Mercury IT Governance Center™

Mercury Demand Management™: Configuring a Request Resolution System

Version: 6.0

MERCURY™

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

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

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

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

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

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

Publication Number: ITG60ConfigReqRel1104A

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

In This Chapter:

About This Document

Mercury Demand Management™ is a Mercury IT Governance Center™ product that can be configured to follow your business processes. At the core of these processes is an integrated workflow engine that enables you to digitize both simple and complex processes.

Creating workflows to follow your business processes requires a variety of Mercury IT Governance entities configured to work together. This document details those entities and how they can be configured to support your business processes.

This document contains the following chapters:

- Chapter 1, *Introduction*, on page 13

  This chapter describes the document and provides an overview of the configuration process.

- Chapter 2, *Gathering Process Requirements and Specifications*, on page 25

  This chapter discusses the information that needs to be collected before developing a request tracking and resolution system.

- Chapter 3, *Configuring Workflows*, on page 43

  This chapter discusses how to create workflows. Workflows represent business processes. Workflows allow you to map business rules and processes to your organization. Information discussed in this chapter includes the following:

  - Demand Management workflows
  - Change Management workflows
  - Release Management workflows

- Chapter 4, *Configuring Workflow Components,* on page 117

  This chapter discusses how to build workflow components. Included in this discussion is how to create execution workflow steps, decision workflow steps and subworkflow workflow steps. Information discussed in this chapter includes the following:

  - Demand Management workflows
  - Change Management workflows
  - Release Management workflows

- Chapter 5, *Configuring Request Types and Request Header Types,* on page 155

  This chapter discusses how to build request types and request header types. Requests are a fundamental work unit of Mercury Demand Management systems. Request types and request header types determine the fields found on requests and the behavior of those fields.

- Chapter 6, *Configuring Contacts,* on page 247

  This chapter discusses how to build contacts. Contacts are users of Mercury IT Governance Center used as a point of reference or information.

- Chapter 7, *Configuring Notification Templates,* on page 253

  This chapter discusses how to build notification templates. Notification templates are pre-configured notifications that can be used to quickly construct the body of a message.

- Chapter 8, *Configuring User Data,* on page 269

  This chapter discusses how to build user data fields. Product entities such as packages, workflows, requests and projects include a set of standard fields that provide information. User data fields are additional fields you can configure to accompany those product entities.

- Chapter 9, *Rolling Out a Request Tracking and Resolution System,* on page 303

  This chapter provides things to consider and checklist when you are ready to rollout a request tracking and resolution system.

- Appendix A: *Worksheets* on page 315

  This appendix provides a series of worksheets to help gather information required to build a workflow.

Who Should Read This Document

This document is for the following audience types:

- Application developers/configurators

For More Information

For information about audience types, see *Guide to Documentation*.

Prerequisite Documents

Prerequisite documents for this document are:

- *Guide to Documentation*
- *Key Concepts*
- *Getting Started*

For More Information

For information about these documents and how to access them, see *Guide to Documentation*.

Related Documents

Related documents for this document are:

- *Commands, Tokens, and Validations Guide and Reference*
- *Open Interface Guide and Reference*
- *Reports Guide and Reference*
- *Security Model Guide and Reference*

For More Information

For information about these documents and how to access them, see *Guide to Documentation*.

Overview of Mercury Demand Management

Mercury Demand Management enables you to model, automate, enforce, and measure your best-practice business processes. End users complete a configurable request form using a standard Web browser. Each request type has a workflow associated with it 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. And, the included Web-based Mercury IT Governance Dashboard delivers the right information to anyone with a browser.

Mercury Demand Management Concepts

- **Requests.** A request is the fundamental work unit of the request resolution piece of Mercury Demand Management. End-users create requests and then submit them along a resolution process (defined in the workflow). The request page contains all information typically required to complete a specific business process.

  Each request has an associated request type that determines which fields are included in the request page. As the request goes through its steps, you will be prompted for any information necessary to bring the request to

closure. Each request includes a corresponding workflow. Once fields are completed and the request submitted, the workflow is used to move the request through the process.

- **Requests and Workflow Interactions.** Request types are tightly integrated with the workflow engine. Each request type has a number of possible statuses (such as **Assigned**, **On Hold,** or **New**). Each status can be linked to a particular workflow step and drive field-level behavior. Additionally, since request statuses can be linked to field behavior through status dependencies, field properties (such as whether the field can be edited or is required) can also be altered as the request proceeds along a workflow.

  It is also possible to configure the workflow to execute the commands contained in the request type at specific points in the process (workflow step). The request type commands are executed at execution workflow steps.

- **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 will be tracked between multiple types of request. This can include such information as who logged the request, its priority, and a description of the issue.

  Request header types contain a set of standard predefined fields that can be enabled or disabled. Request header types can also contain custom fields.

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

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

  Use Mercury Demand Management to model and enforce best practice request resolution processes. Each type of request leverages an optimized workflow tailored for specific business rules to collect required data, gain appropriate approvals and perform specific actions.

  As a request progresses along its workflow, pre-configured steps can trigger:

  - Email notifications to be sent to the proper participants.
  - Automated command-line executions to be performed.

- Field defaulting and logical updates, ensuring that the correct information to resolve a request is available.
- Deployments to be created and initiated or project tasks to be updated.

When the request has been taken to the end of its workflow, it is considered resolved.

- **Request Types.** A request type is a general category that defines the structure of the request. Mercury Demand Management includes such predefined system request types as the Bug request type and Enhancement request type. The fields that are used when a request is created can be customized based on the request type. Request type definitions control much of the request-specific logic in the resolution process. This includes such things as:
  - Defaulting a specific workflow to use when processing this type of request
  - Custom field definition and behavior
  - Layout
  - Data access and security (who can view or edit the request)
  - Configuration security (who can alter the request type)
  - Notifications
- **Security Groups.** Security groups are constructed to provide a set of users with access to specific product screens and functions. Each security group is configured with a set of access grants that enable specific access. Users are then associated with one or more security groups.

  A user's security group memberships determine which windows can be viewed or edit, which workflows can be used, and which workflow steps can be acted on. Each user can be a member of multiple security groups. The collection of security groups to which a user belongs defines that user's role and access within Demand Management.
- **Workflows.** A workflow is a logical series of steps that define the process that requests follow. The workflow can be configured to handle virtually any business practice. This allows a department to create workflows to automate existing processes, rather than forcing users to adopt a fixed set of processes to perform their work.

Workflow steps can range in usage from functional approvals to actual system-level executions. For example, it is possible to create an execution step to automatically connect to another application and import data into the Mercury Demand Management database.

Overview of Request Resolution Systems

This document provides the building blocks required to support Mercury Demand Management. The integrated workflow engine enables you to digitize both simple and complex process requirements at all levels of your IT operations. This support ranges from high-level collaboration requirements, such as an executive review of your IT portfolio, to detailed automation, such as deploying code to an application server. Consider the case of the process illustrated in *Figure 1-1* on page 22. This simplified workflow process includes the following workflow steps:

- **Approve Release.** In this decision workflow step, a request has been submitted for approval. A Mercury IT Governance Center user will approve or reject the request. The configuration elements required for this step include the following:

  - A decision workflow step is configured with security groups, notifications, timeouts, and transitions. Only members of the specified security groups can approve or reject this request. A notification is sent out to the members of the specified security groups. If no one approves or rejects the request within an allotted time, another notification is sent.

  - A request type configured with fields and security groups. On the request type, the Contact field is mandatory. A user data field is included to track requestor satisfaction.

  - A notification template configured to be used by the decision workflow step.

  - Enabled contacts. The request type is configured with a mandatory field (Contact).

  - A user data field is created to track requestor satisfaction. This user data field is configured as part of all request types.

- **Assign Resource.** In this decision workflow step, a request has been approval and a resource must be assigned. A Mercury IT Governance Center user will assign a user to the request. The configuration elements required for this step include the following:

  - A decision workflow step is configured with security groups and transitions.

  - A request type configured with fields and security groups. On the request type, the Contact field is required. A user data field is included to track requestor satisfaction. The request type is configured to make the Assigned To field mandatory for this workflow step.

  - Enabled contacts. The request type is configured with a required field (Contact).

  - A user data field is created to track requestor satisfaction. This user data field is configured as part of all request types.

- **Complete Release.** In this decision workflow step, the assigned resource completes the request. The configuration elements required for this step include the following:

  - A decision workflow step configured with security groups and transitions.

  - A request type configured with fields and security groups. On the request type, the Contact field is required. A user data field is included to track requestor satisfaction. The request type is configured such that when the request is completed, a notification is sent to the contact.

  - A notification template configured to be used by the request type.

  - Enabled contacts. The request type is configured with a required field (Contact).

  - A user data field is created to track requestor satisfaction. This user data field is configured as part of all request types.

- **Close (Success).** In this execution workflow step, the request is automatically closed.

  - An execution workflow step configured with security groups and transitions. The workflow step is configured to automatically close the request.

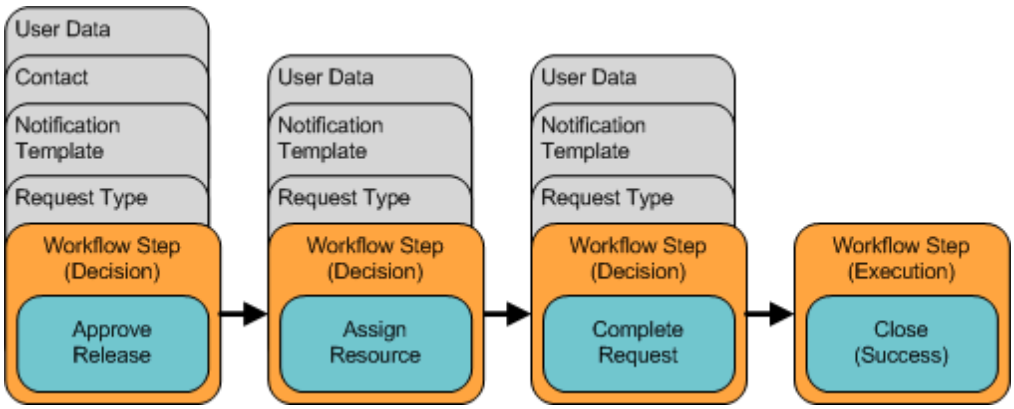


Figure 1-1. Mercury IT Governance Center components

Accessing Mercury IT Governance Center

Businesses often need to control access to certain information and business processes. This can be done to protect sensitive information, such as employee salaries, or to simplify business processes by hiding data that is irrelevant to the user. Mercury IT Governance Center includes a set of features to help control data and process security on the following levels:

- Limiting who can access certain windows or pages
- Limiting who can view or edit certain fields
- Limiting the data displayed in sensitive fields or screens
- Limiting which users can view, create, edit or process Mercury IT Governance Center entities, such as requests, packages, projects, portfolios, and programs
- Limiting which users can view, create or edit Mercury IT Governance Center configuration entities, such as workflows, request types, object types, and security groups
- Limiting which users can alter the security settings

The following features control the data and process security in Mercury IT Governance Center. These features can be combined in a number of ways to provide a secure system:

- **Licenses.** Each user is assigned a license that provides the user with the potential to access to a set of Mercury IT Governance Center product-related screens and functions. Licenses dictate available behavior but need to be used in conjunction with access grants to enable specific fields and functions.
- **Access Grants.** Linked to users through security groups, access grants define which windows and functions users can view, edit, or perform actions in. Access grants also provide varying levels of control over certain entities and fields.
- **Entity-level restrictions.** Settings on the entity that specify who can create, edit, process, and delete Mercury IT Governance Center entities, such as requests, packages, and projects. You can also control which request types and object types can be used with certain workflows. These restrictions are often configured in the configuration entities, such as workflows, request types, and object types.
- **Field-level restrictions.** For each custom field that you define in Mercury IT Governance Center, you can configure when it is visible or editable. For some fields, you can additionally specify which users can view or edit the field.
- **Configuration-level restrictions.** You can specify, using ownership groups settings, which users can modify configuration entities in the system. For example, you can control who is allowed to edit an existing workflow. This allows you to guarantee that only appropriate users are altering your Mercury IT Governance Center-controlled processes.

