The Edit Field Security window opens.

b. Clear the **Visible to all users** option.

c. In the **Select Users/Security Groups** field, select the users who can view this field.

d. After selecting a choice from the list, select the user, security group, or token from the auto-complete.

   To assign the selected user, security group, or token editing rights as well as viewing rights to the field, select the **Provide Editing Rights** option.

e. Click the **Add** arrow button to add the selected user, security group, or token to the **This field is visible to these Users/Security Groups** section.

   To change the Visible and Editable settings for each entry directly in the Edit Field Security window, deselect the box in the Visible or Editable column of the This field is visible to these Users/Security Groups section. To remove viewing rights entirely, select the user, security group, or token, and then click **Remove.**

f. After you finish adding users, security groups, or tokens to the **This field is visible to these Users/Security Groups** section, click **OK.**

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

13. Click **OK.**

    The changes to the request type are saved.

When adding field-level security to existing fields on a request type that has been used to create requests, the Mercury IT Governance database tables are updated to handle this new configuration. Due to the scope of database changes, the Database Statistics need to be rerun on the database. Instructions for this are included in the *System Administration Guide and Reference*. Contact the system administrator for help with this procedure.

Note

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

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

Copying Fields for Request Types

Use the Copy From functionality to streamline the process of adding fields to a request type by copying the definition of existing fields from other request types.

To copy a request type field:

1. From the Workbench shortcut bar, select **Demand Mgmt > Request Types.**

   The Request Type Workbench opens.

2. Open a request type.

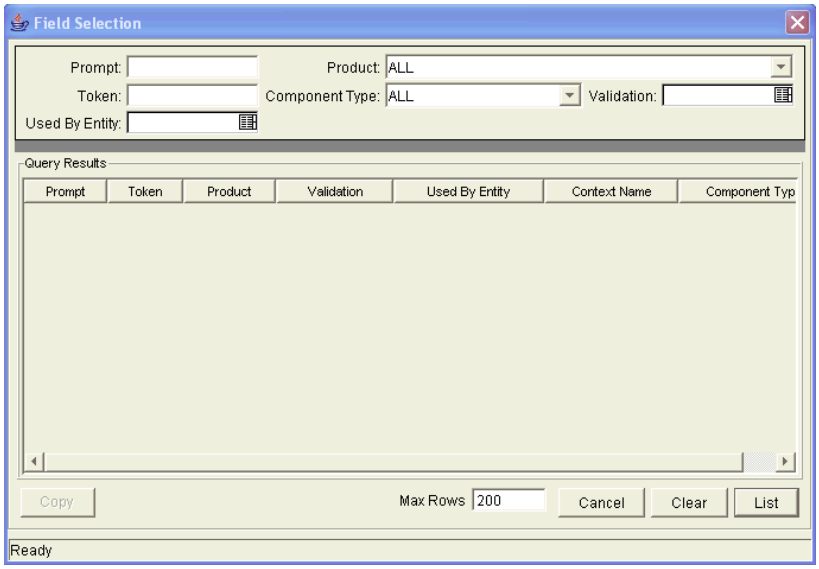
   The Request Type window opens.

3. In the **Field** tab, click **New.**

   The Field window opens.

4. Click **Copy From.**

   The Field Selection window opens.



5. Enter search criteria (such as the token name or field prompt) in the header fields.

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

6. Click **List.**

   The results are listed in the Query Results section.

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

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

Removing Fields for Request Types

To remove a field from a request type:

1. From the Workbench shortcut bar, select **Demand Mgmt > Request Types.**

   The Request Type Workbench opens.

2. Open a request type.

   The Request Type window opens.

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

4. Click **OK.**

Configuring Layouts for Request Types

Request types determine the look and placement of fields on a request.

Modifying Field Widths on Request Types

To change the column width of a field:

1. From the Workbench shortcut bar, select **Demand Mgmt > Request Types.**

   The Request Type Workbench opens.

2. Open a request type.

   The Request Type window opens.

3. Click the **Layout** tab.

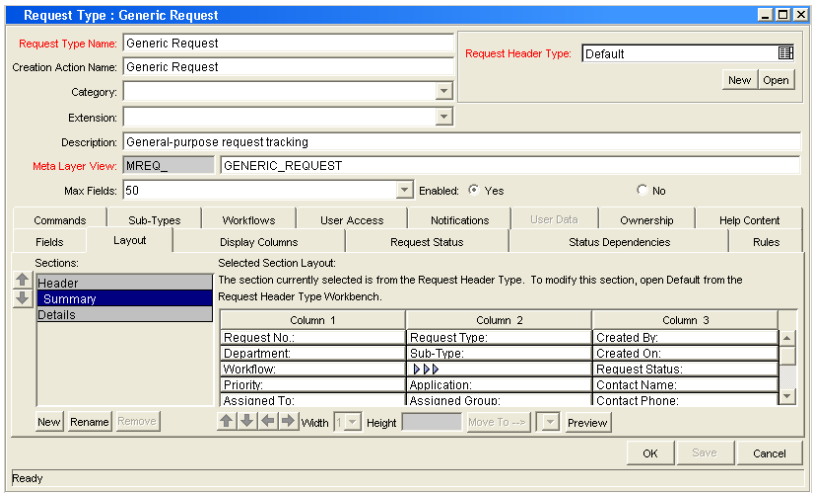
4. In the **Sections** section, select a section of the request type that contains a field.

5. In the **Selection Section Layout** section, select a field.

6. In the **Width** field, select a field width and click **OK.**

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

and change the value in the **Component Lines** field. If the selected field is not of type Text Area, this attribute is blank and non-updateable.



Moving Fields On Request Types

To move a field or a set of fields:

1. From the Workbench shortcut bar, select **Demand Mgmt > Request Types.**

   The Request Type Workbench opens.

2. Open a request type.

   The Request Type window opens.

3. Click the **Layout** tab.

4. In the **Sections** section, select a section of the request type that contains a field.

5. In the **Selection Section Layout** section, select a field.

6. At the bottom of the **Layout** tab, in the layout builder, position the fields. To move a field, select it, and then use the arrow pointers on the **Layout** tab or the corresponding keyboard arrow pointers to change its position.

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

a. Select the field.

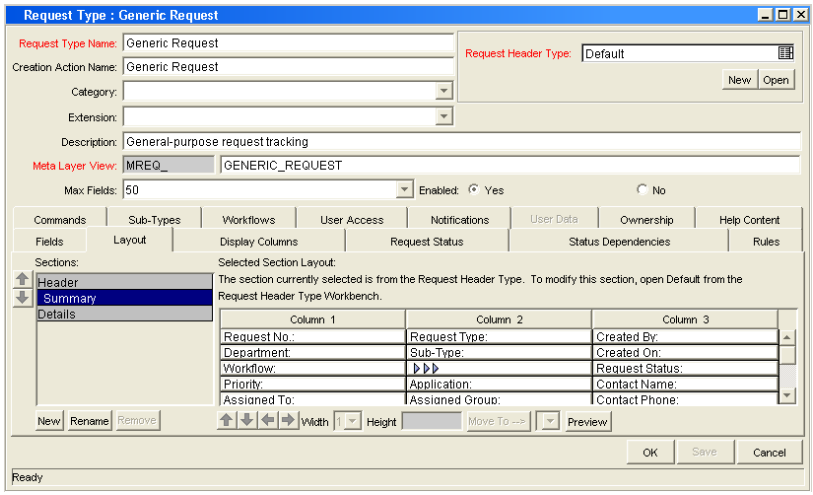
   You cannot choose a request header type field.

b. In the fields next to the **Move To,** select a section from the list.

   You cannot choose a request header type section.

c. Click **Move To.**

   The field is moved to the listed section.



7. Click **OK.**

Adding Sections to Request Types

To add a new section to a request type:

1. From the Workbench shortcut bar, select **Demand Mgmt > Request Types.**

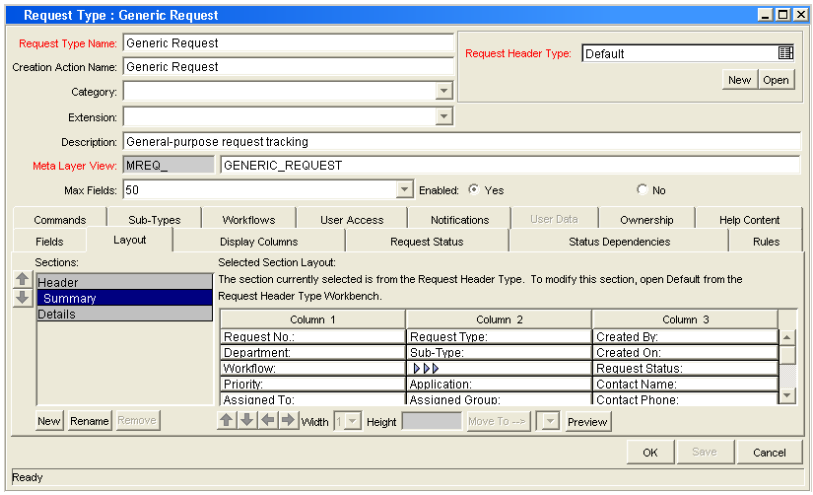
   The Request Type Workbench opens.

2. Open a request type.

   The Request Type window opens.

3. Click the **Layout** tab.

4. Click **New.**



The Input window opens.

5. Enter a new section name.

   Custom section names can contain up to 30 characters.

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

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

   An HTML window opens to show how the fields are to be displayed.

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

   Any hidden fields do not affect the layout. The layout engine considers these as blank fields.

7. Click **OK.**

Changing Section Names on Request Types

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

To change the name of a section:

1. From the Workbench shortcut bar, select **Demand Mgmt > Request Types.**

   The Request Type Workbench opens.

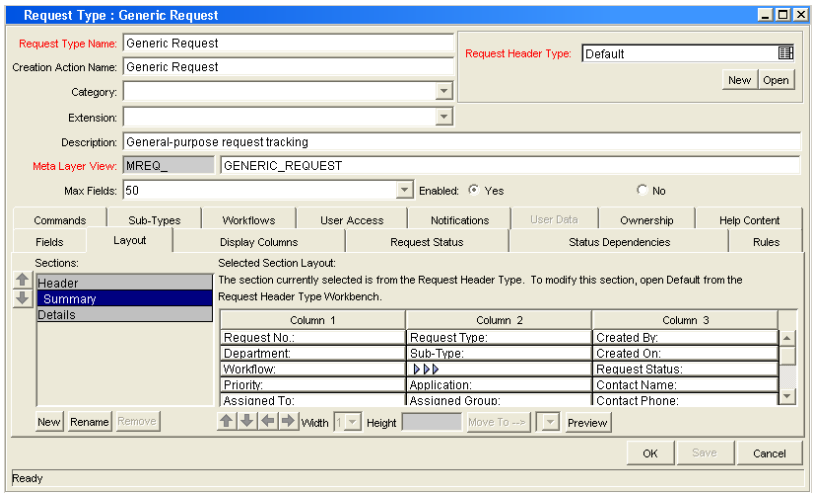
2. Open a request type.

   The Request Type window opens.

3. Click the **Layout** tab.

4. In the **Sections** section, select a section.

5. Click **Rename.**



   The Input window opens.

6. Enter a new section name.

   Custom section names can contain up to 30 characters.

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

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

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

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

   Any hidden fields do not affect the layout. The layout engine treats them as blank fields.

8. Click **OK.**

Deleting Sections on Request Types

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

To delete a section:

1. From the Workbench shortcut bar, select **Demand Mgmt > Request Types.**

   The Request Type Workbench opens.

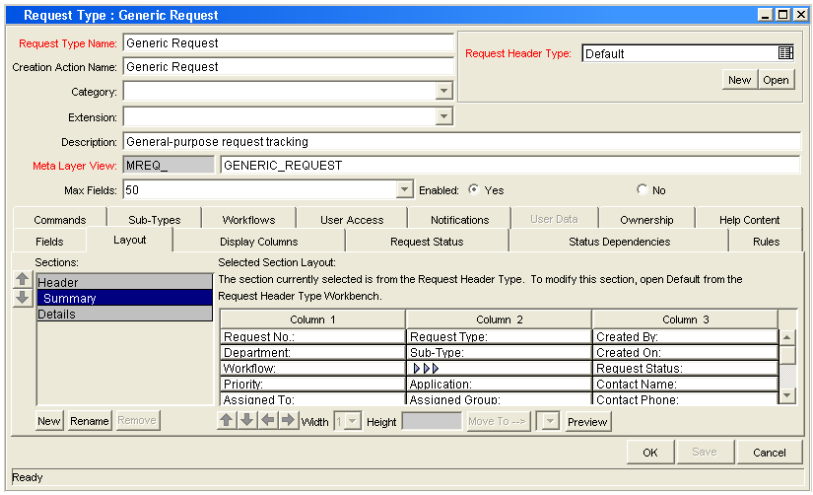
2. Open a request type.

   The Request Type window opens.

3. Click the **Layout** tab.

4. In the **Sections** section, select a section.

5. Click **Remove.**



6. Click **OK.**

Configuring Displayed Columns for Request Types

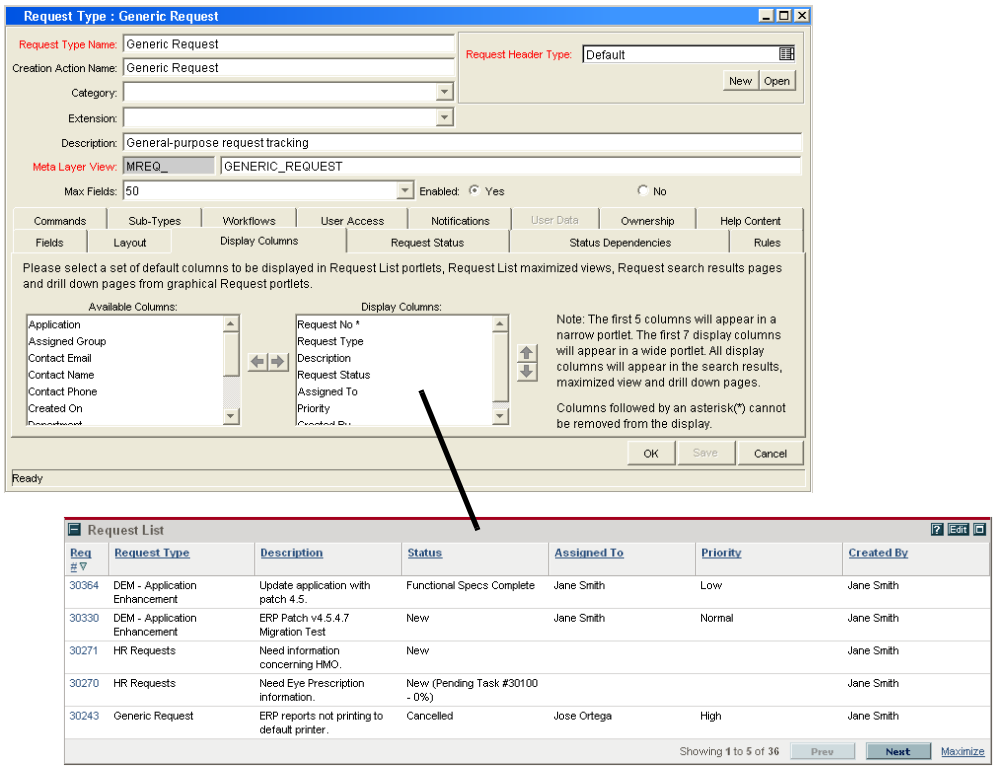
Certain information in a request can provide a useful summary-level description of the request. This can include information such as the request type, a description of the request, and a priority. For each request type, it is possible to control which request columns are displayed to you in the following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

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

You can view the information on these pages to decide if they need to view the details of a specific request.

Figure 5-4 shows how the settings in the Request Type window control the columns that are displayed on a request list portlet page.

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



To configure the columns that are displayed in list portlets:

1. From the Workbench shortcut bar, select **Demand Mgmt > Request Types.**

   The Request Type Workbench opens.

2. Open a request type.

   The Request Type window opens.

3. Click the **Display Columns** tab.

4. In the **Available Columns** field, select the columns to display.

5. Click the right pointer.

   The selected items are moved to the **Display Columns** field.

6. Remove any columns that you do not want to display from the **Display Columns** field.

7. Click **OK.**

In request portlets, this setting represents the default columns that are displayed in the portlet. The user can select to display alternate columns when personalizing the portlet.

Similarly this setting represents the default columns that are displayed when using the advanced search functionality in the Request List portlet or Request Search Results page.

Configuring Request Statuses for Request Types

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

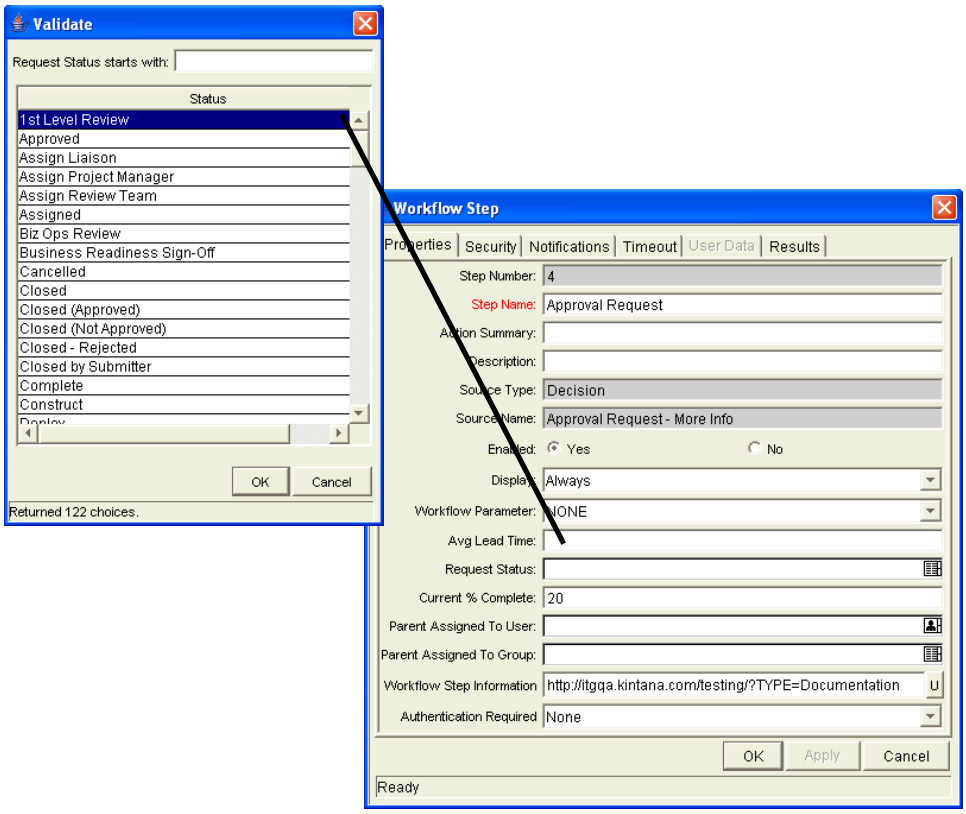
Overview of Request Statuses

Requests can take on different statuses as they progress along their lifecycle. Some possible request statuses include:

- Submitted
- Assigned
- In Progress
- On Hold
- Complete

These statuses are then linked to the workflow steps to drive the request logic. *Figure 5-5* shows how statuses are linked to workflow steps.

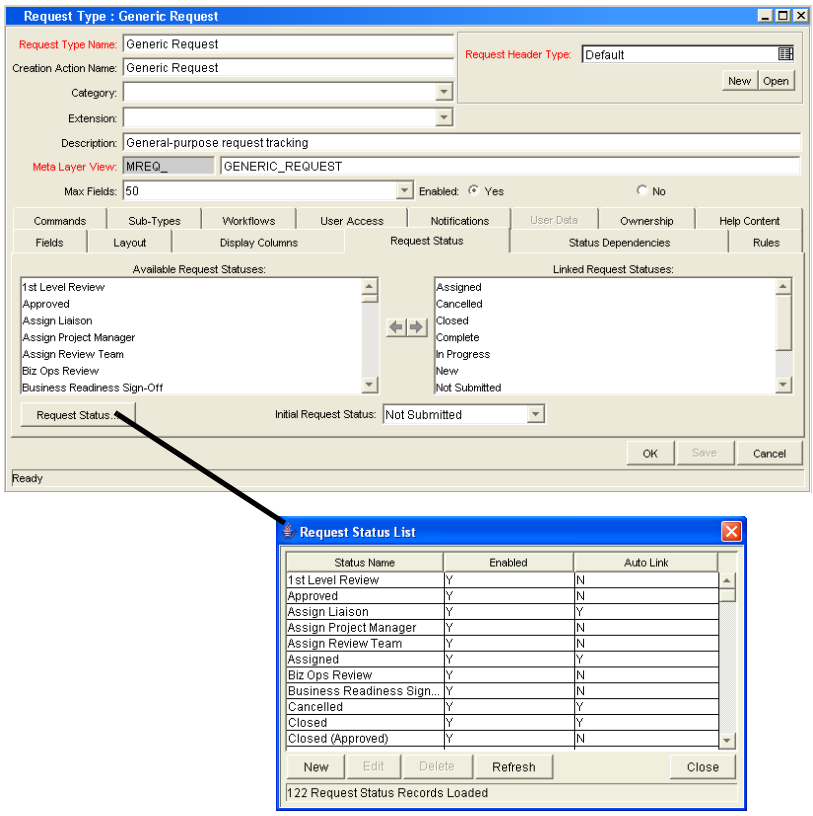
Figure 5-5. Request status specified in the workflow step window



As a request moves along this workflow, its status changes at particular steps. Each status can be linked to request field behavior through the **Status Dependencies** tab. For more information on linking request statuses to field behavior, see *Configuring Status Dependencies* on page 169.

Before linking request statuses to workflow steps, the request type must first possess all required status values. You use the **Request Status** tab (*Figure 5-6*) to set up the list of possible status values.

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



If the **Available Request Statuses** list does not display the value that you want to select, you can create the status value. To set the initial status of a request. use the **Initial Request Status** field.

Creating Request Statuses for Request Types

To create a new request status:

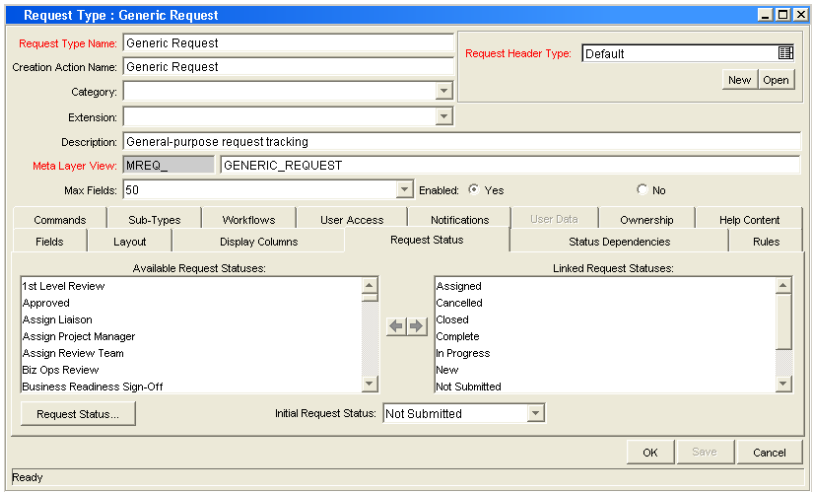
1. From the Workbench shortcut bar, select **Demand Mgmt > Request Types.**

   The Request Type Workbench opens.

2. Open a request type.

   The Request Type window opens.

3. Click the **Request Status** tab.

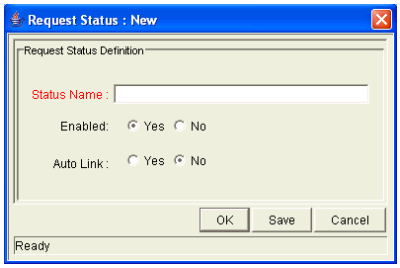


4. Click **Request Status.**

   The Request Status List window opens.

5. Click **New.**

   The Request Status: New window opens.



6. Enter the information specified in the following table.

Field Name	Description
Status Name	The name of the new status.
Enabled	Make the new status available to the system. Select **Yes** for the status to appear in the Available Request Status column for all new request types.
Auto Link	Allows the new status to automatically link to all new request types. Select **Yes** for the status to automatically link.

7. Click **OK.**

   The Request Status List window opens.

8. Click **OK.**

Configuring Status Dependencies

On a request, field behavior can be linked to the status of the field. This is done on the **Status Dependencies** tab in the request type window.

For example, a request cannot reach the Assigned request status unless the **Assigned To User** field has a value. In addition, if a request has a status of Assigned, a user cannot clear the **Assigned To User** field.

To make this work, the field is set to the following parameters for the **Assigned** status:

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

To assign field properties based on request status:

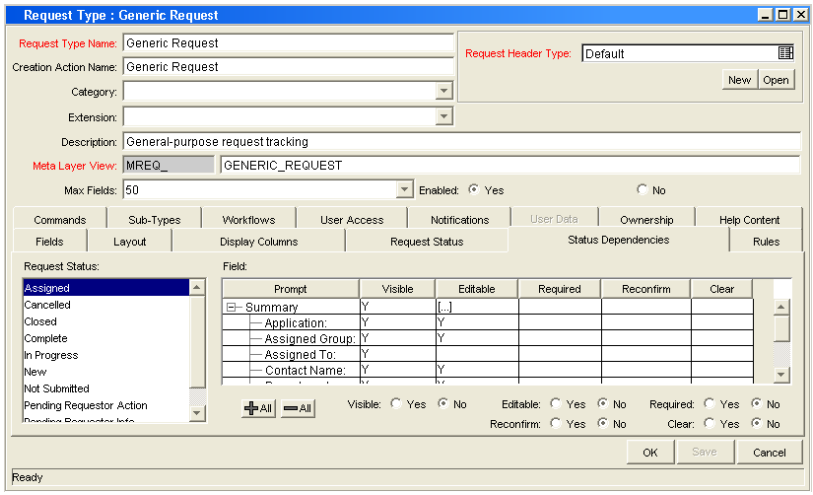
1. From the Workbench shortcut bar, select **Demand Mgmt > Request Types.**

   The Request Type Workbench opens.

2. Open a request type.

   The Request Type window opens.

3. Click the **Status Dependencies** tab.



4. In the **Request Status** field, select a request status.
5. In the **Field** section, select a field.
6. Enter the information specified in the following table.

Field Name	Description
Visible	The **Visible** field determines whether or not a field is visible for a specific request status. If it is set to Visible = **No,** then the field is hidden.
Required	When a field is required, it is necessary to enter a value for the field when changes are made to the request that would affect the request status.
Updateable	If a field is set to Updateable = **No** for a specific request status, then it is not possible to edit the field at the given request status. If a field is set up as **Required, Reconfirm,** or **Clear,** it must be set to Updateable = **Yes.** At certain stages in a request resolution process, it may be desirable to ensure that specific fields do not get updated. For example, when a request of type Vendor Bug is at the status Patch Applied, it may be desirable to make sure that the **Patch Number** field is not updated. This logic is controlled at the request type level. For each request type, it is possible to determine which request fields are updateable and non-updateable when a request is at each possible request status. When a field of a request cannot be updated due to this logic, the field is grayed out in the request. The value is visible but cannot be changed.
Reconfirm	When a field in the request type is set to Reconfirm = **Yes,** it is presented to the user before the request moves to the next step in the workflow. The contents of these fields can then be reviewed and changed.
Clear	The **Clear** field is used in conjunction with other dependencies to remove the contents of a field. The basic uses of the Clear flag are: ■ When **Clear** is set to **Yes** and the **Required** and **Reconfirmed** are set to **No,** the field is not presented to the user or cleared entering this status, but the contents of that field are cleared before moving to the next step in the workflow. ■ Any fields that have the **Clear, Required,** and **Reconfirmed** enabled cause the field to show up in red, but cleared. Appropriate values must then be entered. ■ All of the **Clear** events are logged in the request's **Notes** section as a status change from the old value to ""; if a new value for that field is chosen, then the new value is indicated in the **Notes.**

Multiple fields can be configured simultaneously by using the **Ctrl** or **Shift** keys to select the fields and then change the attribute values. Select a tab row, such as **Header Fields,** to configure all fields in the tab simultaneously. It is also possible to select multiple statuses and change the same fields if those states require the same attribute values for the same fields.

7. Click **OK.**

Status Dependencies Interactions

Table 5-4 illustrates the results of different combinations of the Required, Reconfirm, and Clear functions. For each request status within a request type, there can be up to a maximum of 250 fields with a required state and 250 fields with a reconfirm state.

Table 5-4. Status dependencies interactions

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

Configuring Rules for Request Types

Request rules can be used to set up the automatic population of request fields based on various dependencies. Request rules are ideal for the following scenarios:

- A default workflow, assigned to user or assigned group should be specified when a request of this type is initially created.
- Multiple request fields should be populated depending on the value of a single field.

The two types of request rules are:

- **Simple Default Rules.** Allow a default workflow to be specified, as well as the **Assigned To** and **Assigned Group** fields, depending on the department or application the user specifies. The **Workflow, Assigned To,** and **Assigned Group** fields can also be specified upon request creation.
- **Advanced Default Rules.** Define logic for the automatic population of fields in the request based on user entries.

When configuring request rules, use the **Rule Type** field to switch between **Simple** and **Advanced Defaults.** However, when switching between rule types, whatever work has been done in the first type is lost when the switch is made.

Creating Simple Default Rules for Request Types

Simple default rules are used to automatically fill the **Workflow, Assigned To,** and **Assigned Group** fields. These fields can be filled based on the **Rule Event** and **Dependencies** fields. Using any appropriate combination of these control fields, the **Workflow, Assigned To,** or **Assigned Group** fields can be specified.