For More Information

For more information concerning accessing the Mercury IT Governance Center, security groups, and access grants, see *Security Model Guide and Reference*.

Creating Request Resolution Systems

To configure a request resolution system or process:

1. *Gathering Process Requirements and Specifications*

   Before configuring Mercury Demand Management to manage a request tracking and resolution process, collect specific information related to the business process, any information needed to process the request, users who will create and process requests, and the communication devices surrounding the process.

2. *Configuring Workflows*

   Using the information gathered in the *Gathering Process Requirements and Specifications* chapter, build the workflow. This includes setting up required workflow step sources, creating validations to be used by the transitions, and adding steps and transitions to the workflow. Also see *Configuring Workflow Components*.

3. *Configuring Request Types and Request Header Types*

   Using the information gathered in the *Gathering Process Requirements and Specifications* chapter, build the request types. This includes creating and configuring request type fields and field logic. It also includes configuring the request types and workflows to work together. Also see, *Configuring Notification Templates* and *Configuring User Data*.

4. *Rolling Out a Request Tracking and Resolution System*

   It is recommended a formal change management process be followed when configuring the Mercury IT Governance Center. This includes testing configurations, migrating them into a production instance, enabling the processes, and training the user base.

Chapter

2

Gathering Process Requirements and Specifications

In This Chapter:

Overview of Gathering Process Requirements

This chapter discusses the following information that needs to be collected before developing a request tracking and resolution system:

- **Business process.** What are the steps in the process and which steps need to be reviewed and approved?
- **Information needed to resolve the request.** What information needs to be gathered to resolve the request? This information will translate to fields on the request. For example, in order to resolve a software bug type of request, you need to gather information on the bug: such as what is it, how can it be reproduced, and which machines does it occur on.
- **Participants who will create and process request.** What level of security will be placed on this system?
- **Communication devices surrounding the process.** Determine whether to communicate using notifications, the Dashboard, or reports.

Gathering Requirements for Workflows

The first step to configuring a request resolution process is to define the process—the actual steps required to resolve a request. This includes process information such as when to obtain reviews and approvals on the request, who needs to approve the workflow steps, when to execute commands, and the path (transitions) between steps in the process.

Defining Business Flows

Map the business process. This consists of the steps (decisions, conditions, and any executions) and transitions needed to resolve requests. It is helpful to graphically map these processes.

In this phase:

- Identify all decision points in the process
- Determine a flow between steps (transitions). Consider all possible exit values from each step (such as, **Approved, Not Approved, Rework Required,** or **Error**)
- Identify process closure points (success or failure)

The following example illustrates the process design issues that should be considered.

Example: Defining Business Flows

A company needs to configure a resolution process for processing change requests for their financial applications system. This system consists of over ten modules (including billing, accounts payable, accounts receivable, fixed asset management, inventory, reporting, payroll, and cash management). The corporate IT group needs to create a process that can address the complications related to evaluating and approving changes to this system.

Example: Overview of Business Process

The corporate IT group first creates a high-level business process. The process begins when someone in the financial group submits a request for an enhancement.

1. **Submit.** The request is submitted by someone in the financial group. If the priority of the request is **High** or **Critical,** the IT manager and financial group manager are immediately informed.

2. **Validate.** The request is reviewed by the IT manager. A feedback loop is built into the business process at this point, in case more information is needed from the original requestor.

3. **Approve.** The request is approved or rejected by the IT manager.

4. **Schedule.** If the request is approved, the work needed to create the enhancement must be scheduled. If the IT group lacks the necessary resources at the moment, the request is put on hold.

5. **Develop.** The requested enhancement is developed by IT group.

6. **Deploy.** The finished enhancement is deployed to the financial group.

Additional Process Requirement:

When a request is submitted, its Priority should be evaluated. If the request is **Critical** or **High,** the financial manager and IT manager should be informed.

To address this requirement, an execution workflow step is added that would evaluate the request's priority. If the request is **Critical** or **High,** it is routed to another execution workflow step that sends a notification to the appropriate users.

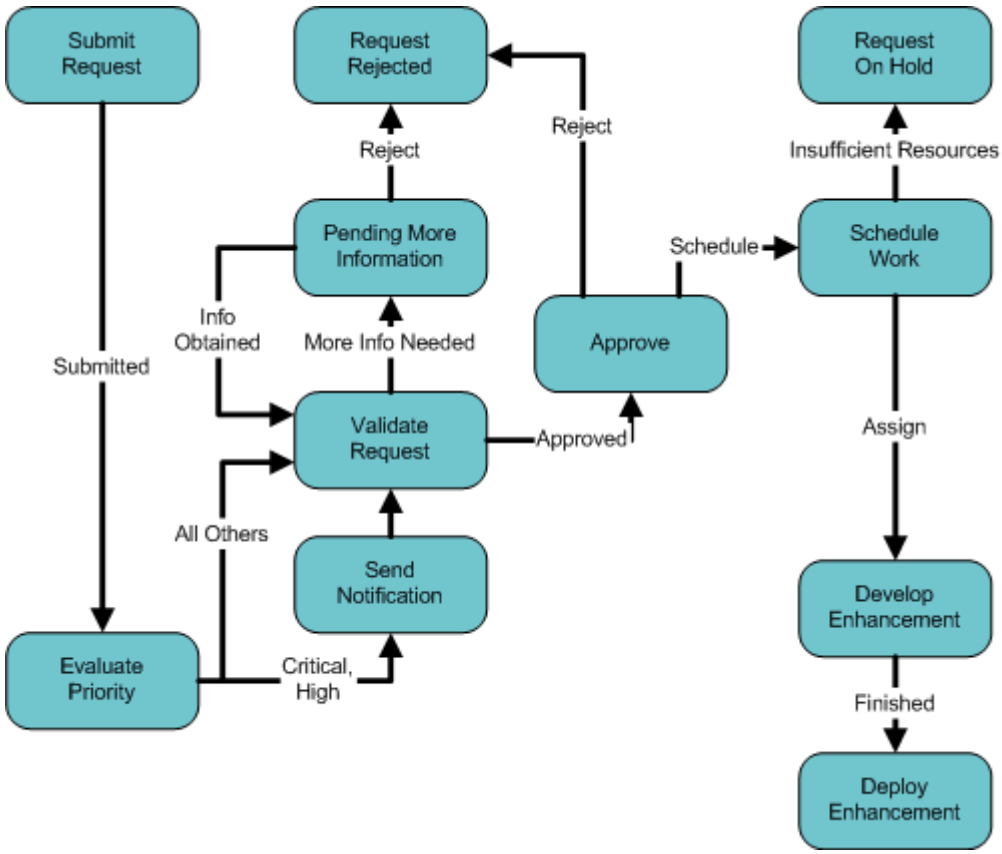


Figure 2-1. Revised business process

Gathering Information Steps in the Process

After designing the business flow of a request resolution process, gather detailed information on each step and transition in the process. This section discusses the information that needs to be collected. Use the worksheet provided in *Worksheets* on page 315 to collect the required information.

For each step in the process, collect the following information:

- **Step name**
- **Description.** Describe the goal of the step.
- **Step Type.** Decision, execution, condition, or subworkflow.
  - Decision step specific information: number of approvals required, timeouts, and so on.
  - Execution step specific information:
    - The desired results of the execution. This will help you to choose the execution type and build any required commands.
    - Execution timing. Determine whether the execution will occur immediately or be processed manually.
  - Subworkflow step specific information
- **Transition values and Validation.** Transition values are the possible results for the step. Depending on the result, the process will proceed in different directions. Use one of Mercury Demand Management's system validations or create a custom one.

Considering Subworkflows

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.

Workflows can be used as subworkflows within a parent workflow. An entire subworkflow is represented by a single step in the parent workflow window's **Layout** tab. This simplifies the potentially complex graphical layout and enables the easy reuse of common workflow configurations.

Example: Using Subworkflows

The IT group decides to use a subworkflow for the development portion of their process. This subworkflow can be referenced in one part of the process. *Figure 2-2* on page 31 illustrates where the IT group could implement a subworkflow.