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

By setting the workflow and the rule event to **Apply On Creation,** you can set the default workflow to be used each time a request of that type is used.

To add a simple default rule to a request type:

1. From the Workbench shortcut bar, select **Demand Mgmt > Request Types.**

   The Request Type Workbench opens.

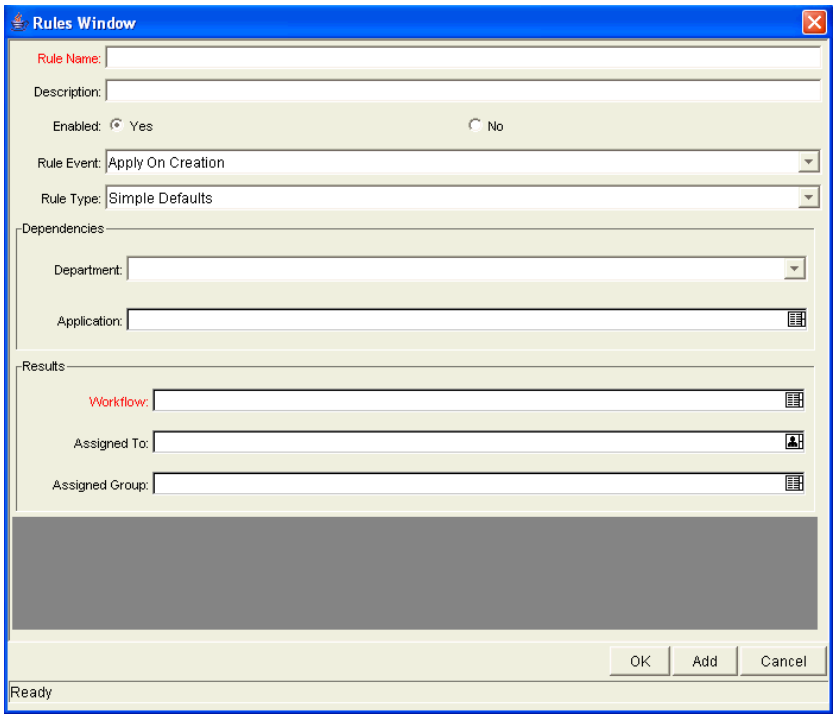
2. Open a request type.

   The Request Type window opens.

3. Click the **Rules** tab.

4. Click **New.**

   The Request Type Rules window opens in Simple Defaults mode.



5. Enter the information specified in the following table:

Field Name	Description
Rule Name	The name of the new rule.
Description	A description of the rule.
Enabled	Selects if the rule is available to the system. Selecting **Yes** means the rule is available to the system.

Field Name	Description
Rule Event	Specifies the event that triggers the rule. ■ **Apply On Creation.** The rule will fire when the request is created, filling in whichever of the **Results** fields have been specified. ■ **Apply On Field Change.** The rule will fire when one of the **Dependencies** fields changes to the specified value. ■ **Apply On Field Change And Stop Processing Rules.** The rule will fire when one of the **Dependencies** fields is changed to the specified value, and all subsequent rules in the **Rules** tab are not.
Rule Type	The type of rule. ■ **Simple Defaults** ■ **Advanced Defaults**
Department	Specifies the department that triggers the rule.
Application	Specifies the application that triggers the rule.
Workflow	The workflow applying to this rule.
Assigned To	The user assigned by this rule.
Assigned Group	The group assigned by this rule.

6. Click **OK.**

   Once this rule is saved, any new request matching the combination of Request Type, Department, and Application for the rule automatically updates the **Workflow, Assigned To,** and **Assigned Group** fields to the default values that the rule specifies.

   If more than one rule applies for a given request, then the system uses a more specific rule. For detailed information, see *Creating Advanced Default Rules for Request Types* on page 176.

Creating Advanced Default Rules for Request Types

Advanced default rules define logic for the automatic population of fields in the request based on user entries. Advanced default rules differ from simple default rules in the following ways:

- Simple default rules can only trigger from request creation or changes to the **Department** or **Application** fields. Advanced default rules can trigger from changes to any field in the request.
- Simple default rules can only populate the **Workflow, Assigned To,** or **Assigned Group** fields. Advanced default rules can populate any field or set of fields in the request simultaneously, including fields in the request or in the request header.


Note

Configuring advanced default rules requires knowledge of SQL.

Advanced default rules are often used with the following values from the **Rule Event** field:

- **Apply On Creation.** The rule will fire when the request is created, and populate the specified **Results** fields.
- **Apply On Field Change.** The rules applies when values in other fields change. This functions two ways:
  - **Specific value.** The rule applies when a field specified in the **Dependencies** section is changed to a specific user-defined value. If multiple dependency fields are defined for a rule, all of them must match in actual use for the rule to take effect.
  - **All values.** The rule applies for any value of a field specified in the **Dependencies** section.

When the field or fields specified in the **Dependencies** section are changed, any fields specified in the **Results** section are automatically populated according to rule order. This is useful in the event of multiple **Dependency** field matches.

- **Apply On Field Change And Stop Processing Other Rules.** The rule applies when a field specified in the **Dependencies** section is changed to a user-defined value. If the field is changed, any fields specified in the **Results** section automatically populated according to the first rule defined. Any other rule processing stops immediately after the last **Result** field is

populated. This is useful for multiple **Dependency** field matches where one particular rule should be evaluated without changing.

To create an advanced default rule:

1. From the Workbench shortcut bar, select **Demand Mgmt > Request Types.**

   The Request Type Workbench opens.

2. Open a request type.

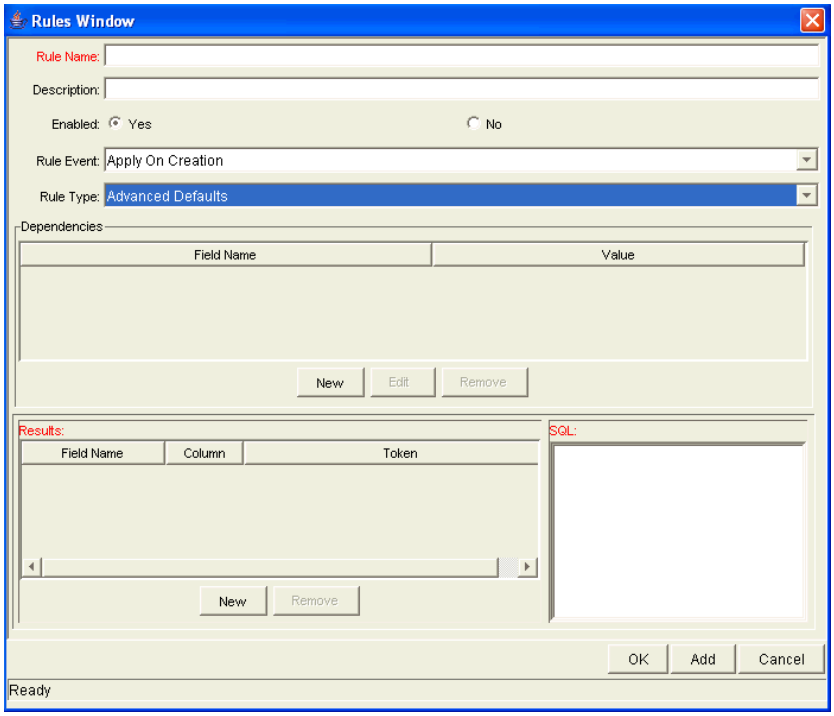
   The Request Type window opens.

3. Click the **Rules** tab.

4. Click **New.**

   The Request Type Rules window opens in Simple Defaults mode.

5. In the **Rule Type** field, select **Advanced Defaults.**



6. Enter the information specified in the following table:

Field Name	Description
Rule Name	The name of the new rule.
Description	A description of the new rule.
Enabled	Selects if the rule is available to the system. Selecting **Yes** means the rule is available to the system.
Rule Event	Specifies the event that triggers the rule. ■ **Apply On Creation.** The rule fires when the request is created, filling in the specified **Results** fields. ■ **Apply On Field Change.** The rule will fire when one of the Dependencies fields is changed to the specified value. ■ **Apply On Field Change And Stop Processing Rules.** The rule will fire when one of the **Dependencies** fields is changed to the specified value, and all subsequent rules in the **Rules** tab will not.
Rule Type	The type of rule. **Simple Defaults** or **Advanced Defaults.**

7. In the **Dependencies** section, click **New.**

   The Dependencies window opens. This window selects a field or fields to trigger the rule.

8. Enter the information specified in the following table:

Field Name	Description
Field	Selects the field from the auto-complete. Request default rules cannot be configured to trigger from a multiple select auto-complete. Do not choose a multiple select auto-complete field for the **Field.**
Value	The value of the field.
All Values	Use all values of the field. Selecting **Yes** disables the **Value** field.
Field Type	The type of field selected, such as request header type. This field is filled automatically.
Validation Name	The field's type of validation, such as Numeric Text Field - 2 decimals. This field is filled automatically.
Visible Token	The name of the visible token, such as `REQ.VP.KNTA_SCHED_EFFORT`. This field is filled automatically.
Token	The name of the token, such as `REQ.P.KNTA_SCHED_EFFORT`. This field is filled automatically.

9. Click **OK.**

10. In the **Results** section, click **New.**

    The Results window opens. This window selects the fields for the rule to automatically populate.

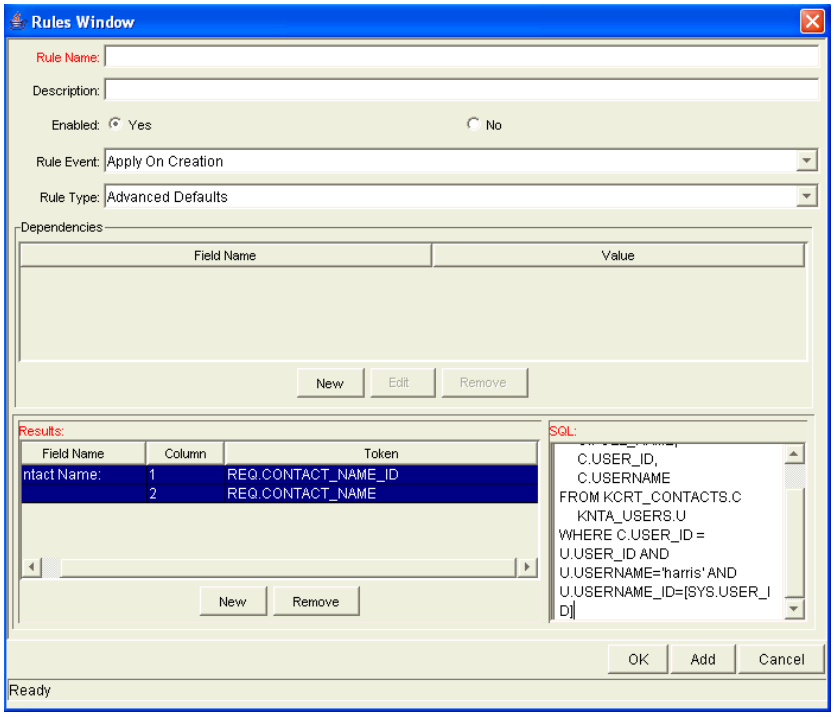
11. Enter the information specified in the following table:

Field Name	Description
Field	Selects the field from the auto-complete.
Field Type	The type of field selected, such as request header type. This field is filled automatically.
Validation Name	The field's type of validation, such as Numeric Text Field - 2 decimals. This field is filled automatically.
Visible Token	The name of the visible token, such as REQ.VP.KNTA_ SCHED_EFFORT. This field is filled automatically.
Token	The name of the token, such as REQ.P.KNTA_SCHED_ EFFORT. This field is filled automatically.

12. Click **OK.**

13. In the **SQL** section, define the SQL statement that is to load values into the fields specified in the **Results** section.

    Each SELECT value is loaded into its corresponding column in the Results table in order. The system validates the SQL statement in the SQL section to ensure that it contains the correct tokens: [SYS] tokens, [AS] tokens, or tokens of fields present in the **Dependencies** section. If the SQL statement is invalid, an error message is displayed.



14. Click **OK.**

15. Click **Save.**

Configuring Commands for Request Types

Request types can have many commands and each command can have many command steps. A command can be viewed as a particular function for an request. Copying a file can be one command and checking that file into version control can be another. To perform these functions, a series of events needs to take place, and these events are defined in the command steps.

An additional level of flexibility is introduced when some commands must only be executed in certain cases. This is powered by the condition field of the commands and is discussed in *Command Conditions* on page 184.

Adding Commands to Request Types

To add commands to request types:

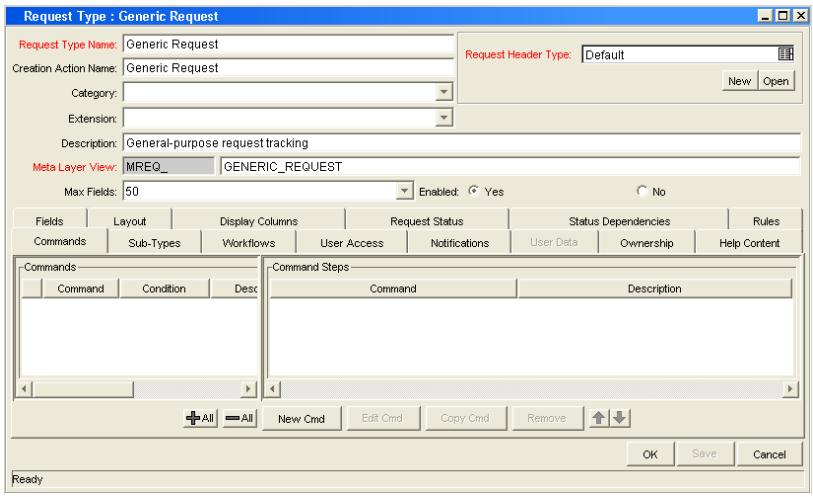
1. From the Workbench shortcut bar, select **Demand Mgmt > Request Types.**

   The Request Type Workbench opens.

2. Open a request type.

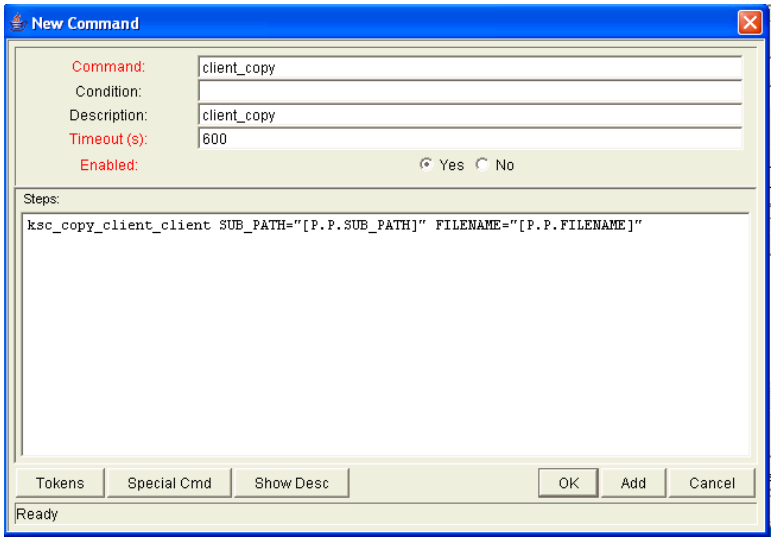
   The Request Type window opens.

3. Click the **Commands** tab.



4. Click **New Cmd.**

   The New Command window opens.



5. 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 184 for more information).
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 too long 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 Request Types

To edit a command on a request type:

1. From the Workbench shortcut bar, select **Demand Mgmt > Request Types.**

   The Request Type Workbench opens.

2. Open a request type.

   The Request 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 184 for more information).
Description	A description of the command.
Timeout	The amount of time the command is allowed to run before its process is terminated. This mechanism is used to abort commands that are hanging or taking too long to run.
Enabled?	Indicates whether the command is enabled for execution.

7. Click **OK.**

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

8. Click **OK.**

Copying Commands in Request Types

To copy a command in a request types:

1. From the Workbench shortcut bar, select **Demand Mgmt > Request Types.**

   The Request Type Workbench opens.

2. Open a request type.

   The Request Type window opens.

3. Click the **Commands** tab.

4. Select the command to copy.

5. Click **Copy Cmd.**

6. Click **OK.**

Deleting Commands in Request Types

To copy a command in a request types:

1. From the Workbench shortcut bar, select **Demand Mgmt > Request Types.**

   The Request Type Workbench opens.

2. Open a request type.

   The Request Type window opens.

3. Click the **Commands** tab.

4. Select the command to delete.

5. Click **Remove.**

6. Click **OK.**

Command Conditions

In many situations, it might be necessary to run a different set of commands depending on the context of execution. 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 5-5*. Be sure to place single quotes around string literals or tokens that will evaluate strings.

Table 5-5. Example conditions (page 1 of 2)

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.

Table 5-5. Example conditions (page 2 of 2)

Condition	Evaluates to
'[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 document *Commands, Tokens, and Validations Guide and Reference*.

Configuring Sub-Types for Request Types

Sub-types are a way to further classify a request type. For example, a request type for software bugs might list each of the application software supported by the IT organization as sub-types.

Adding Sub-Types to Request Types

To add sub-types to the request type:

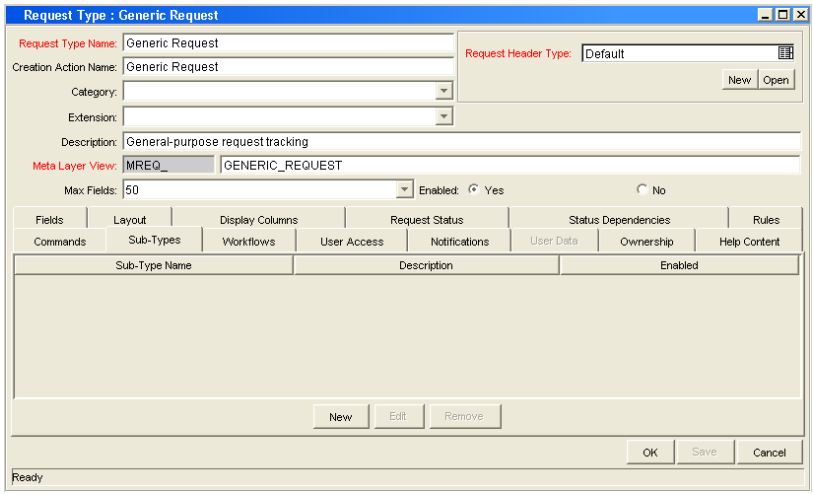
1. From the Workbench shortcut bar, select **Demand Mgmt > Request Types.**

   The Request Type Workbench opens.

2. Open a request type.

   The Request Type window opens.

3. Click the **Sub-Types** tab.



4. Click **New.**

   The Request Sub-Type window opens.

5. Enter the information specified in the following table:

Field Name	Description
Sub-Type Name	The name of the sub-type.
Description	A description of the sub-type.
Enabled	Select to make the sub-type available to the system. Select **Yes** to make the sub-type available to the system.

6. Click **OK.**

7. From the **Sub-Types** tab, click **OK.**

Editing Sub-Types for Request Types

To edit a sub-type:

1. From the Workbench shortcut bar, select **Demand Mgmt > Request Types.**

   The Request Type Workbench opens.

2. Open a request type.

   The Request Type window opens.

3. Click the **Sub-Types** tab.

4. Select a sub-type, and then click **Edit.**

   The Request Sub-Type window opens.

5. Enter the information specified in the following table:

Field Name	Description
Sub-Type Name	The name of the sub-type.
Description	A description of the sub-type.
Enabled	Select to make the sub-type available to the system. Select **Yes** to make the sub-type available to the system.

6. Click **OK.**

7. From the **Sub-Types** tab, click **OK.**

Deleting Sub-Types from Request Types

To delete sub-types from a request type:

1. From the Workbench shortcut bar, select **Demand Mgmt > Request Types.**

   The Request Type Workbench opens.

2. Open a request type.

   The Request Type window opens.

3. Click the **Sub-Types** tab.

4. Select the sub-type to delete, and then click **Remove.**

5. Click **OK.**

Configuring Request Types to Work with Workflows

You can set up request types to work with all workflows, or only selected workflows.

Adding Workflows to Request Types

To add workflows to the request type:

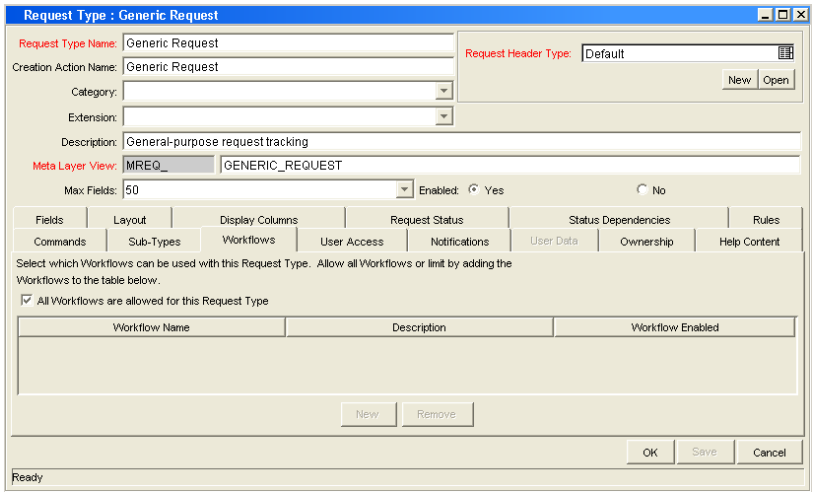
1. From the Workbench shortcut bar, select **Demand Mgmt > Request Types.**

   The Request Type Workbench opens.

2. Open a request type.

   The Request Type window opens.

3. Click the **Workflows** tab.



4. Do one of the following:

   To let all workflows use this request type, select the **All Workflows are allowed for the Request Type** option.

   To specify the workflows that can use the request

   a. Clear the All **Workflows are allowed for the Request Type** option.

   b. Click **New.**

      The Workflow window opens.

   c. In the **Workflow** field, select a workflow.

   d. Click **OK.**

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

Deleting Workflows from Request Types

To delete workflows from the request type:

1. From the Workbench shortcut bar, select **Demand Mgmt > Request Types.**

   The Request Type Workbench opens.

2. Open a request type.

   The Request Type window opens.

3. Click the **Workflows** tab.

4. Select a workflow to delete, and then click **Remove.**

5. Click **OK.**

Configuring Participants for Request Types

You can give users different levels of access to request types.

Adding Participants to Request Types

To add participants to the request type:

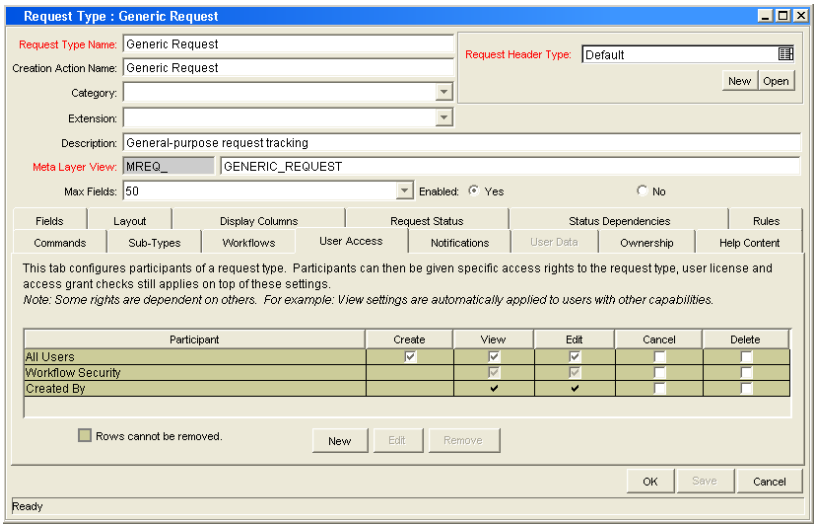
1. From the Workbench shortcut bar, select **Demand Mgmt > Request Types.**

   The Request Type Workbench opens.

2. Open a request type.

   The Request Type window opens.

3. Click the **User Access** tab.



4. Click **New.**

   The Participant Security window opens.

5. In the list, select one of the following security types:

   - **Enter a Security Group Name.** Select a security group to act upon the workflow step. Selecting a security group changes the name of the auto-complete field to **Security Group.** The security type is dynamically changed to **Security Group.**
   - **Enter a Username.** Select a user to act upon the workflow step. Selecting a user changes the name of the auto-complete to **Username.** The security type is dynamically changed to **Username.**
   - **Enter a Standard Token.** Select a standard token to act upon the workflow step. Selecting a standard token changes the name of the auto-complete to **Standard Token.** The security type is left undefined. Select a standard token from the auto-complete. The **Security Type** field is defined based on the standard token chosen.

- **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. To open the Token Builder window and 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.

The participant is added to the **User Access** tab.

6. Add the attributes for the participant.

   Attributes are attached to a participant by clicking **Create, View, Edit, Cancel,** or **Delete.**

7. In the **User Access** tab, click **OK.**

   The changes to the request type are saved.

Editing Participants on Request Types

To edit participants of a request type:

1. From the Workbench shortcut bar, select **Demand Mgmt > Request Types.**

   The Request Type Workbench opens.

2. Open a request type.

   The Request Type window opens.

3. Click the **User Access** tab.

4. Select a participant to edit, and then click **Edit.**

   The Participant Security window opens.

5. Edit the attributes for the participant.

   Attributes are attached to a participant by clicking **Create, View, Edit, Cancel,** or **Delete.**

6. In the **User Access** tab, click **OK.**

   The changes to the request type are saved.

Deleting Participants from Request Types

To delete participants from the request type:

1. From the Workbench shortcut bar, select **Demand Mgmt > Request Types.**

   The Request Type Workbench opens.

2. Open a request type.

   The Request Type window opens.

3. Click the **User Access** tab.

4. Select a participant to delete, and then click **Remove.**

5. Click **OK.**

Configuring Notifications for Request Types

You can configure a request type to send notifications based on field contents. You can send notification at different times, different intervals, different events, and to different recipients.

Adding Notifications

To add a notification:

1. From the Workbench shortcut bar, select **Demand Mgmt > Request Types.**

   The Request Type Workbench opens.

2. Open a request type.

   The Request Type window opens.

3. Click the **Notifications** tab.

4. Click **New.**

   The Add Notification for Step window opens.

5. Configure the **Setup** tab.

   For information on how to configure the **Setup** tab, see *Configuring the Setup Tab* on page 194.

6. Configure the **Message** tab.

   For information on how to configure the **Message** tab, see *Configuring Message Tab* on page 197.

7. Click **OK.**

   The **Notifications** tab lists the notifications added.

8. Click **OK.**

Configuring the Setup Tab

To configure the **Setup** tab:

1. From the Workbench shortcut bar, select **Demand Mgmt > Request Types.**

   The Request Type Workbench opens.

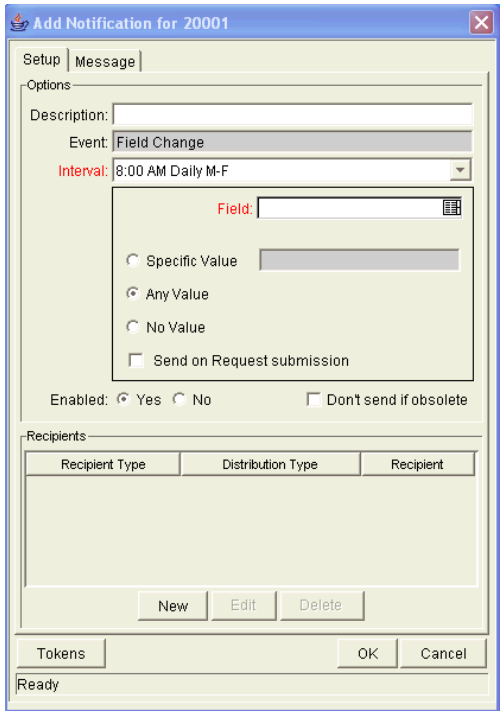
2. Open a request type.

   The Request Type window opens.

3. Click the **Notifications** tab.

4. Click **New.**

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



5. In the **Options** section, enter the information specified in the following table:

Field Name	Description
Description	A brief description of the notification.
Event	The type of event that triggers sending the notification. **Field Changes** is the default and cannot be edited.
Interval	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 batch and send to occur. The following is a list of the available interval options: ■ **8:00AM Daily M-F** ■ **Hourly Daily M-F** ■ **Immediate**
Field	Selects the request type field that triggers the notification from the list. If a change occurs in the selected field, the notification is sent.

Field Name	Description
Specific Value	Send the notification when the selected field is the specified value. Selecting Specific Value enabled the text field. Enter the value in the text field. Selecting **Specific Value** deselects **Any Value** and **No Value.**
Any Value	Send the notification when the selected field is changes to any value. Selecting **Any Value** deselects **Specific Value** and **No Value.**
No Value	Send the notification when the selected field is empty. Selecting **No Value** deselects **Specific Value** and **Any Value.**
Send on Request Submission	Send the notification when the request is first submitted.
Enabled	Make the notification available to the system. Selecting **Yes** makes the notification available to the system.
Don't send if obsolete	Do not send the notification if the trigger values are no longer true. For repeating messages: ■ **8:00AM Daily M-F** ■ **Hourly Daily M-F** For example, if a notification is sent hourly when the field is empty, the notification will automatically stop when the field has a value.