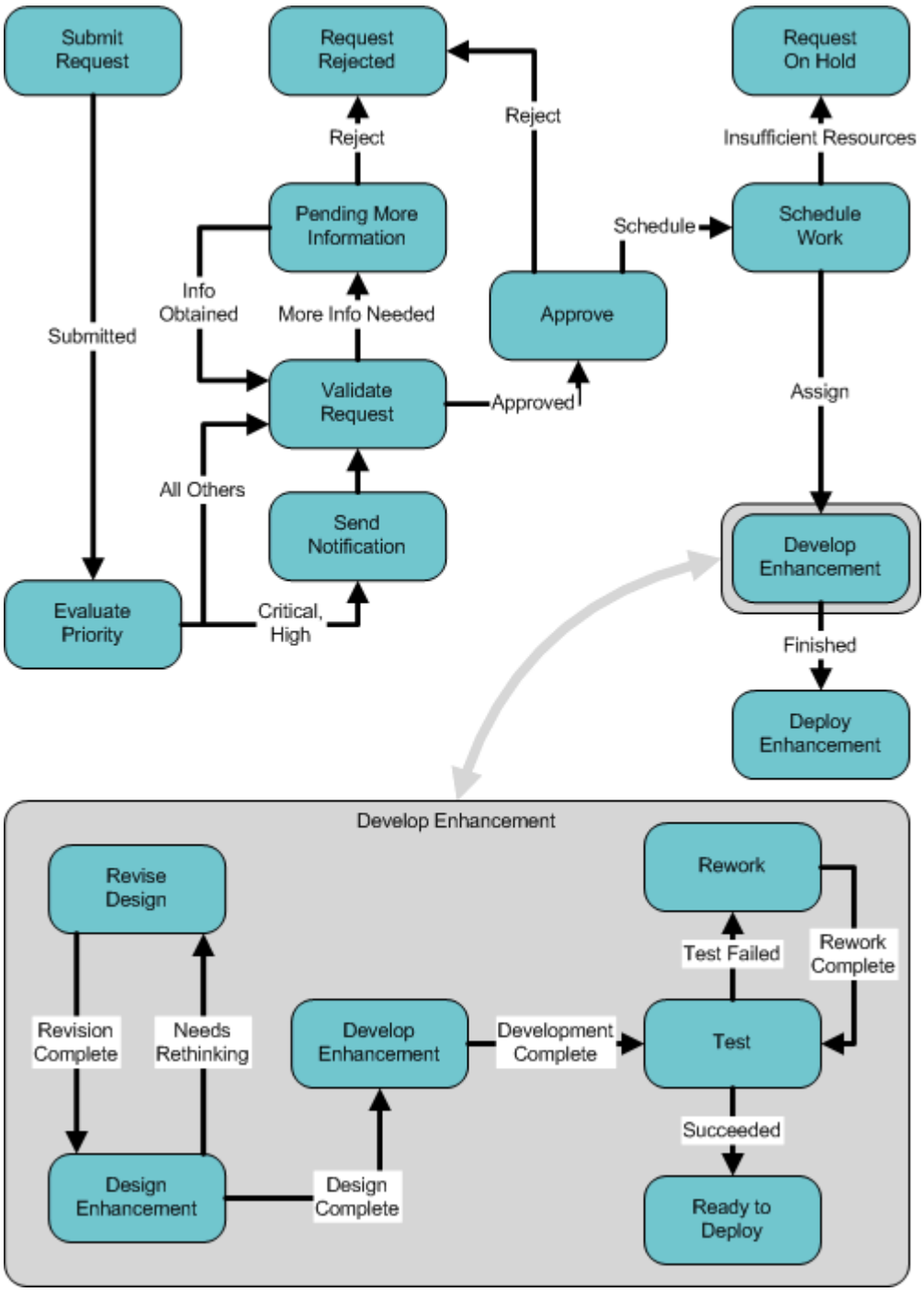


Figure 2-2. Business process with subworkflow

Considering Request Statuses

A request status can be associated with a workflow step. The request's status at a particular workflow step will drive field logic during the request's lifecycle. Build the request type before associating its statuses with workflow steps.

Gathering Requirements for Request Types

Many different types of requests can be sent through a workflow. As a request moves through its resolution process, its fields and status can change.

Request Type Fields

Each request requires different information to process it. For example, to resolve a software bug, you need to know the software unit, product version, problem and priority. This information is captured using request type fields.

For each field in the request, collect the following information:

- **Field name.** The field's prompt should help ensure that the correct information is captured.
- **Information type.** What type of information needs to be collected? Should it be a text field? Will users pick from a predetermined list of values? The field information type is governed by its validation, which defines the field's component type as well as what information can be entered into the field. For example, a field using a numeric text field validation will accept only numeric values.
- **Field behavior.** There are many aspects of a field that can be controlled:
  - The field can be configured to become non-editable or required depending on the value of other fields, or at a certain workflow step.
  - The field can also be configured to automatically populate itself based on values in other fields.
  - The field can also be configured to be non-editable or invisible based on which user is looking at the request.
- **Field Statuses**

Use the worksheets in *Worksheets* on page 315 to collect the request type field specifications.

Example: Information Collected for the Software Change Request

The IT group needs to know the following information in order to properly resolve a request for a software change to the financial system:

- The name of the user creating the request
- Whether the request is for new or enhanced functionality
- The module to be enhanced (such as Billing, Inventory, or Payroll)
- The priority of the request
- A description of the new or enhanced functionality
- Any supporting documents (such as detailed proposals or web sites)

As the request progresses along the business process, other information becomes necessary:

- Whether the IT group has the budget to develop the change
- The estimated time to completion for the change
- The name of the developer assigned to build the change

To describe the request type, the IT group decided they needed to define the following fields:

- Created By: The user who created the request.
- New or Enhanced: Whether the change being requested is a new piece of functionality or an enhancement to an existing module.
- Priority: The priority of the request. **Critical** requests are acted on much faster than requests with **Low** priority.
- Impacted Module: The software module to be changed.
- Description: A brief description of the change being requested.
- Supporting Documents: A place for the requestor to attach any supporting documents to the request. These documents might be URLs or more detailed proposals in Rich Text Format.

See *Configuring Request Types and Request Header Types* on page 155 for additional examples on request type fields.

Request and Workflow Interaction

The list of possible statuses the request can take on as it moves through its resolution process can be configured. Each request status can control 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 its fields' attributes.

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 tends to suggest a one-to-one mapping.

It is also possible to restrict which workflows and request types can be used together. Determine what, if any, restrictions are to be put in place at this level.

Request Header Types

Request header types define a standard collection of fields that appear in the header or any other region of a request using that request type. Each request type must have an associated request header type. Request header types contain a standard set of fields that can be enabled or disabled. It is also possible to add custom fields to the request header type.

Request Type Commands

Commands can be contained in a request type that allow it to perform command-line executions. Request type commands often reference information stored in its fields. These commands are executed at specific points (execution steps) in the workflow.

Collect the following information for each request type command that will be designed:

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

Use the worksheet in *Worksheets* on page 315 for assistance in collecting the correct data.

See *Commands, Tokens, and Validations Guide and Reference* for additional information on building commands.

Identifying Participants and Security

A great deal of control can be implemented over a request resolution system. Restrict user actions around:

- Request creation:
  - Who can create requests.
  - Who can use a specific workflow.
  - Who can use specific request types.
- Request processing:
  - Who can approve/process each step in the workflow.
  - Who can view and edit fields in the request.
  - Who can delete a request.
- Configuring your request resolution process:
  - Who can edit the workflow.
  - Who can edit each request type.

Configuring this data and process security often involves a setting a number parameters: licenses, access grants, entity level settings and field level settings.

For the request resolution process, collect information that will help to identify users, group them into security groups, and restrict access to certain functionality.

Use security groups or dynamic access (tokens) whenever possible. Avoid specifying a list of users to control an action; for example, specifying a list of users who can act on a workflow step. If the list of users changes (due to a departmental reorganization), that list would have to be updated manually in many places on the workflow. 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.

Example: The IT Group Determines Participants and Security

The process of approving changes to the financial system application involves many groups and individuals within the company.

- The Finance Group, users of the financial system application
- Director of Finance
- Finance Business Analyst
- Director of IT
- IT Personnel Manager
- IT Development: Engineers
- IT Lead Engineer
- Manager of Release Team
- IT Configuration Manager

Within this group of users, there are some logical divisions of labor. Using this division, the IT group constructs the following security groups.

Table 2-1. Security groups

Security Group	Members	Responsibilities
Financial Apps - Create and View Requests	• Finance Group • Director of Finance	Responsible for creating requests. The users can create requests at any time and view the status of requests they are involved in.
Financial Apps - Manage Resolution System	• IT Configuration Manager	Responsible for request tracking and resolution system. The user has the ability to modify the request resolution process (workflow, request types, and security groups). The user can also act on any step in the process.
Financial Apps - Validate and Approve Requests	• Director of Finance • Finance Business Analyst • Director of IT	Responsible for evaluating and approving incoming requests. Can reject or approve requests for development.
Financial Apps - Schedule Requests	• Director of IT • IT Personnel Manager	Responsible for approving requests for development, scheduling and assigning work, or putting requests on hold until sufficient resources are available.

Table 2-1. Security groups [continued]

Security Group	Members	Responsibilities
Financial Apps - Develop Requests	• IT Development: Engineers • IT Lead Engineer • Manager of Release Team	Responsible for developing enhancements specified in requests, including functional design, implementation, and QA.
Financial Apps - Deploy Changes	• Manager of Release Team	Responsible for overseeing deployments to the Financial group.

Using these security groups and user definitions, IT collects specific information related to their request resolution process. This information will be considered later when defining security groups and workflows.

Table 2-2. Request creation security

Action	Users allowed to perform action	Controlled by: (Users, Security Group, Token)
Create a Request	• The Financial Group • Director of Finance	Financial Apps - Create and View Requests, Financial Apps - Manage Resolution System
Use the Financial System Change Workflow	• Everyone	Financial Apps - Create and View Requests, Financial Apps - Manage Resolution System, Financial Apps - Validate and Approve Requests, Financial Apps - Schedule Requests, Financial Apps - Develop Requests, Financial Apps - Deploy Changes
Use the Financial System Change Request Type	• Everyone	Financial Apps - Create and View Requests, Financial Apps - Manage Resolution System, Financial Apps - Validate and Approve Requests, Financial Apps - Schedule Requests, Financial Apps - Develop Requests, Financial Apps - Deploy Changes

Notice that the IT Configuration Manager was added to each action by adding the **Financial Apps - Manage Resolution System** security group to each step. This provides a single, relevant user with override privileges to keep the process moving.

IT also indicates how they would like to control which users can act on each step. They select to exclusively use security groups and tokens. Notice that multiple criteria can be specified to enable access to a single step: for example, you could specify two security groups and a TOKEN [REQ.CREATED_BY] to enable access. Users who meet any of the requirements (members of at least one security group or the value of the token) can act on the step.

Table 2-3 specifies which users can act on a specific step in the workflow. See *Figure 2-3* on page 39 to see the process referenced in this table.

Table 2-3. Request processing security - financial system change workflow

Workflow Step Name	Users allowed to act on	Controlled by: (Users, Security Group, Token)
Validate Request	• Director of Finance • Finance Business Analyst • Director of IT	Financial Apps - Validate and Approve Requests Financial Apps - Manage Resolution System
Pending More Information	• Financial group member who created the request • Director of Finance	TOKEN (REQ. CREATED_BY); Financial Apps - Create and View Requests (Security Group) Financial Apps - Manage Resolution System
Approve	• Director of Finance • Finance Business Analyst • Director of IT	Financial Apps - Validate and Approve Requests Financial Apps - Manage Resolution System
Schedule Work	• Director of IT • IT Personnel Manager	Financial Apps - Schedule Requests Financial Apps - Manage Resolution System
Develop Enhancement	• IT Development Engineers • IT Lead Engineer • Manager of Release Team	Financial Apps - Develop Requests Financial Apps - Manage Resolution System
Deploy Enhancement	• Manager of Release Team	Financial Apps - Deploy Changes Financial Apps - Manage Resolution System