6. To configure the **Recipients** section:

   a. In the **Recipients** section, click **New.**

      The Add New Recipient window opens.

   b. Click **To, Cc,** or **Bcc.**

   c. To specify the recipient:

      i. **Enter a Username.** Select a user as the recipient of the notification. Selecting a user changes the name of the auto-complete to **Username.** The security type dynamically changes to **Username.**

      ii. **Enter an Email Address.** Select an email address as the recipient of the notification. Selecting an email address changes the name of the auto-complete to **Email Address.** The security type is dynamically changed to **Email Address.**

iii. **Enter a Security Group.** Select a security group as the recipient of the notification. Selecting a security group changes the name of the auto-complete to **Security Group.** The security type is dynamically changed to **Security Group.**

iv. **Enter a Standard Token.** Select a standard token to act upon the workflow step. Selecting a standard token changes the name of the auto-complete to **Standard Token.** The security type is left undefined. Select a standard token from the auto-complete. The **Security Type** field is defined based on the standard token chosen.

v. **Enter a User Defined Token.** Select a user defined token to act upon the workflow step. Selecting a user defined token changes the name of the auto-complete to **User Defined Token.** The security type is dynamically changed to a list. The **Tokens** button is enabled. Click **Tokens** to open the Token Builder window and select a token. Select one of the following from the list:

- **Username.** The selected token resolves to a username.
- **User ID.** The selected token resolves to a user ID.
- **Security Group Name.** The selected token resolves to a security group.
- **Security Group ID.** The selected token resolves to a security group ID.

d. Click **OK.**

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

Configuring Message Tab

You can construct the notification's message to ensure that it contains the correct information for the recipient. For example, if a notification is sent to instruct you that a request requires your approval, the message should instruct you to log onto Mercury IT Governance Center and update the request status. Additionally, the notification should include a link (URL) to the referenced request.

The following features to make notifications simpler to configure and use:

- Select from a number of pre-configured notification templates to more quickly construct the body of your message.
- The body of the notification can be plain text or HTML.

- Multiple tokens can be included in the notification. These tokens will resolve to information relevant to the recipient. For example, you can include tokens for the URL to the request approval page, information on request status and priority, and emergency contacts.

To configure the message tab:

1. From the Workbench shortcut bar, select **Demand Mgmt > Request Types.**

   The Request Type Workbench opens.

2. Open a request type.

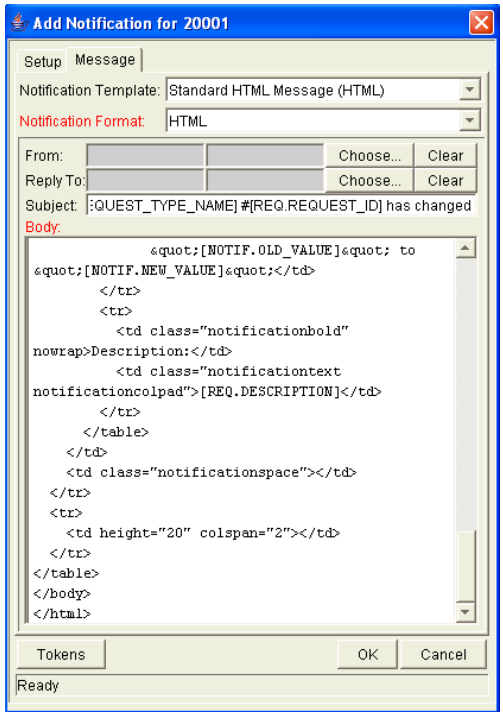
   The Request Type window opens.

3. Click the **Notifications** tab.

4. Click **New.**

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

5. Click the **Message** tab.



6. In the **Notification Template** field, select a template.

   This updates the contents in the **Body** section with the information defined in the selected template.

7. In the **Notification Format** field, select the message format.

   The HTML format provides more flexibility in creating the look and feel of the notification. You can write and test the HTML code in any HTML editor, and then paste the content into the Body window.

8. Enter values for the **From** and **Reply to** fields, as follows:

   a. To the right of the **From** or **Reply to** field, click **Choose.**

      The Email Header Field window opens.

   b. Select the recipient, as follows:

      - **Enter a Username.** Select a user as the recipient of the notification. Selecting a user changes the name of the auto-complete to **Username.** The security type is dynamically changed to **Username.**
      - **Enter an Email Address.** Select an email address as the recipient of the notification. Selecting an email address changes the name of the auto-complete to **Email Address.** The security type is dynamically changed to **Email Address.**
      - **Enter a Standard Token.** Select a standard token to act upon the workflow step. Selecting a standard token changes the name of the auto-complete to **Standard Token.** The security type is left undefined. Select a standard token from the auto-complete.
      - **Enter a User Defined Token.** Select a user defined token to act upon the workflow step. Selecting a user defined token changes the name of the auto-complete to **User Defined Token.** The **Tokens** button is enabled. Click **Tokens** to open the Token Builder window and select a token. Select one of the following from the list:
        - **Username.** The selected token resolves to a username.
        - **User ID.** The selected token resolves to a user ID.
        - **Security Group Name.** The selected token resolves to a security group.
        - **Security Group ID.** The selected token resolves to a security group ID.

   c. Click **OK.**

      The **Message** tab lists the selected recipients.

9. Construct the body of the message.

   When constructing the body, consider using the following:

   - Token for the URL to the Request Detail page.
   - Token for the URL to the package (Workbench or standard interface).
   - Tokens in the body of the message:

     Click **Tokens** to access the Token Builder window where tokens can be added to the message body.

   - Tokens related to specific package lines:

     Add tokens to the **Linked Token** field to include tokens that resolve information related to the individual package line.

10. Click **OK.**
11. From the **Notifications** tab, click **OK.**

Editing Notifications

To edit a notification:

1. From the Workbench shortcut bar, select **Demand Mgmt > Request Types.**

   The Request Type Workbench opens.

2. Open a request type.

   The Request Type window opens.

3. Click the **Notifications** tab.
4. Select a notification that you want to change, and then click **Edit.**

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

5. Edit the **Setup** tab (see *Configuring the Setup Tab* on page 194).
6. Edit the **Message** tab (see *Configuring Message Tab* on page 197).
7. Click **OK.**

Copying Notifications

To copy a notification:

1. From the Workbench shortcut bar, select **Demand Mgmt > Request Types.**

   The Request Type Workbench opens.

2. Open a request type.

   The Request Type window opens.

3. Click the **Notifications** tab.

4. Select the notification you want to copy, and then click **Copy.**

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

   To edit the **Setup** tab, see *Configuring the Setup Tab* on page 194.

   To edit the **Message** tab, see *Configuring Message Tab* on page 197.

5. In the **Notifications** tab, click **OK.**

   The changes to the request type are saved.

Deleting Notifications

To delete a notification:

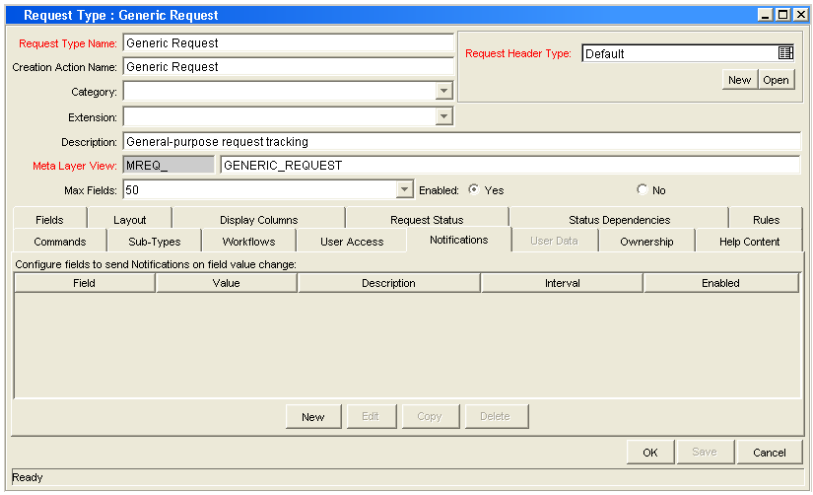
1. From the Workbench shortcut bar, select **Demand Mgmt > Request Types.**

   The Request Type Workbench opens.

2. Open a request type.

   The Request Type window opens.

3. Click the **Notifications** tab.



4. Select a notification that you want to remove, and then click **Delete.**
5. Click **OK.**

Configuring Ownerships of Request Types

To configure request type ownership groups, you add security groups to the **Ownership** tab. If no ownership groups are associated with the entity, the entity is treated as global, and any user who can edit request types can edit, copy, or delete the entity. For more information about access grants, see the document *Security Model Guide and Reference*.

If a security group is disabled or loses the its ability to edit a request type, that group can no longer edit the entity.

Adding Ownerships to Request Types

To add an ownership:

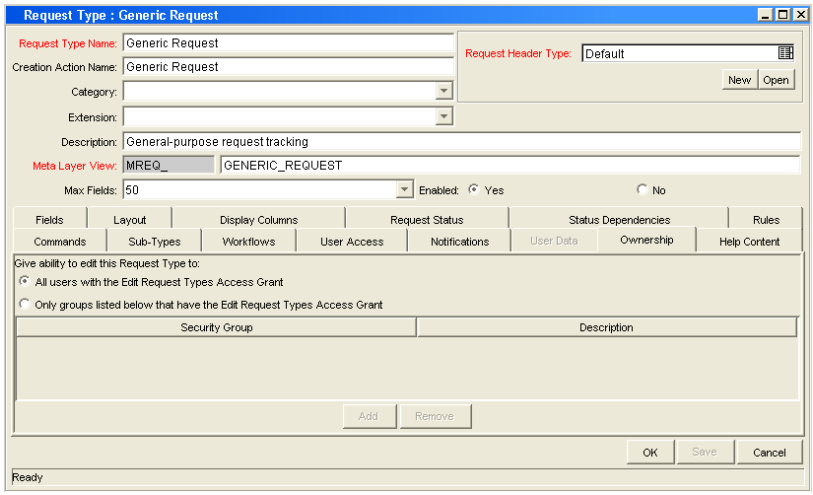
1. From the Workbench shortcut bar, select **Demand Mgmt > Request Types.**

   The Request Type Workbench opens.

2. Open a request type.

   The Request Type window opens.

3. Click the **Ownership** tab.



4. Select the ownership option.

   The All users with the Edit Request Type access grant option gives all users who can edit request types access to the request type. The Only groups listed below that have the Edit Request Type access grant option requires selected groups to be added to the ownership of the request type.

   If you select, **Only groups listed below that have the Edit Request Type,** complete the following:

   a. In the **Ownership** tab, click **Add.**

      The Add Security Groups window opens.

   b. In the **Security Groups** field, select the security groups.

      The Validate window opens.

   c. Select one or more security groups, and then click **OK.**

      The Add Security Groups window lists the selected security groups.

   d. Click **OK.**

      From the **Ownership** tab, the **Security Group** column lists the selected security groups.

5. Click **OK.**

Deleting Ownerships from Request Types

To delete an ownership:

1. From the Workbench shortcut bar, select **Demand Mgmt > Request Types.**

   The Request Type Workbench opens.

2. Open a request type.

   The Request Type window opens.

3. Click the **Ownership** tab.

4. Select an ownership.

   The All users with the Edit Request Type access grant option gives all users who can edit request type access to the request type. The Only groups listed below that have the Edit Request Type access grant option requires selected groups to be added to the ownership of the request type.

5. Click **Remove.**

6. Click **OK.**

Configuring Help Contents for Request Types

You can provide accessible online information to users who are processing the requests. Configure the request type to display additional, custom information about the request, sections or fields.

To add help to the request type:

1. From the Workbench shortcut bar, select **Demand Mgmt > Request Types.**

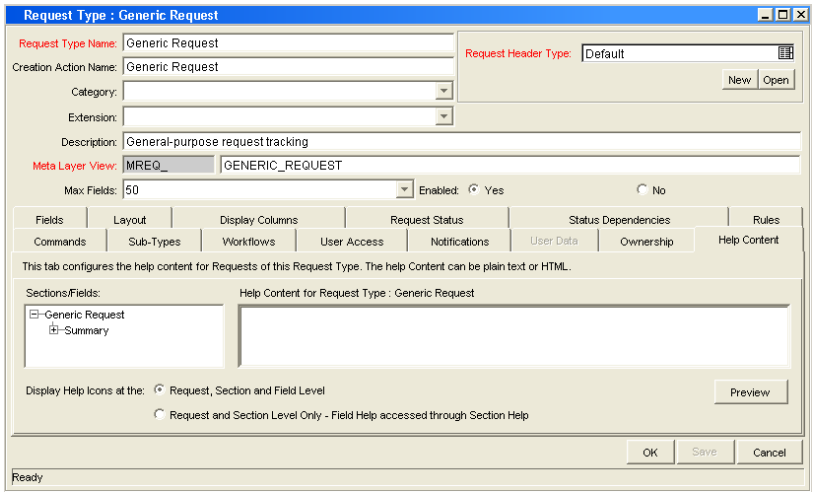
   The Request Type Workbench opens.

2. Open a request type.

   The Request Type window opens.

3. Click the **Help Content** tab.

4. In the **Sections/Fields** section, select the item to which content is to be added.



5. In the **Help Content for Request Type** section, enter the help content for the selected item.

   Enter plain text or HTML-formatted text.

6. To see what the text looks like in the actual help display, click **Preview.**

7. Enter additional, optional information to further define help content for those items.

8. From the **Display Help Icons at the:** field, specify how the help icons are to be displayed in the standard interface.

   - **Request, Section and Field Level.** Display a help icon (question mark) beside each request, section and field that has associated help content.
   - **Request and Section Level Only.** Does not display the help icon at the individual field level. Any help content defined for the fields can be accessed from the section level help.

9. From the **Help Content** tab, click **Save.**

Configuring Request Header Types

Request header types define the collection of fields that appear in the header region of the requests. Request header types typically include more general information that is tracked between multiple types of requests. This can include such information as who logged the request, its priority, and a description of the issue.

Every request type must include a request header type. A single request header type can be used for multiple request types.

Table 5-6 lists the Mercury-supplied request header types.

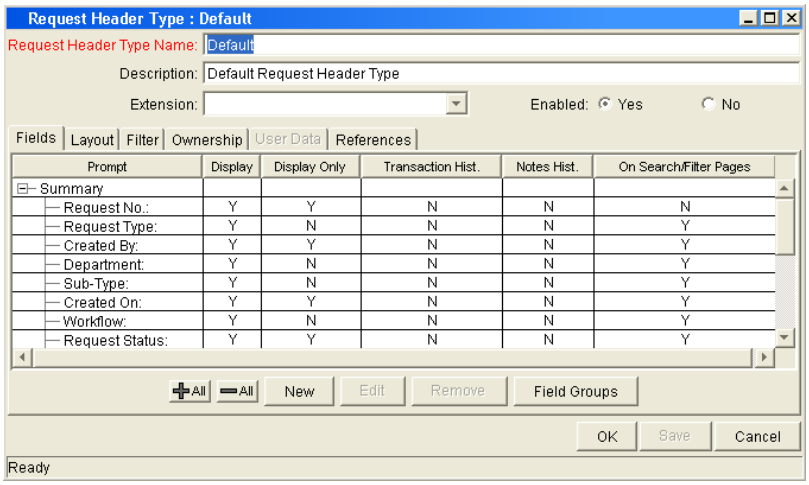
Table 5-6. Request header types

System Header Type (REFERENCE)	Description
Default	The default request header type. Includes a percentage complete (% Complete) field.
Comprehensive	Displays all information. Consistent with previous versions of Mercury IT Governance Center.
Simple	Displays only the most essential information.
Departmental	An example request header type for simple cross-departmental requests.
Application	An example request header type for simple cross-application requests.
Help Desk	An example request header type for help desk requests, including contact and assignment information.

Overview of Request Header Types

Request header types contain a set of standard predefined fields that can be enabled or disabled. Request header types can also contain custom fields. Request header types are created and configured in the Request Header Type window (*Figure 5-7*).

Figure 5-7. Request Header Type window



The main components of a request header type are as follows:

- **General information.** General information includes basic information concerning the request type, such as the request type name and the request type category. See *Configuring General Information for Request Header Types* on page 211.
- **Fields.** Every request header type has a set of predefined fields. The **Fields** tab is used to create additional fields for the request header type. Creating fields for request header type is identical to creating fields for request types. See *Configuring Fields for Request Types* on page 146.
- **Layout.** The layout of fields can be configured using the **Layout** tab. Laying out fields for request header types is identical to laying out fields for request types. See *Configuring Layouts for Request Types* on page 157.
- **Filter.** Several fields on request header types can be filtered to display specific information in a request. See *Configuring Filters for Request Header Types* on page 212.

- **Ownership.** Configure who can edit the request header type. Configuring who can edit the request header type is identical to configuring who can edit a request type. See *Configuring Ownerships of Request Types* on page 202.
- **User Data.** Product entities such as packages, workflows, requests and projects include a set of standard fields that provide information about those entities. While these fields are normally sufficient for day to day processing, user data fields provide the ability to capture additional information specific to each organization. User data is defined under the **User Data** tab. If there are no user data fields, the **User Data** tab is disabled.
- **References.** Displays reference information concerning the request header type.
- **Field Groups.** Request header type field groups are a way for Mercury IT Governance Center to distribute a collection of fields required for certain functionality. For more information, see *Request Header Type Field Groups* on page 208.

Request Header Type Field Groups

Request header type field groups are a way for Mercury IT Governance Center to distribute a collection of fields required for certain functionality. For example, Mercury Demand Management distributes a collection of fields for Service Level Agreements in a SLA Field Group.

Field group fields behave just as normal fields do, with the restrictions that you cannot remove them except by removing the entire field group and you might not be able to modify some of the field properties. *Table 5-7* on page 209 lists the request header type field groups that are delivered with various Mercury IT Governance Center products.

You can add field groups to request header types by clicking **Field Groups** in the Request Header Type window.

Each request header type field group has a custom token prefix that allows the user to access the data of that field by using the format:

`REQ.P.<field_group_token_starting_with_KNTA_>`

When field groups are associated with existing request types (through the request header type definition), Mercury IT Governance database tables are updated to handle this new configuration. Because of the scope of database changes, the Database Statistics should be rerun on your database. Instructions

for this are included in the *System Administration Guide and Reference*. Contact the application administrator for help with this procedure.

Figure 5-8. Request Header Type Field Groups window

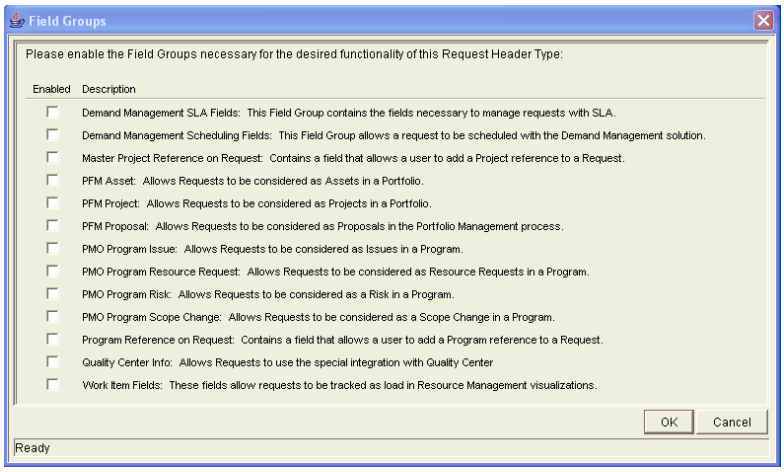


Table 5-7. Request header type field groups (page 1 of 2)

Field Group	Description
Demand Management SLA	Contains the fields necessary to manage requests with SLA.
Demand Management Scheduling	Allows a request to be scheduled with Mercury Demand Management
Master Project Reference on Request	Contains a field that allows a user to add a project reference to a request
PMO Program Issue	Allows requests to be considered as issues in a program
PMO Program Resource Request	Allows requests to be considered as resource requests in a program
PMO Program Risk	Allows requests to be considered as risks in a program
PMO Program Scope Change	Allows requests to be considered as scope changes in a program
Portfolio Management Proposal	Contains the fields necessary to create a PFM proposal
Portfolio Management Project	Contains the fields necessary to create a PFM project

Table 5-7. Request header type field groups (page 2 of 2)

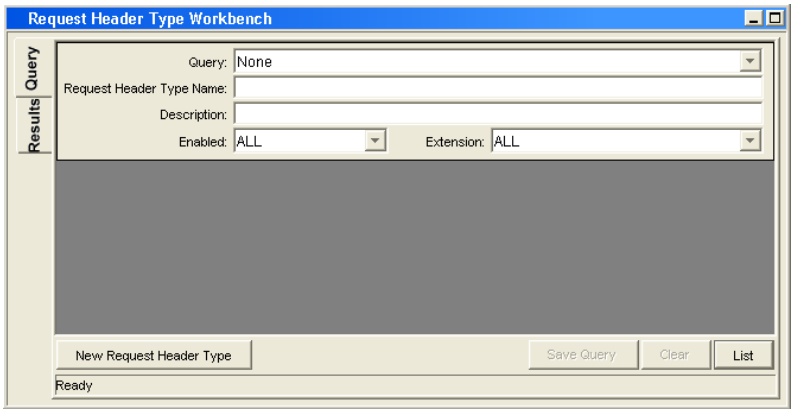
Field Group	Description
Portfolio Management Asset	Contains the fields necessary to create a PFM asset
Program Reference on Request	Contains a field that allows a user to add a program reference to a request
Work Item Fields	Work item fields contain fields that allow requests to be scheduled as a work item

Opening the Request Header Type Workbench

To open the Request Header Type Workbench:

1. From the Workbench shortcut bar, select **Demand Mgmt > Request Header Types.**

   The Request Header Type Workbench opens.



Configuring General Information for Request Header Types

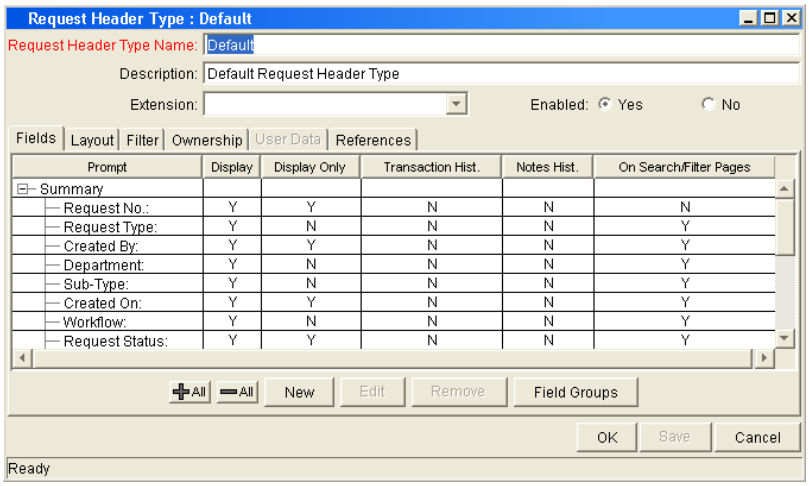
To configure the general information of a request header type:

1. From the Workbench shortcut bar, select **Demand Mgmt > Request Header Types.**

   The Request Header Type Workbench opens.

2. Open a request header type.

   The Request Header Type window opens.



3. Enter the information specified in the following table:

Field Name	Description
Request Header Type Name	The name of the request header type.
Description	A useful description of how the request header type is used.
Extension	For request header types created for a Mercury Deployment Management extension. Select the extension from the list.
Description	A useful description of how the request header type is used.
Enabled	Indicates whether or not the request header type is available to Mercury IT Governance Center.

4. Do one of the following:

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

Configuring Filters for Request Header Types

To configure filters for a request header type:

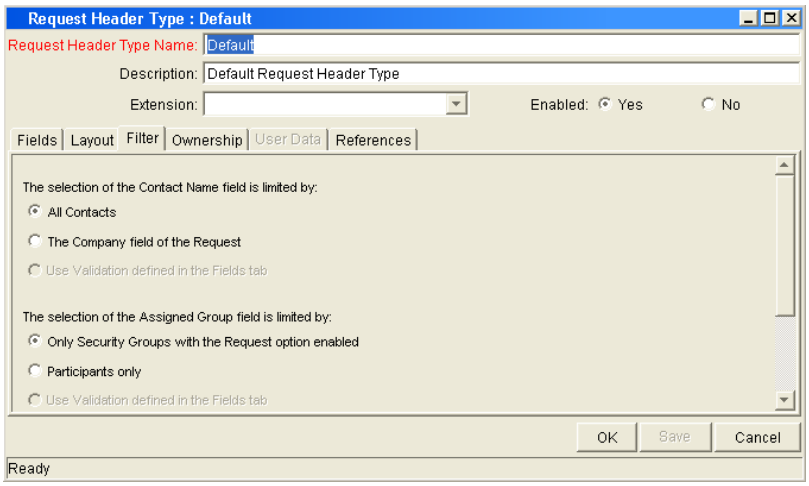
1. From the Workbench shortcut bar, select **Demand Mgmt > Request Header Types.**

   The Request Header Type Workbench opens.

2. Open a request header type.

   The Request Header Type window opens.

3. Click the **Filter** tab.



4. Enter the information specified in the following table:

Field Name	Description
This section of the Contact Name field is limited by:	■ **All Contacts.** Limit the number of contact names seen in the **Contact Name** field when creating or updating a request header type by selecting one of the contact name options available in the Filter tab. Selecting this option will display all users with no restrictions on the list of contact names. ■ **The Company field of the Request.** Users can limit the number of contact names they would see in the **Contact Name** field when creating or updating a request header type by selecting one of the contact name options available in the **Filter** tab. Selecting this option will restrict the list of contact names the user would see to those found in the **Company** field of the request. ■ **Use Validation defined in the Fields tab.** Selecting this option will restrict the list of contact names the user would see to those found in the **Contact Name** field of the request.
This section of the Assigned Group Field is limited by:	■ **Only Security Groups with the Request option enabled.** Users can limit the number of group names they would see when creating or updating a request header type by selecting one of two Assigned Group options available in the **Filter** tab. Selecting this option will restrict the list of group names the user would see to only those security groups where the request option is enabled. ■ **Participants only.** Users can limit the number of group names they would see when creating or updating a request header type by selecting one of two Assigned Group options available in the **Filter** tab. Selecting this option will restrict the list of group names the user would see to participants in the request. ■ **Use Validation defined in the Fields tab.** Selecting this option will restrict the list of contact names the user would see to those found in the **Contact Name** field of the request.

Field Name	Description
This section of the Assigned To field is limited by:	■ **Only users who are in Security Groups with the Request option enabled.** Limit the number of user names seen in the **Assigned To** field when creating or updating a request header type by selecting one of two Assigned To options available in the **Filter** tab. Selecting this option restricts the list of user names the user would see to only those security groups where the request option is enabled. ■ **Participants only.** Users can limit the number of user names they would see in the **Assigned To** field when creating or updating a request header type by selecting one of two Assigned To options available in the **Filter** tab. Selecting this option restricts the list of user names the user would see to participants of the request. In this instance, participants are defined as: the assigned user, the creator of the request, members of the assigned group, or members of the workflow. ■ **Use Validation defined in the Fields tab.** Selecting this option will restrict the list of contact names the user would see to those found in the **Contact Name** field of the request.

5. Click **OK.**

Chapter

Configuring Contacts

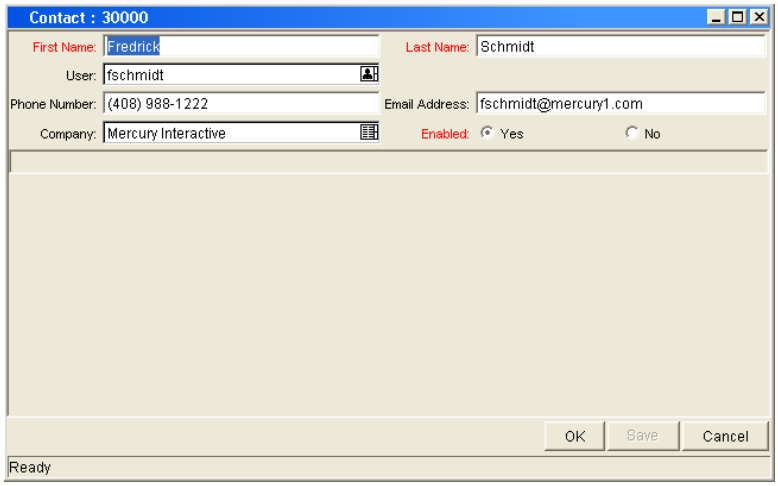
In This Chapter:

Overview of Contacts

Contacts are resources used as a point of reference or information. Contacts must have a valid Mercury IT Governance Center username and the company they work for must be included in the validation, CRT - Company Validation. Contact information can be added for users in Mercury IT Governance Center as well as external users.

Contacts are created in the Contact window. The Contact window consists of a general information section and a large section reserved for potential user data fields (*Figure 6-1*).

Figure 6-1. Contact window



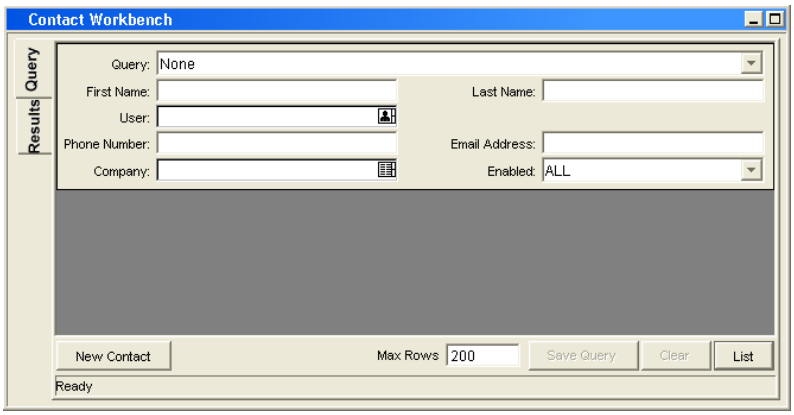
Opening the Contact Workbench

To open the Contact Workbench:

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

   The Workbench opens.
3. From the Workbench shortcut bar, select **Demand Mgmt > Contacts.**

   The Contact Workbench opens.