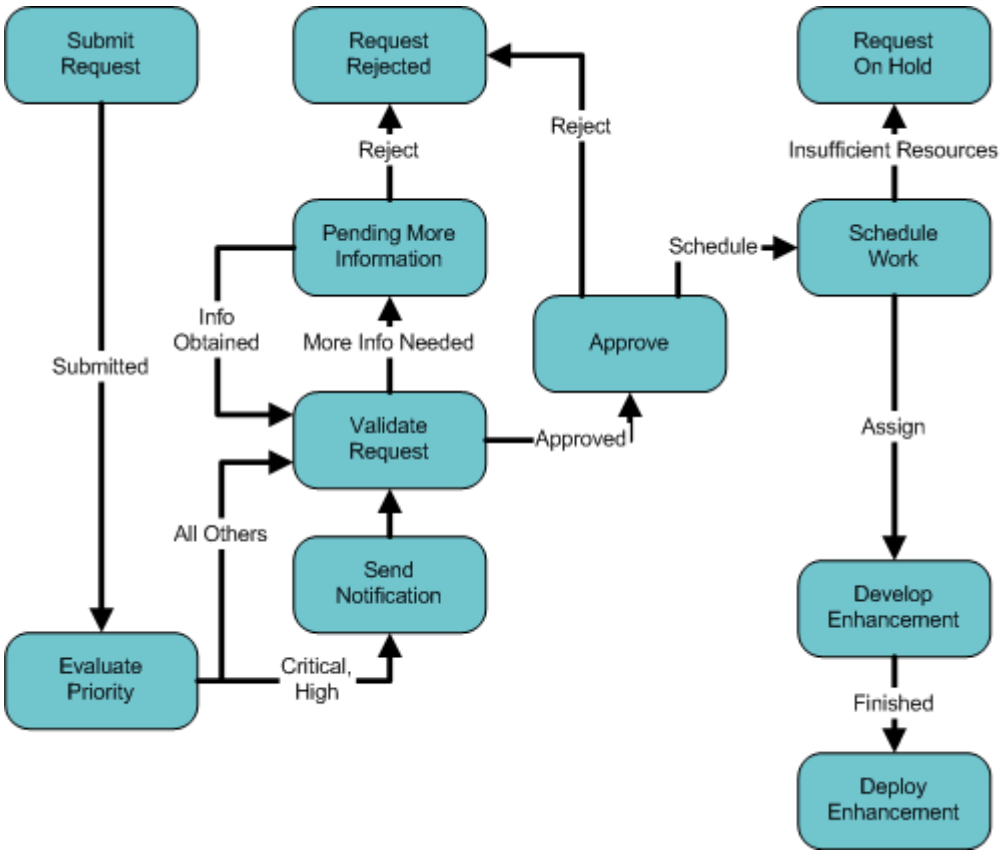


Figure 2-3. Business process

IT must also specify who can modify the existing process. This level of security is configured using ownership settings and security group access grants. See *Security Module Guide and Reference* for more information on these topics.

Table 2-4. Security around managing the financial system change process

Action	Users allowed to perform action	Controlled by: (Users, Security Group, Token)
Modify the Workflow	• IT Configuration Manager	Financial Apps - Manage Resolution System
Modify the Financial System Change Request Type	• IT Configuration Manager	Financial Apps - Manage Resolution System

Establishing Communication Points and Visibility

Determine the communication points and methods for providing visibility into the process and request statuses. This section lists the information that needs to be gathered to define notifications. For more information on defining and using portlets and reports, refer to the following documents:

- *Getting Started*
- *Reports Guide and Reference*

Notifications on Workflow Steps

It is possible to send a notification when a workflow step becomes eligible, has a specific outcome, or has a specific error. For each workflow step in the process, collect the following information:

Table 2-5. Information to gather for workflow steps

Workflow step name	Include notification for step? (Yes/No)
Step 1 - Name	Yes
Step 2 - Name	No
Step 3 - Name	No

For each step that requires a notification, gather the following information:

Table 2-6. Information to gather for workflow step notifications

Parameter	Description
Workflow Step Name	The name of the step that requires a workflow.
Notification Event (All, Eligible, Specific Result, Specific Error)	Specifies the event that triggers the notification. the possible values are **All, Eligible, Specific Result,** or **Specific Error.**
Value (for Specific Result)	Specifies that a notification is sent for the selected result of the workflow step.
Error (for Specific Error)	Specifies that a notification is sent for the selected error of the workflow step.

Table 2-6. Information to gather for workflow step notifications [continued]

Parameter	Description
Recipient	Determine who should receive the message. you can choose to send the notification to users based on: • **Username** • **Email Address** • **Security Group** • **Standard Token** • **User Defined Token**
Message	Determine what the message will say. Also determine if it will contain a link to the request.

Example: IT Configures Notifications

IT determines that a notification needs to be added to the following steps:

Table 2-7. Workflow steps with notifications

Workflow step name	When to send notification	Recipients
Send Notification	[REQ.PRIORITY] = 'High', 'Critical'	Financial Apps - Validate and Approve Requests Financial Apps - Schedule Requests
Pending More Information	Eligible	TOKEN [REQ.CREATED_BY]
Schedule Work	Eligible	Financial Apps - Schedule Requests
Develop Enhancement	Eligible	Financial Apps - Develop Enhancement
Deploy Enhancement	Eligible	TOKEN [REQ.CREATED_BY] Financial Apps - Deploy Enhancement

Notifications on Field Changes

Notifications can be sent when a field in a request changes value. For more detailed information on setting up these notifications, see *Configuring Request Types and Request Header Types* on page 155.

Chapter

3

Configuring Workflows

In This Chapter:

Overview of Workflows

This chapter covers information concerning Demand Management workflows, Change Management workflows, and Release Management workflows.

A workflow represents a business process and is used to map business rules and processes to your organization.

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. For 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 results of a decision step is **Approved** and **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 a Approve Request decision step, only the IT project manager can **Approve** or **Not Approved** the request.

- **Notification.** Notifications are emails alerts sent out at specific workflow steps. For example, for a 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.

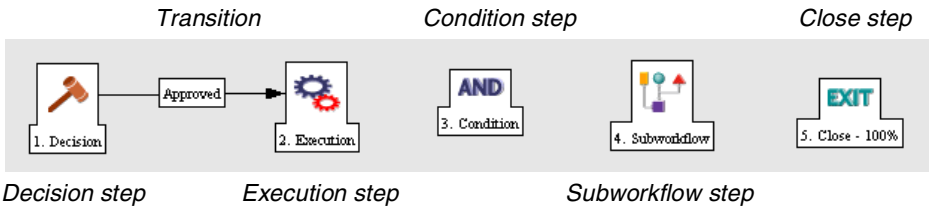


Figure 3-1. Workflow components

Mapping Workflows

Mapping all of the individual workflow steps into a single workflow is a two-step process:

Step 1. Create a block diagram. Map each Workflow Step Worksheet as a one block in the diagram. On the block diagram include transitions, workflow step security, and notifications.

Step 2. Map the block diagram to the workflow. Open the Workflow Workbench window and start a new workflow. Map each component from the block diagram to the new workflow.

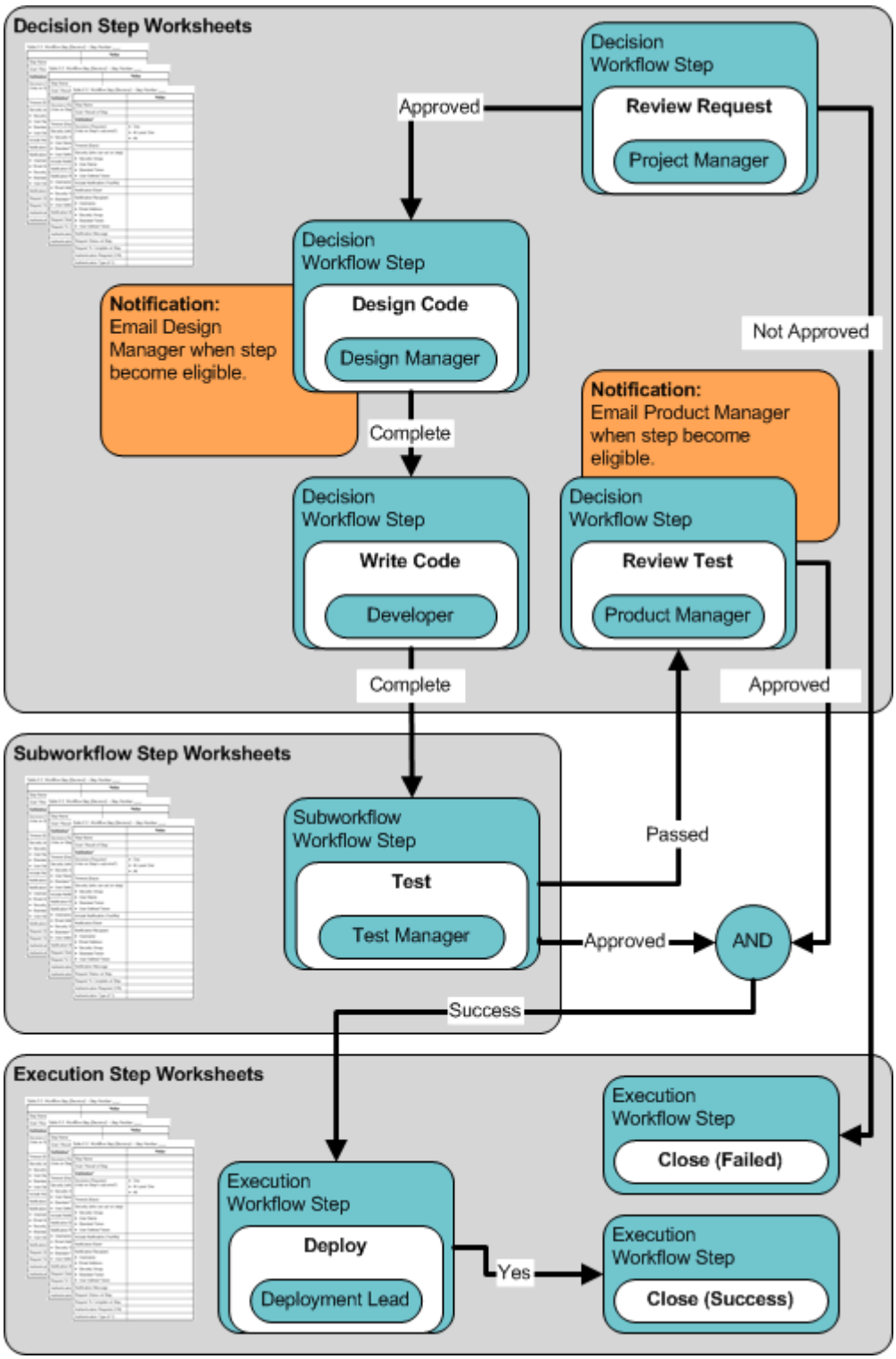


Figure 3-2. Step 1. Create a block diagram

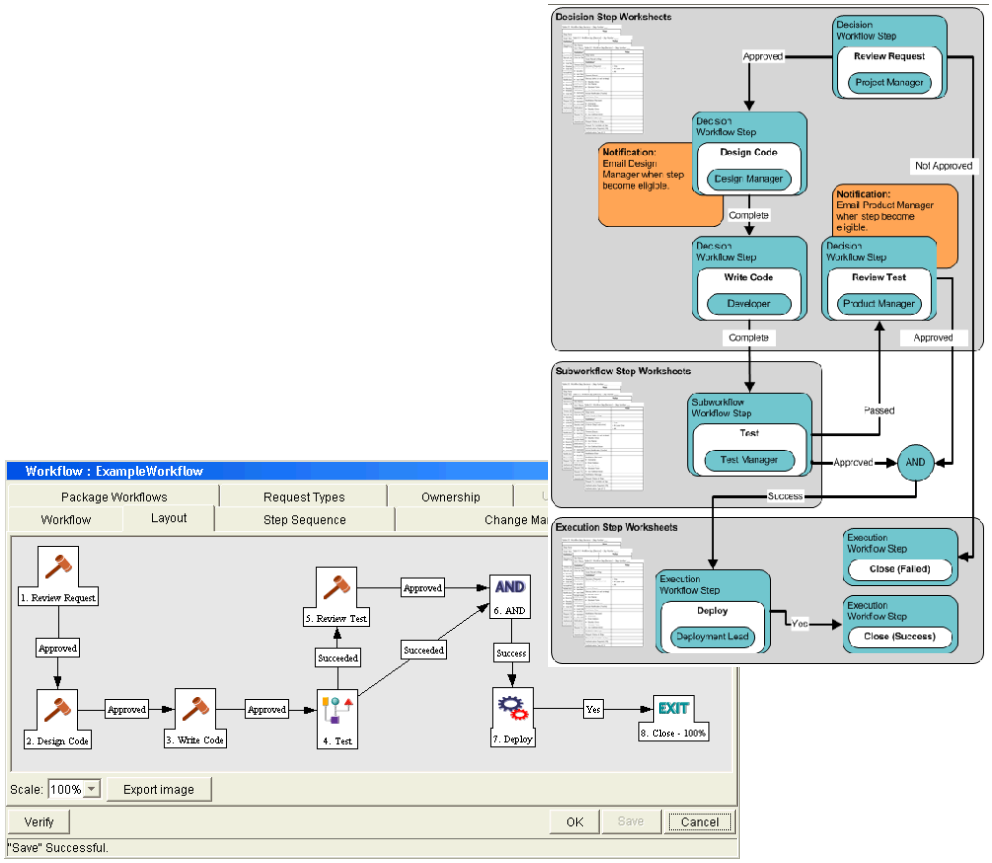


Figure 3-3. Step 2. Create the workflow

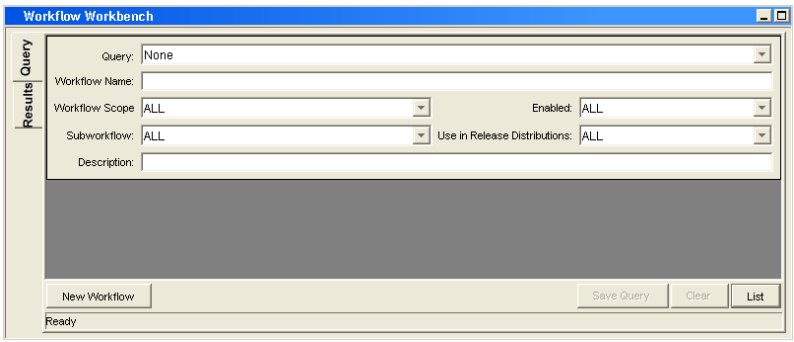
Opening the Workflow Workbench

To open the Workflow Workbench:

1. Log on to the Mercury IT Governance Center.
2. From the menu bar, select **Administration > Open Workbench.**
3. A Workbench status window opens. A few minutes later, a Warning - Security window opens.
4. In the Warning - Security window, select **Yes.**

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

   The Workflow Workbench window opens.



For More Information

For information on how to search and select an existing workflow, copy a workflow, and delete a workflow, see *Getting Started*.

Creating Workflows

Starting a new workflow requires knowing how to use the Workflow Workbench. This section covers the basics on how to create a workflow.

Configuring General Information for Workflows

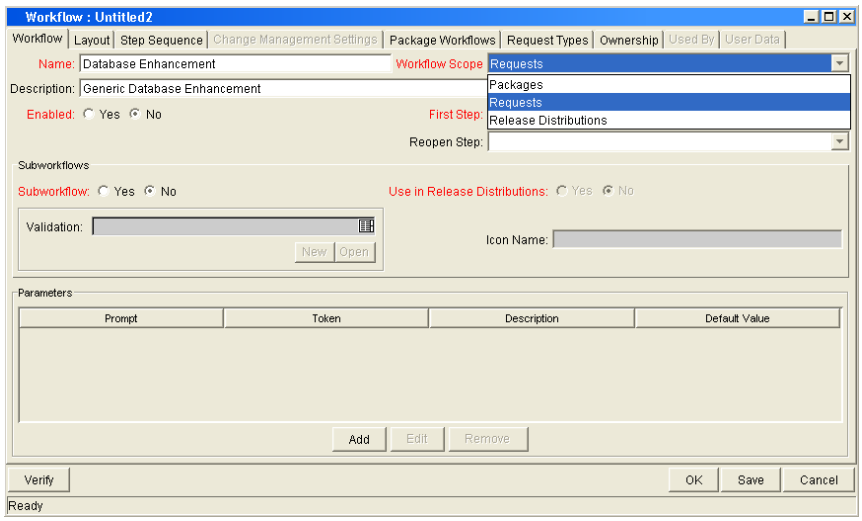
To enter basic workflow information:

1. Open the Workflow Workbench.

   To open the Workflow Workbench, see *Opening the Workflow Workbench* on page 49. The Workflow Workbench window opens.

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

   The Workflow window opens.



3. In Name, enter the name of the new workflow.

4. In Workflow Scope, select one of the following from the drop-down list:

   - For Mercury Change Management packages, select **Packages.**
   - For Mercury Demand Management requests, select **Requests.**
   - For Mercury Change Management releases and distributions, select **Release Distributions.**

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

   Click **OK** to save the changes and close the Workflow window. Click **Save** to save the changes and leave the Workflow window open. Click **Cancel** to drop the changes and close the Workflow window.

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, allowing you to see only the workflow steps available for you to use.

Workflow steps are assembled into workflows in the **Layout** tab of the Workflow window. Select a workflow step from the Workflow Step Sources window and drag and drop the workflow step onto the **Layout** tab. As part of the drag and drop process, a Workflow Step window opens. The Workflow Step window is used to configure the following:

- General information concerning the workflow step
- Security for the workflow step
- Notifications for the workflow step
- Timeouts for the workflow step

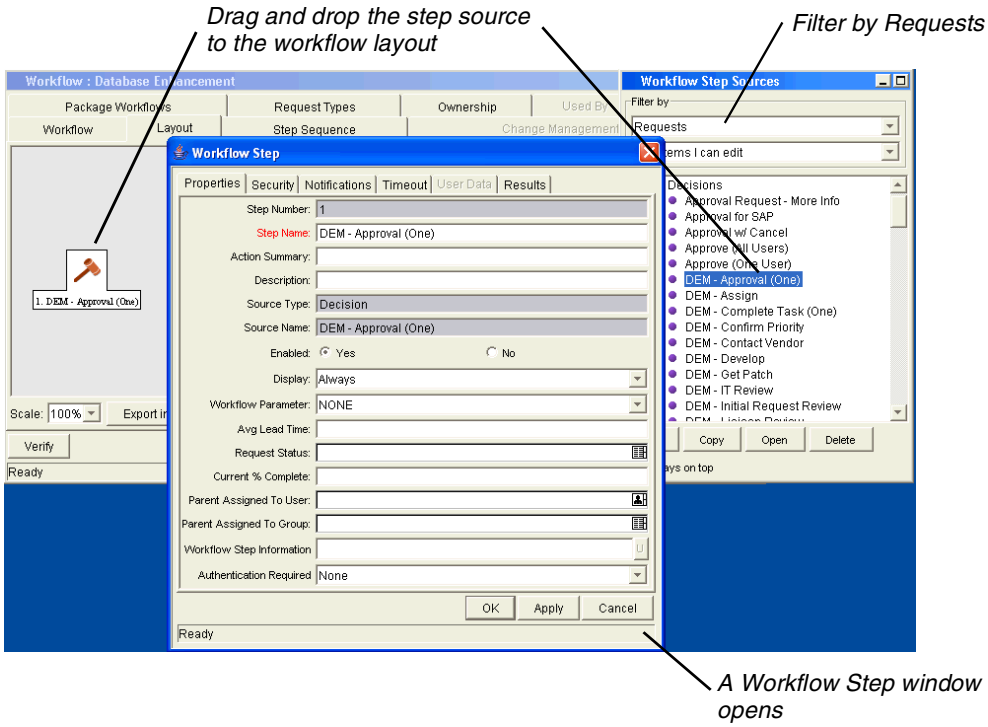


Figure 3-4. Drag and drop

Choosing Workflow Steps

Mercury IT Governance Center comes with many pre-defined 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. Select an entry from the Filter by drop-down list filter the workflow steps. The following lists the folders found in the Workflow Step Source window:

- Decision
- Conditions
- Executions
- Subworkflows

To evaluate a workflow step, determine which of the four workflow folders is required for your workflow step. Open the Workflow Step Source folder and open those workflow steps that seem to best meet your needs (see *Figure 3-5* on page 53).

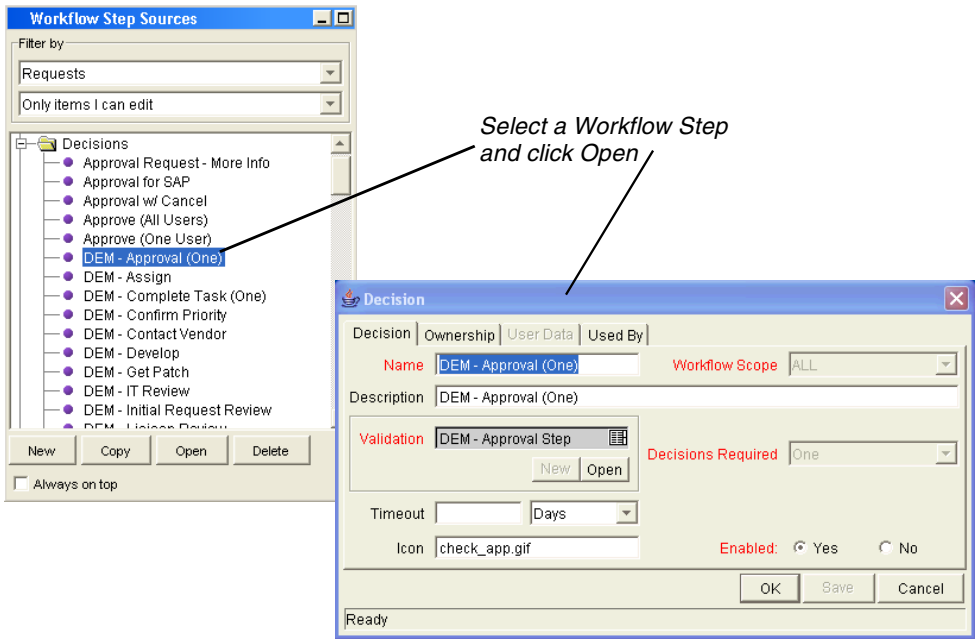


Figure 3-5. Workflow step source

Check the validation to see if the validation values meet your transition requirements. The validation values are the acceptable values a workflow step can have (see *Configuring Validations for Workflow Steps* on page 101).