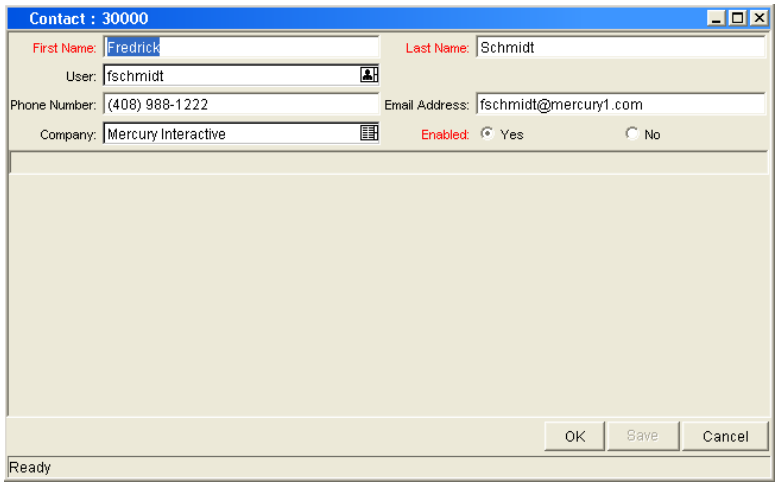
Creating Contacts

To create a new contact:

1. From the Workbench shortcut bar, select **Demand Mgmt > Contacts.**

   The Contact Workbench opens.
2. Click **New Contact.**

   The Contact window opens.



3. Enter the information specified in the following table:

Field Name	Description
First Name	The first name of the contact.
Last Name	The last name of the contact.
User	The Mercury IT Governance Center username of the contact. This field is populated from the **KNTA - User Id - All Validation** auto-complete and cannot be edited.Select a username from the validation auto-complete.
Phone Number	The phone number of the contact.
Email Address	The email address of the contact.
Company	The company employing the contact. This field is populated from **CRT - Company Validation** auto-complete and cannot be edited. Select a company from the validation auto-complete.
Enabled	Make the notification template available to the system. Select **Yes** to make the notification available to the system.

4. In the Contact window, click **OK.**

   The changes to the notification template are saved.

Chapter

Configuring Notification Templates

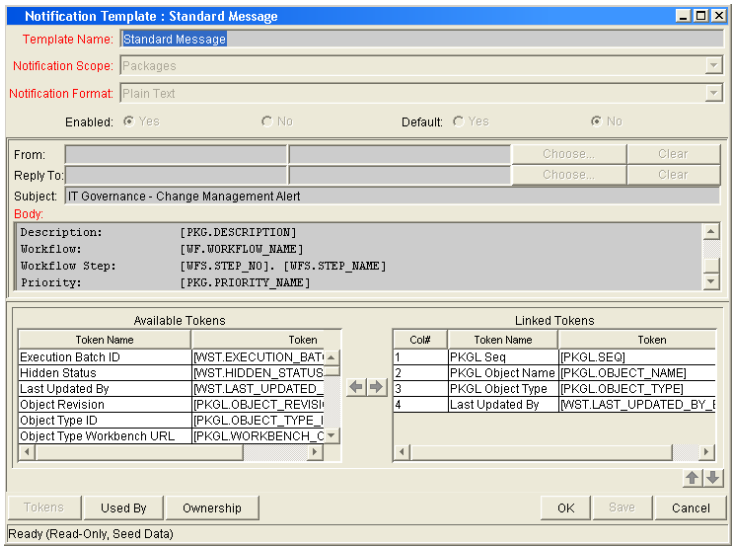
In This Chapter:

Overview of Notification Templates

Notification templates are pre-configured notifications that can be used to quickly construct the body of your message (see *Figure 7-1*). Notification templates are used with the following Mercury IT Governance Center entities:

- Tasks
- Projects
- Requests
- Packages
- Releases
- Workflows
- Reports

Figure 7-1. Notifications Template window



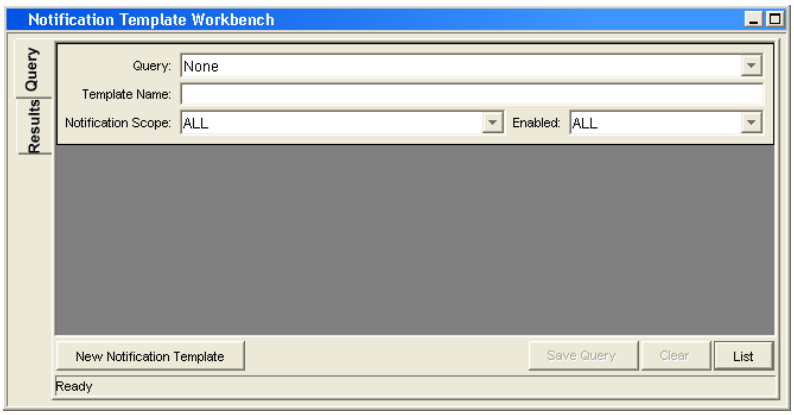
Opening the Notification Template Workbench

To open the Notification Template Workbench:

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

   The Workbench opens.
3. From the shortcut bar, select **Configuration > Notification Templates.**

   The Notification Template Workbench opens.



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 230.

Creating Notification Templates

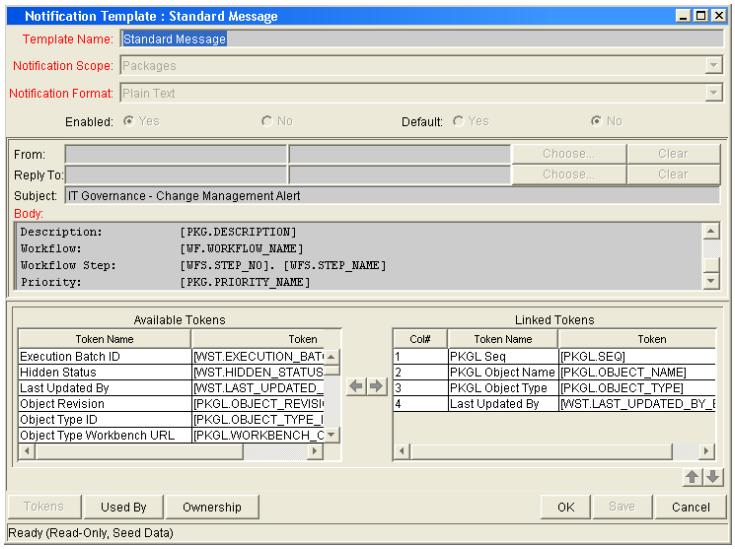
To create a new notification template:

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.



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.**

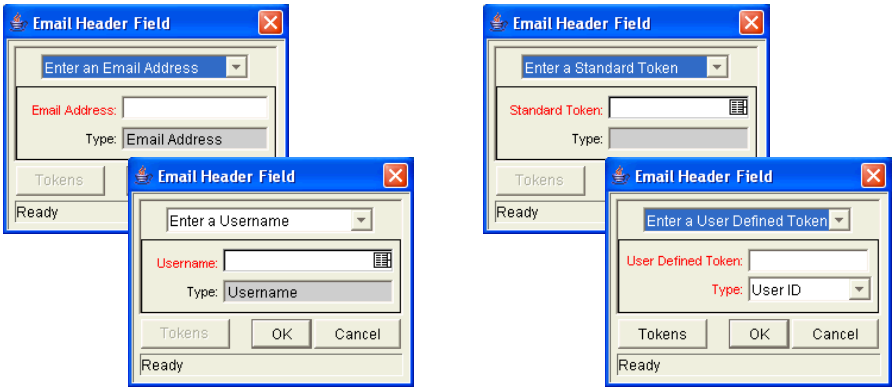
4. Enter a **From** address.

   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.



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.

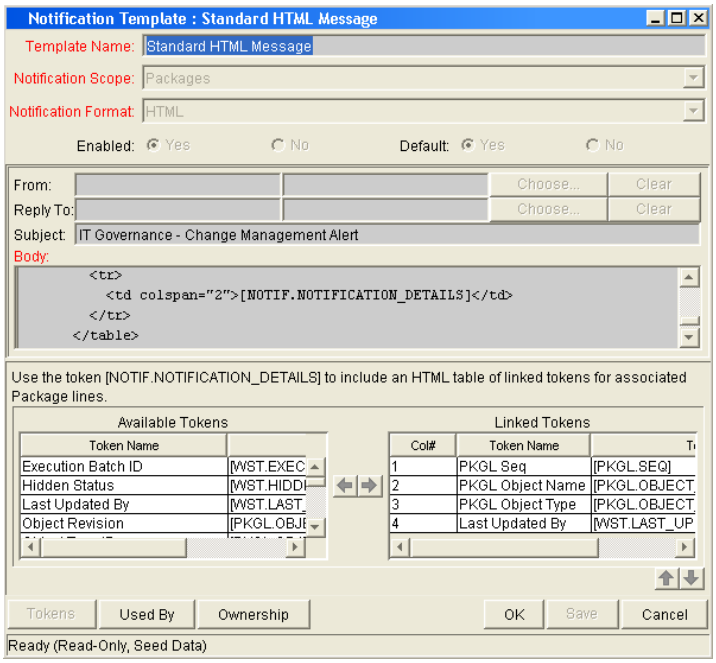
   c. Enter the information in the required field.

   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 Mercury Deployment Management must include the token '[NOTIF.NOTIFICATION_DETAILS]' within the <*body*> tags to incorporate linked tokens.



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 226.

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 document *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.

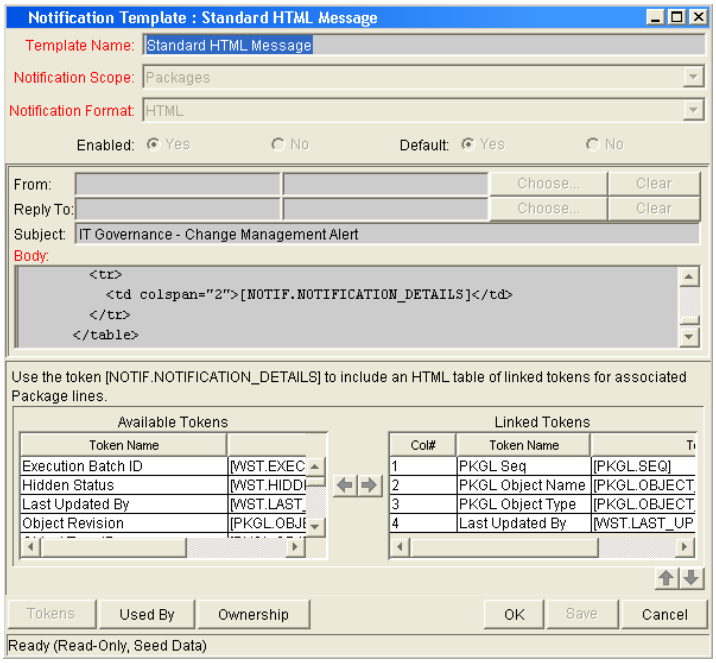
To configure the ownership of a notification template:

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.



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

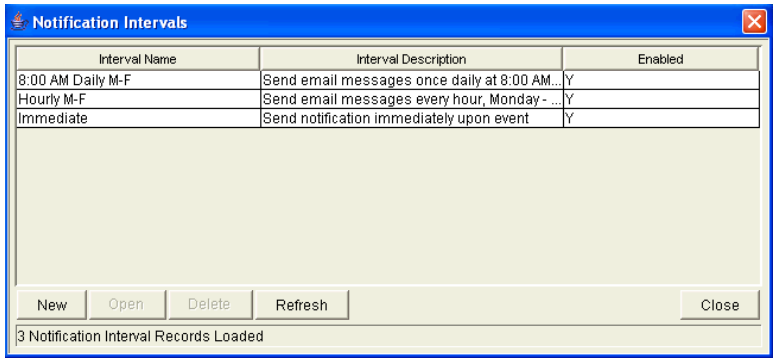
To create a new notification template:

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

   The Notification Template Workbench opens.

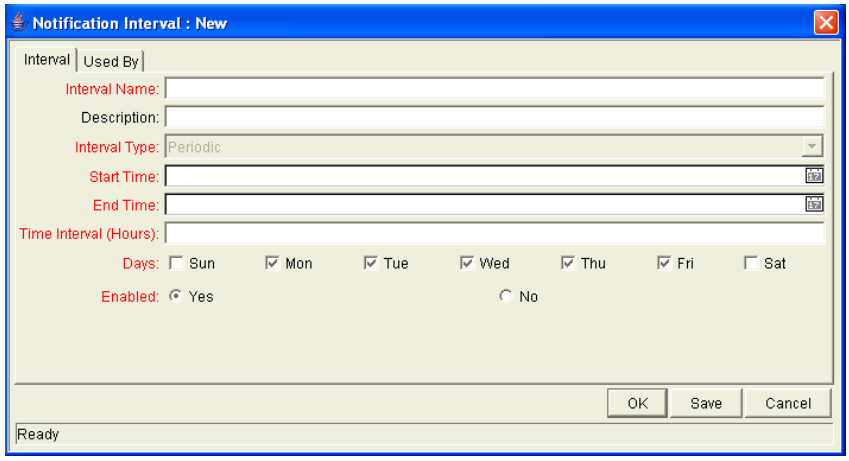
2. From the menu, select **Notification Templates > Intervals.**

   The Notification Intervals window opens.



3. Click **New.**

   The Notification Interval: New window opens to the **Interval** tab.



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

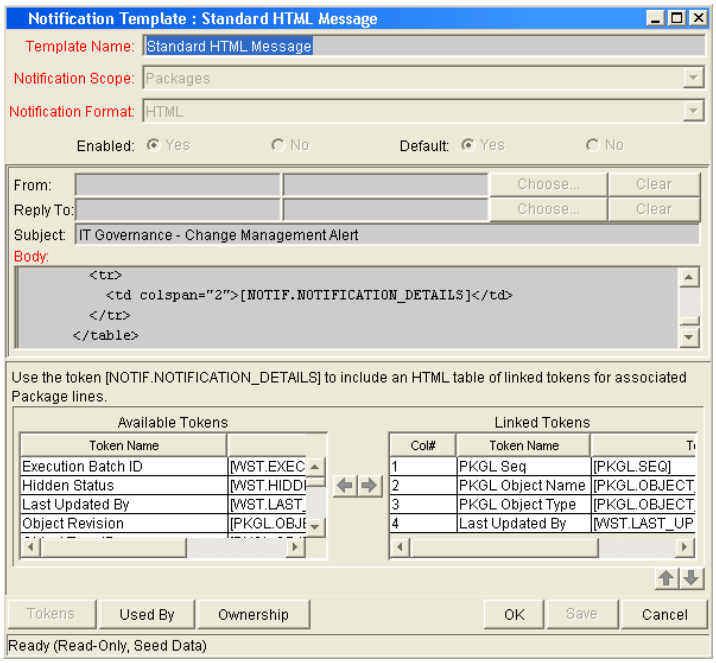
To check the usage of a notification template:

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

   The Notification Template Workbench opens.

2. Open notification template.

   The Notification Template window opens.