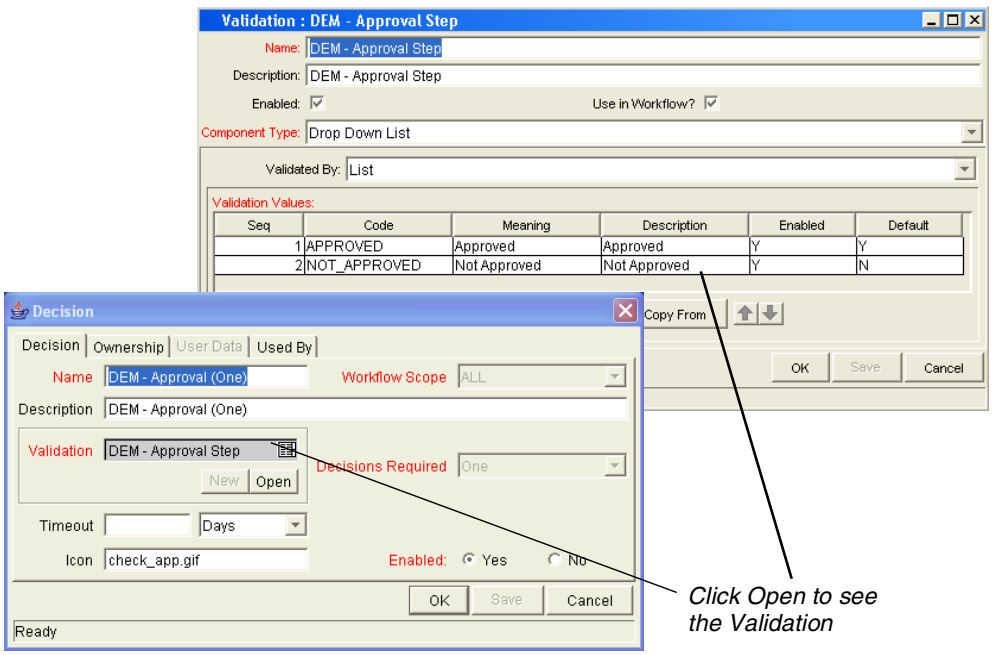


Figure 3-6. Workflow step source validation

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 when each of the workflow steps are completed. The following is a list of the conditions workflow steps.

- **AND**. An AND condition is satisfied only if all workflow steps leading to it reach the status they are supposed to attain.

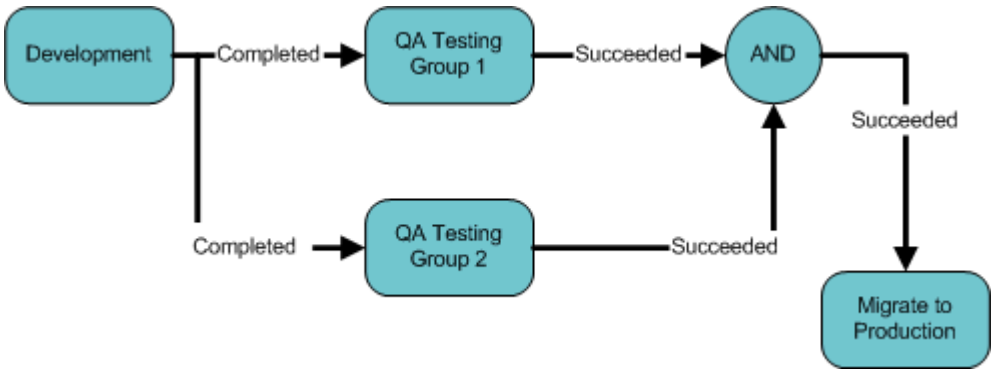


Figure 3-7. AND example

- **OR.** An OR condition is successful when at least one of the workflow steps leading to it reaches the status it is supposed to attain.

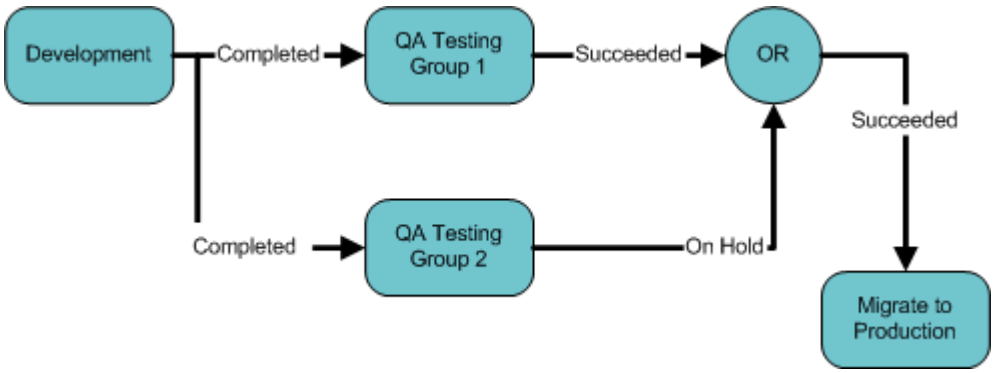


Figure 3-8. OR example

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 represent multiple workflow steps (the subworkflow) in a workflow. When the workflow process reaches the subworkflow step, it follows the path defined in that subworkflow. Subworkflows can either close within that workflow or return to the parent workflow.

Adding Close Workflow Steps

Every workflow, no matter how long or short, must include a close workflow step (see *Figure 3-9*). A close workflow step is a specific kind of execution workflow step and can be found in the Executions folder of the Workflow Step Sources window.

There are three close workflow steps:

- **Close (Immediate Success).** Close (Immediate Success) immediately completes a request or package with a status of **Success**.
- **Close (Manual Success).** Close (Manual Success) requires manual intervention to complete a request or package. The status of the request or package is set to **Success**.
- **Close (Immediate Failure).** Close (Immediate Failure) immediately completes a request or package with a status of **Failure.**

Add a close workflow step to a workflow as you would any other workflow step. See *Figure 3-9*.

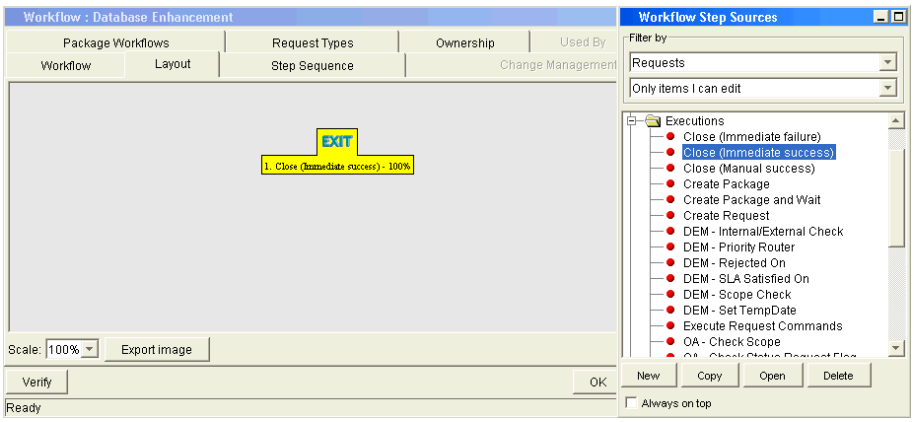


Figure 3-9. Close workflow step

Configuring Reopen Workflow Steps

Closed requests can be reopened by users with the proper access grants. A reopened request begins at a workflow step specified as the reopen workflow step for the workflow.

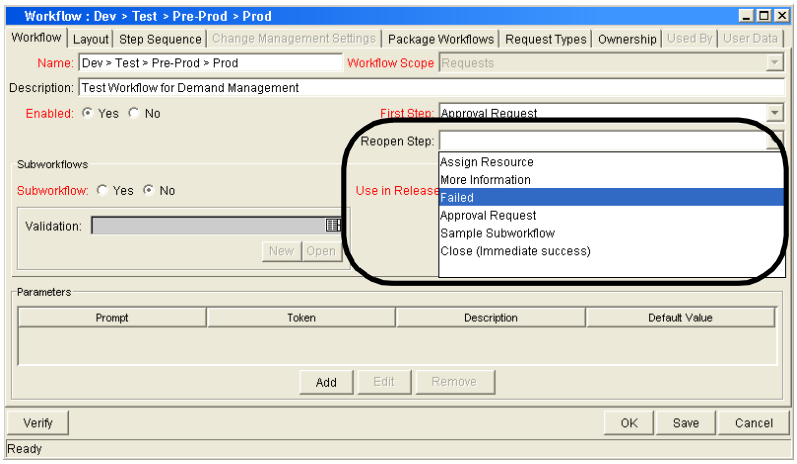


Figure 3-10. Workflow window reopen step drop-down list

To specify a reopen workflow step for a workflow, open the **Workflow** tab in the Workflow Workbench window (see *Figure 3-12* on page 58). Open the drop-down list from the Reopen Step field and select the reopen workflow step.

Adjusting Workflow Step Sequences

Once all of the workflow steps are assembled in the **Layout** tab, you can adjust the sequence of the steps. In the Workflow window, select the **Step Sequence** tab. The **Step Sequence** tab lists all of the workflow steps. Select a workflow step and click the **Arrow** icons at the bottom of the tab to move the selected workflow step up or down.

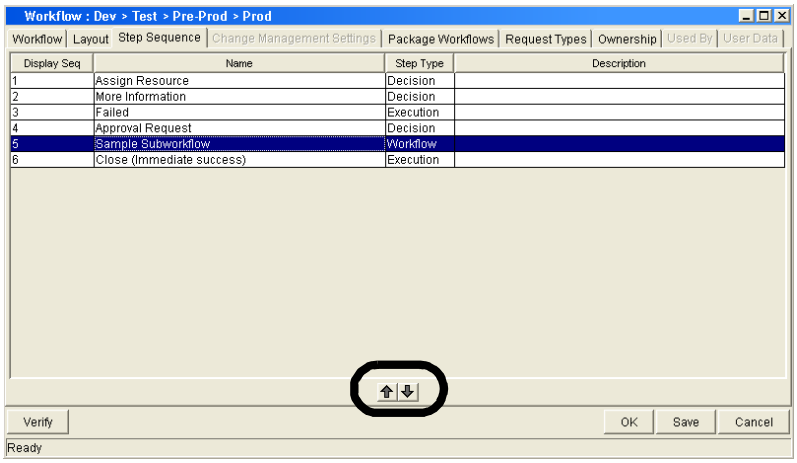


Figure 3-11. Step sequence tab

Specifying the First Step

Once all of the workflow steps are assembled and properly sequenced, you must specify the first step in the workflow process. To specify the first step, open the **Workflow** tab in the Workflow Workbench window (see *Figure 3-12*). Open the drop-down list in the First Step field and select the first step.