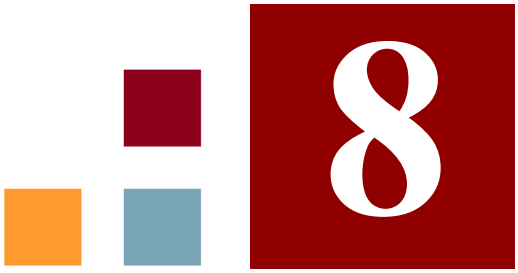
3. Click **Used By.**

   The Used By window opens and lists all references to the notification template.

4. Click **OK.**

5. In the Notification Template window, click **OK.**

Chapter



Configuring User Data

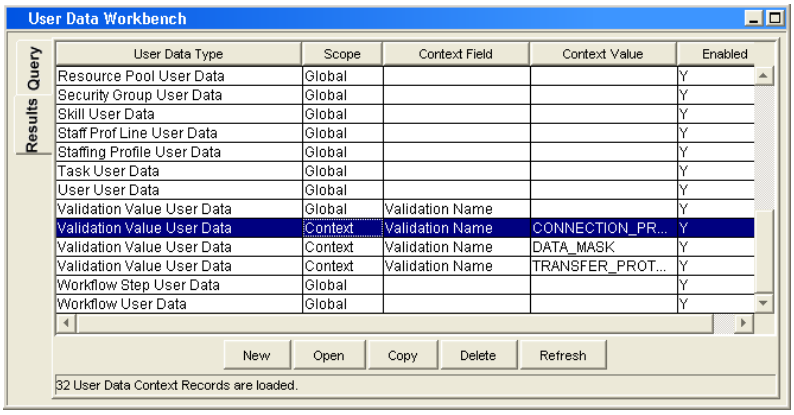
In This Chapter:

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 8-1* the **Results** tab in User Data Workbench shows a partial list of the available user data types.

Figure 8-1. User data types



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. Mercury IT Governance 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 235.
- **Fields.** Used to create additional fields for a user data type. See *Creating User Data Fields* on page 237.
- **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 246.

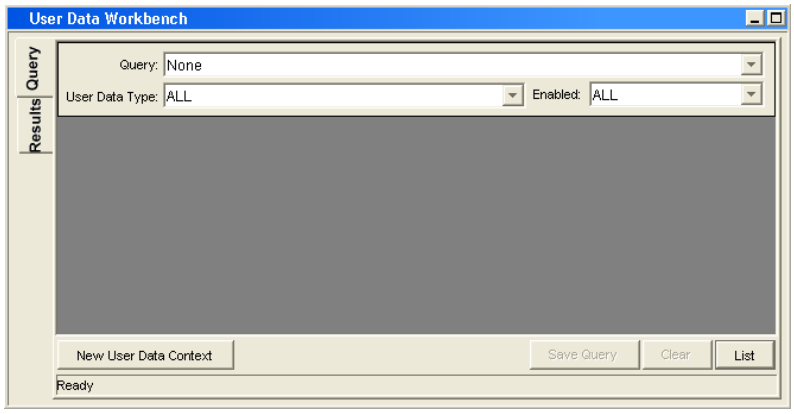
Opening the User Data Workbench

To open the User Data Workbench:

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

   The Workbench opens.
3. From the shortcut bar, select **Configuration > User Data.**

   The User Data Workbench opens.



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.**

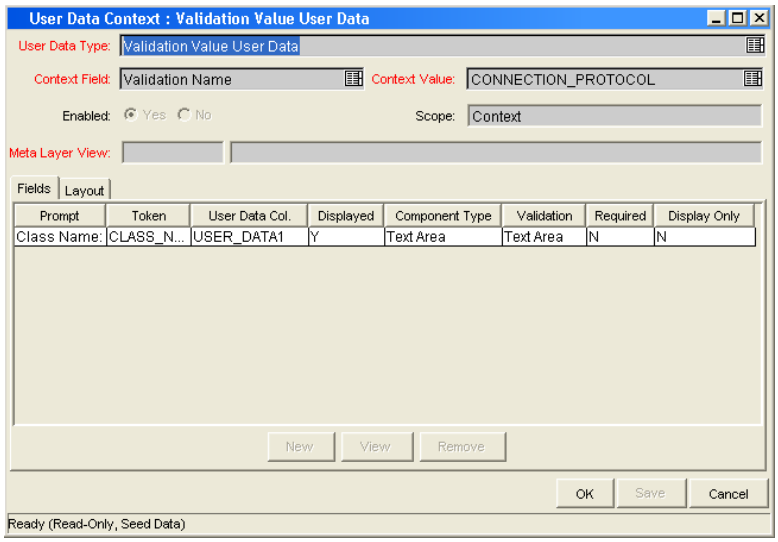
   The User Data Workbench opens.

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 Mercury IT Governance 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.



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 Mercury IT Governance 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 Mercury IT Governance 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

Note

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.**

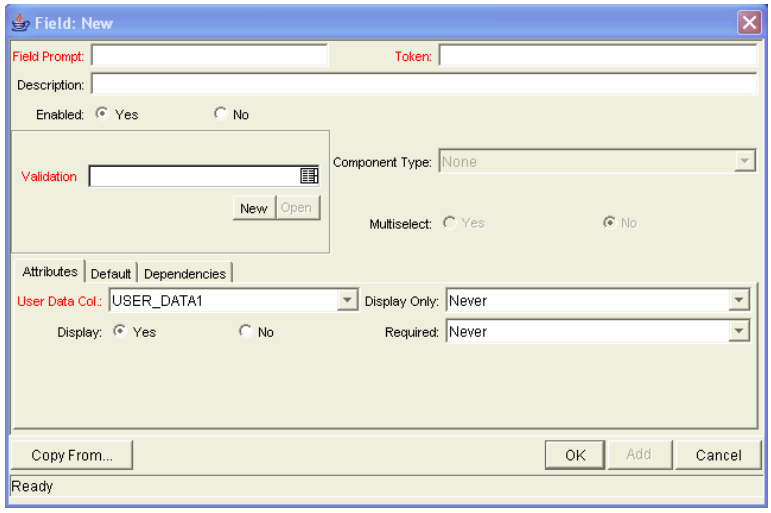
   The User Data Workbench opens.

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.



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.
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.

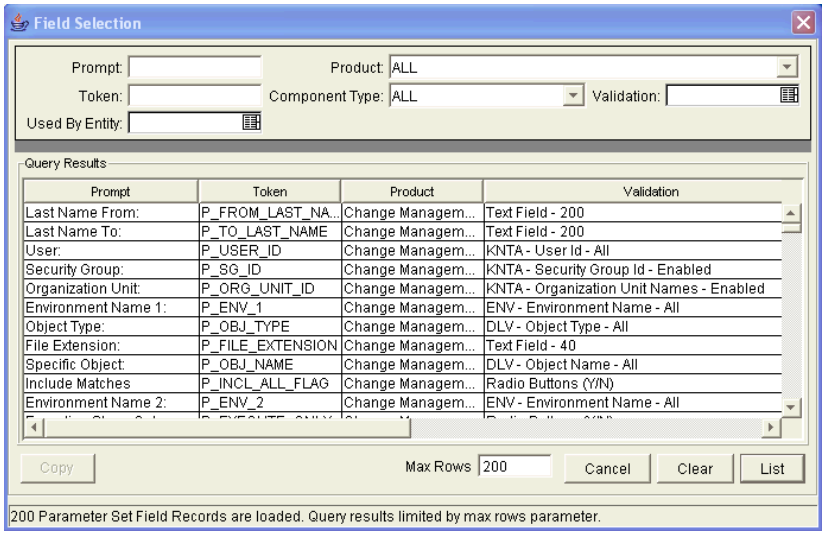
5. Click **Copy From.**

   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.



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.**

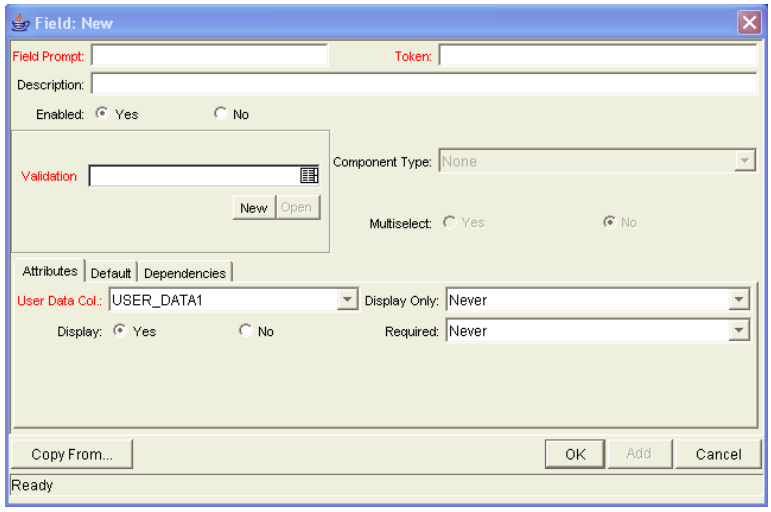
   The User Data Workbench opens.

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.



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 237.

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 8-1*.
- Become required after another field meets a logical condition defined in *Table 8-1*.

Table 8-1. 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.

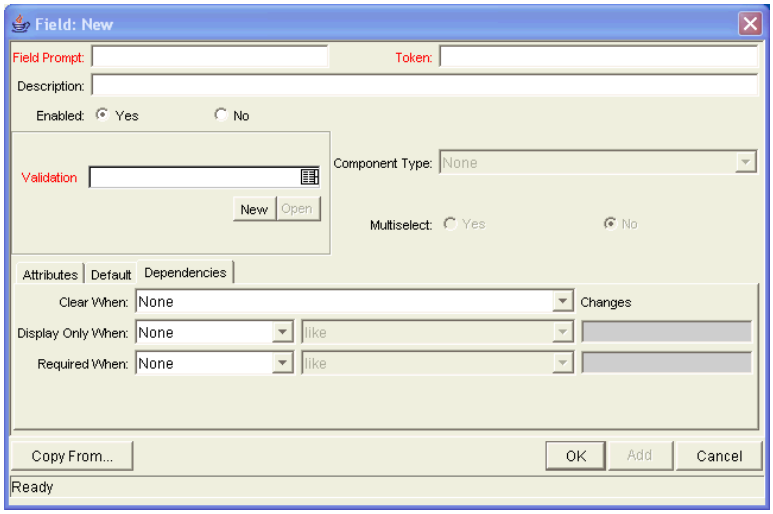
2. Select an existing user data type or create a new user data type.

   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.



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.

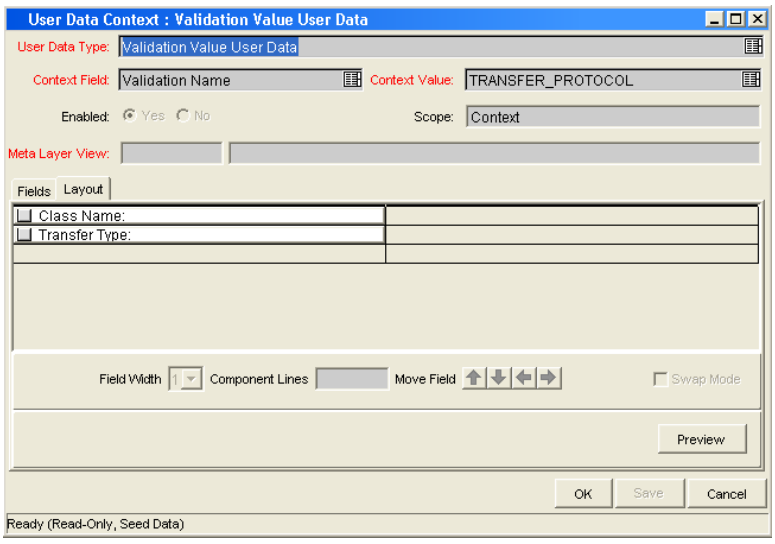
3. Select the field to remove, and then click **Remove.**

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 8-2. User Data window Layout tab



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

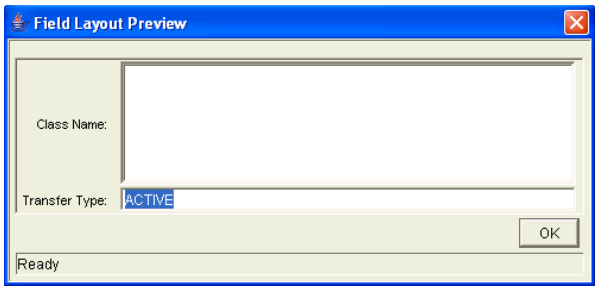
You can check to see what the layout looks like to users.

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 8-3. Preview mode



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.

Appendix 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 ____

Workflow Step	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. Validation Information

Validation Information	Value
Existing Validation?	
New Validation?	
Validation Type: (text field, auto-complete, list, and so on.)	
Validation Definition (list of values or SQL)	

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

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

Decision Workflow Step Worksheets

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

Workflow Step	Value
Step Name	
Goal/Result of Step	
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)	

Table A-6. Workflow step [decision], step number ____ validation

Validation Information*	Value
Existing Validation?	
New Validation?	
Validation Type: (text field, auto-complete, list, and so on.)	
Validation Definition (list of values or SQL)	

Subworkflow Workflow Step Worksheets

Table A-7. Workflow step [subworkflow], step number ____ (page 1 of 2)

Workflow Step	Value
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)

Workflow Step	Value
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, list, and so on)	
Validation Definition (list of values or SQL)	

Request Type Configuration Sheets

Table A-9. Request type information

Information	Value
Request Type Name	
Associated Request Header Type	
Description	

Table A-10. Request type field information

#	Field Name	Description
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		

Table A-11. Request type commands

Command	Value
Goal of Commands	
Command Steps	
Conditions (When to execute)	

Table A-12. Request type status values

Status	Corresponds to Workflow Step

Table A-13. Request type attributes

Information	Value
Field Name	
Validation	
Field Behavior:	
Attributes (select one):	■ Display ■ Editable ■ Display Only ■ Required
Default Value	
Users/Security Groups allowed to View Field	
Users/Security Groups allowed to Edit Field	
Status Dependencies:	
Clear field when Status = ?	
Display only when Status = ?	
Reconfirm only when Status = ?	
Required when Status = ?	
Auto-Population Behavior:	
Auto-Population triggered by (Depends on) Field:	
Value used to populate Field:	

Table A-14. Field validation information

Validation Information	Value
Existing Validation?	
New Validation?	
Validation Type: (text field, auto-complete, list, and so on.)	
Validation Definition (list of values or SQL)	
Notes on Validation (data masks, auto-complete behavior, and so on.)	

Table A-15. Request header type information

Request Header Type	Value
Request Header Type Name	
Associated Request Type(s)	
Description	
Associated Field Group(s)	

Table A-16. Existing request header type field information

Prompt	Display	Display Only	Transaction History	Notes History	Search Filter Page
Request No					
Request Type					
Created By					
Department					
Sub-Type					
Created On					
Workflow					
Request Status					
Priority					
Application					
Contact Name					
Assigned To					
Assigned Group					
Contact Phone					
Request Group					
Contact Email					
Description					
Company					
% Complete					

Index

D

E

F

S

T

U