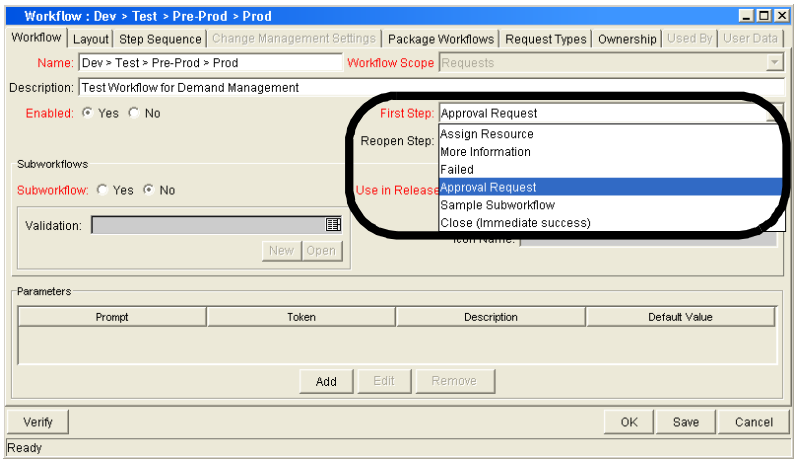


Figure 3-12. Workflow tab

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. Open the Workflow Workbench.

   To open the Workflow Workbench, see *Opening the Workflow Workbench* on page 49. The Workflow Workbench window opens.

2. Open a workflow.

   The Workflow window opens. The **Workflow** tab is displayed.

3. At the bottom left-hand corner of the **Workflow** tab, click **Verify.**

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

To enable a workflow:

1. Open the Workflow Workbench.

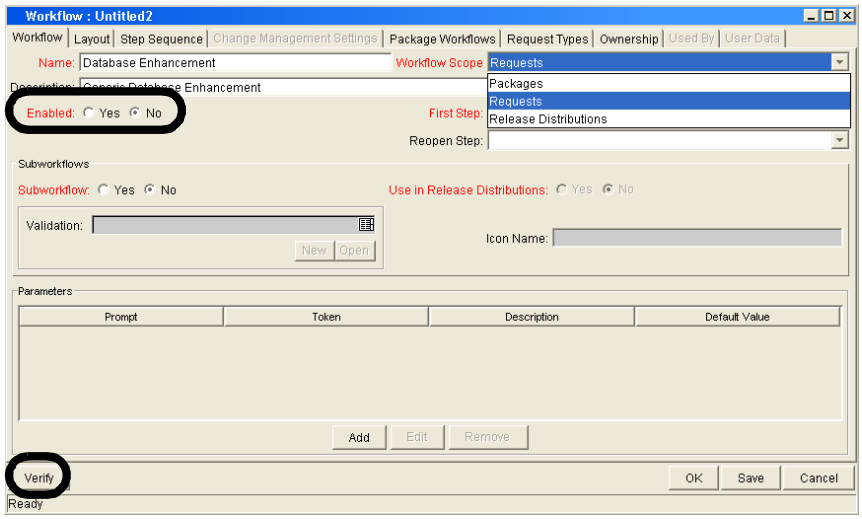
   To open the Workflow Workbench, see *Opening the Workflow Workbench* on page 49. The Workflow Workbench window opens.

2. Open a workflow.

   The Workflow window opens. The **Workflow** tab is displayed.

3. In the **Workflow** tab, in the Enable field, select **Yes.**

   The workflow is enabled.



4. In the **Workflow** tab, click **Save.**

   The changes to the workflow are saved.

Configuring Workflow Steps

Every time a workflow step is dragged and dropped from the Workflow Step Source window to the **Layout** tab of 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 entered in the Workflow Step window can be gathered from the appropriate Workflow Step Worksheets. Each Workflow Step window includes the following tabs:

- **Properties.** General information concerning the workflow step is defined under the **Properties** tab.
- **Security.** Permission settings for specific individuals or groups authorized to act on a workflow step are defined under the **Security** tab.
- **Notifications.** Emails 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. Notifications are defined under the **Notifications** tab.
- **Timeout.** Timeouts determine how long a workflow step can remain inactive before generating an error. Timeouts are defined under the **Timeout** tab.
- **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.** Each workflow step includes a validation. The **Results** tab lists the validation, the component type and the results.

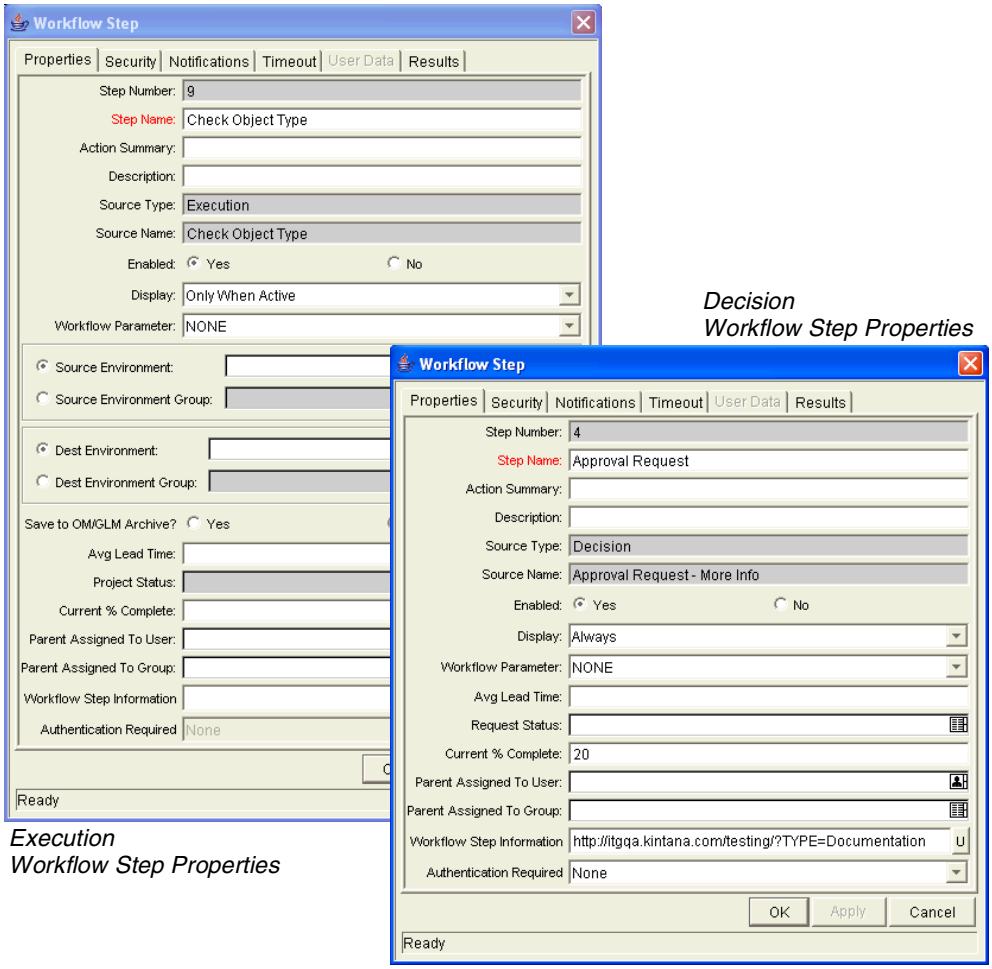


Figure 3-13. Workflow step properties

Configuring General Information for Workflow Steps

General information concerning the workflow step is defined in the Workflow Step window under the **Properties** tab.

To add general information to a workflow step:

1. Open the Workflow Workbench.

   To open the Workflow Workbench, see *Opening the Workflow Workbench* on page 49. The Workflow Workbench window opens.

2. Open a workflow.

   The Workflow window opens.

3. Open the **Layout** tab of the Workflow window.

4. Right-click a workflow step.

   The workflow step is highlighted. A menu window opens.

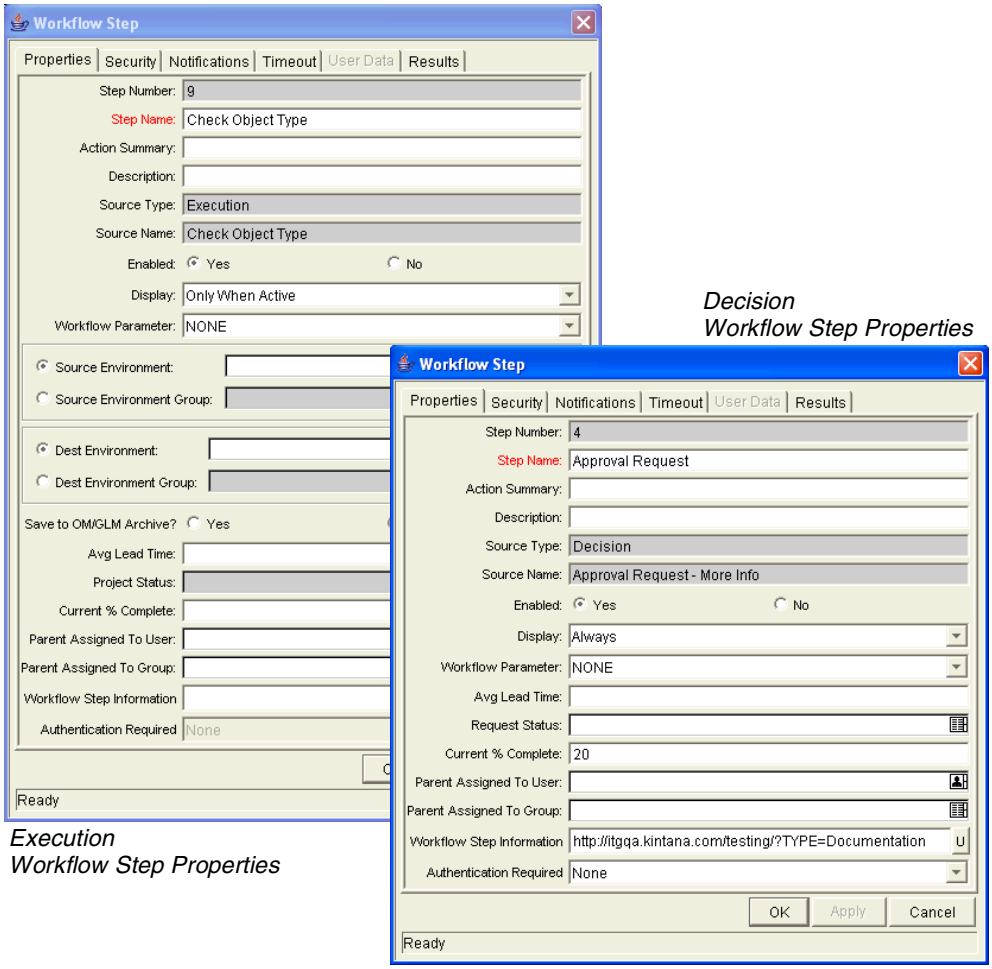
5. In the menu window, select **Edit.**

   The Workflow Step window opens.

6. Make sure you are in the **Properties** tab.

   The **Properties** tab is the default tab for the Workflow Step window.

7. Complete the fields in the **Properties** tab.



Execution
Workflow Step Properties

Decision
Workflow Step Properties

8. In the **Properties** tab of the Workflow Step window, click **Save.**

   The changes to the workflow are saved.

Configuring Security for Workflow Steps

Workflow steps need to have permission settings to define the specific individuals or groups who are authorized to act on each workflow step.

To add workflow step security to a workflow step:

1. Open the Workflow Workbench.

   To open the Workflow Workbench, see *Opening the Workflow Workbench* on page 49. The Workflow Workbench window opens.

2. Open a workflow.

   The Workflow window opens.

3. Open the **Layout** tab of the Workflow window.

4. Right-click a workflow step.

   The workflow step is highlighted. A menu window opens.

5. In the menu window, select **Edit.**

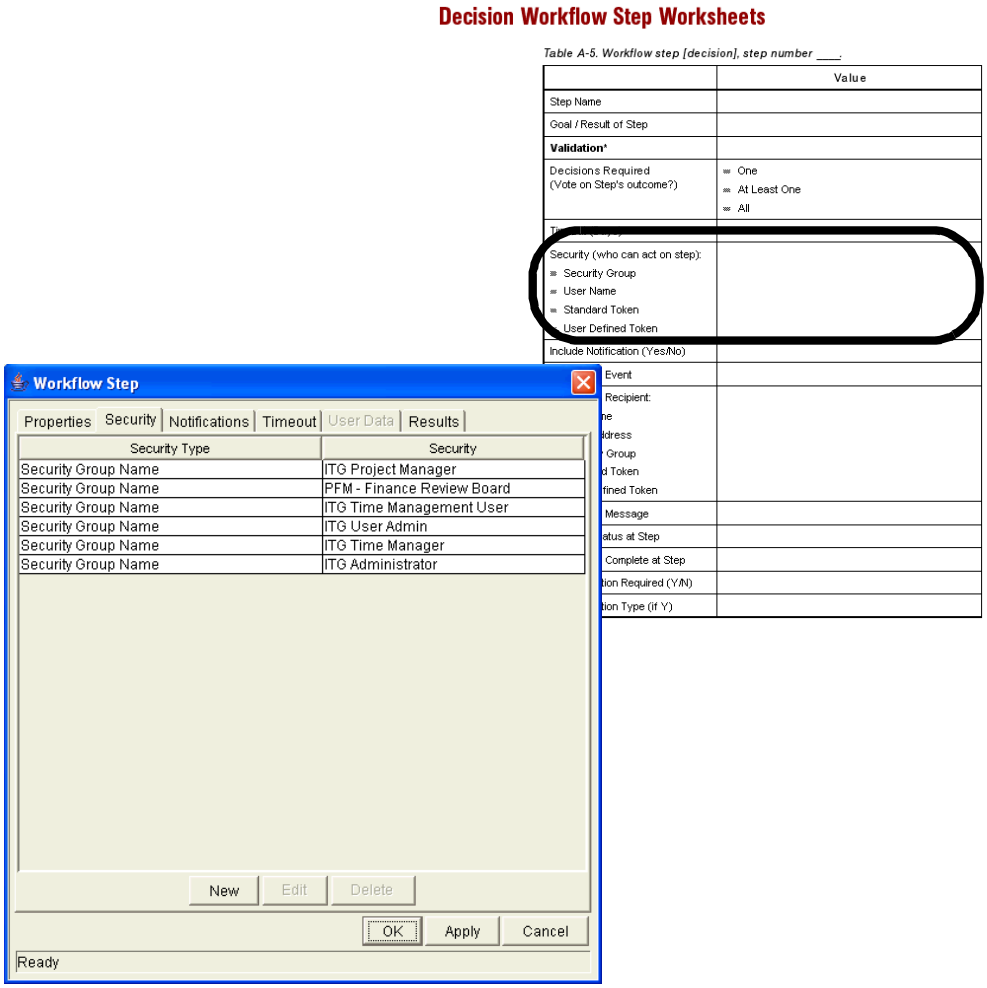
   The Workflow Step window opens.

6. In the Workflow Step window, select the **Security** tab.

   The **Security** tab opens.

7. In the **Security** tab, click **New.**

   The Workflow Step Security window opens.



8. In the Workflow Step Security window, select the security type from the drop-down list.

   The security type options are:

   - **Enter a Security Group Name.** Select a security group to act upon the workflow step. Selecting a security group changes the name of the autocomplete 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 autocomplete field 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 autocomplete field to Standard Token. The security type is left undefined. Select a standard token from the autocomplete field. 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 autocomplete field to User Defined Token. The security type is dynamically changed to a drop-down 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.**

   The Workflow Step Security window closes.

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

    The Workflow Step window closes.

11. In the **Security** tab of the Workflow Step window, 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.

   To open the Workflow Workbench, see *Opening the Workflow Workbench* on page 49. The Workflow Workbench window opens.

2. Open a workflow.

   The Workflow window opens.

3. Open the **Layout** tab of the Workflow window.

4. Right-click a workflow step.

   The workflow step is highlighted. A menu window opens.

5. In the menu window, select **Edit.**

   The Workflow Step window opens.

6. In the Workflow Step window, select **Security** tab.

   The **Security** tab opens.

7. In the **Security** tab, click **New.**

   The Workflow Step Security window opens.

8. In the Workflow Step Security window, select the security type from the drop-down list.

   The security type options are:

   - **Enter a Security Group name.** Select a security group to act upon the workflow step. Selecting a security group changes the name of the autocomplete 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 autocomplete field 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 autocomplete field to Standard Token. The security type is left undefined. Select a standard token from the autocomplete field. 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 autocomplete field to User Defined Token. The security type is dynamically changed to a drop-down 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.**

   The Workflow Step Security window closes.

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

    The Workflow Step window closes.

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

    The changes are added to the workflow.

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 in the **Notifications** tab of the Workflow Step window.

When configuring a notification for a workflow step, consider the following:

- When to send the notification
- Who should receive the notification
- What the notification should say

Review the Workflow Step Worksheet for notification information.

To add a notification to a workflow step:

1. Open the Workflow Workbench.

   To open the Workflow Workbench, see *Opening the Workflow Workbench* on page 49. The Workflow Workbench window opens.

2. Open a workflow.

   The Workflow window opens.

3. Open the **Layout** tab of the Workflow window.

4. Right-click a workflow step.

   The workflow step is highlighted. A menu window opens.

5. In the menu window, select **Edit.**

   The Workflow Step window opens.

6. In the Workflow Step window, select the **Notifications** tab.

   The **Notifications** tab opens.

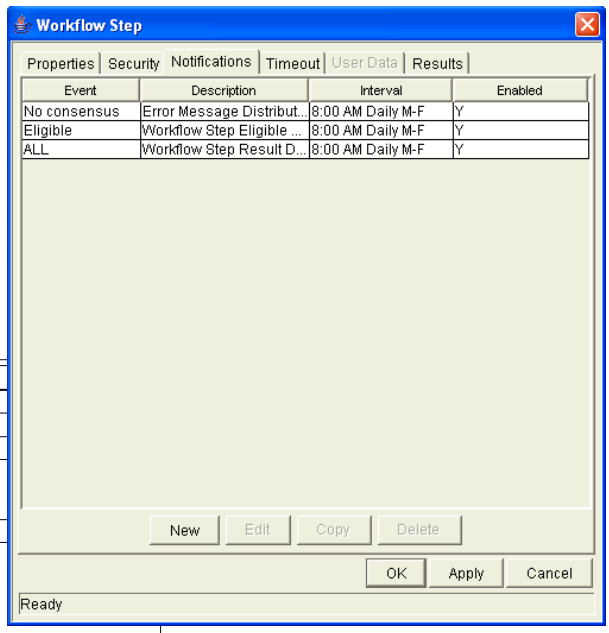
7. In the **Notifications** tab, click **New.**

   The Add Notification for Step window opens. The Add Notification for Step window has two tabs: **Setup** and **Message.**

8. In the **Setup** tab of the Add Notification for Step window, configure:

   - When the notification should be sent (Event and Interval)
   - Who receives the notification (Recipients)

9. In the **Message** tab of the Add Notification for Step window, configure the body of the notification.



Decision Workflow Step Worksheets

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

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)	

10. In the Add Notification for 1Step window, click **OK.**

    The notification is added to the **Notifications** tab. You can send different notifications to different recipients for different events by clicking **New** and repeating the previous 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

11. In the **Notifications** tab of the Workflow Step window, 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, open the **Notifications** tab.

   See *Configuring Notifications for Workflow Steps* on page 70. The **Notifications** tab opens.

2. In the **Notifications** tab, click **New.**

   The Add Notification for Step window opens. The Add Notification for Step window has two tabs: **Setup** and **Message.**

3. In the Add Notification for Step window, select the **Setup** tab.

   The **Setup** tab is the default tab.

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

Field	Value	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. In the **Setup** tab, click **OK.**

   The **Setup** tab closes. The Workflow Step window opens.

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

   The changes are added to the workflow.

Sending Notifications when Workflow Steps have Specific Results

It is possible to configure a notification to be sent 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, open the **Notifications** tab.

   See *Configuring Notifications for Workflow Steps* on page 70. The **Notifications** tab opens.

2. In the **Notifications** tab, click **New.**

   The Add Notification for Step window opens. The Add Notification for Step window has two tabs: **Setup** and **Message.**

3. In the Add Notification for Step window, select the **Setup** tab.

   The **Setup** tab is the default tab.

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

Field	Value	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. In the **Setup** tab, click **OK.**

   The **Setup** tab closes. The Workflow Step window opens.

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

   The changes are added to the workflow.

Sending Notifications When Workflow Steps Have Specific Errors

It is possible to 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 70. The **Notifications** tab opens.

2. In the **Notifications** tab, click **New.**

   The Add Notification for Step window opens. The Add Notification for Step window has two tabs: **Setup** and **Message.**

3. In the Add Notification for Step window, select the **Setup** tab.

   The **Setup** tab is the default tab.

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

Field	Value	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 75.
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. In the **Setup** tab, click **OK.**

   The **Setup** tab closes. The Workflow Step window opens.

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

   The changes are added to the workflow.

Specifying the Time Notifications are Sent

Use the Interval field in the workflow step to specify when the notification will be sent. The interval determines how frequently the notification will be sent.

To send the time notification are sent:

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

   See *Configuring Notifications for Workflow Steps* on page 70. The **Notifications** tab opens.

2. In the **Notifications** tab, click **New.**

   The Add Notification for Step window opens. The Add Notification for Step window has two tabs: **Setup** and **Message.**

3. In the Add Notification for Step window, select the **Setup** tab.

   The **Setup** tab is the default tab.

4. Configure the **Setup** tab, 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. In the **Setup** tab, click **OK.**

   The **Setup** tab closes. The Workflow Step window opens.

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

   The changes are added to the workflow.

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, open the **Notifications** tab.

   See *Configuring Notifications for Workflow Steps* on page 70. The **Notifications** tab opens.

2. In the **Notifications** tab, click **New.**

   The Add Notification for Step window opens. The Add Notification for Step window has two tabs: **Setup** and **Message.**

3. In the Add Notification for Step window, select the **Setup** tab.

   The **Setup** tab is the default tab.

4. In the **Setup** tab, configure the Interval field as follows:

Field	Value	Notes
Event		Selects any event except for **All.**
Send Reminder	Yes	Selecting **Yes** enables the Reminder Days field.
Reminder Days	Enter the number of days.	The number of days to wait before sending a reminder notification.

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

   The **Setup** tab closes. The Workflow Step window opens.

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

   The changes are added to the workflow.

Configuring Notification Recipients

When creating a notification, at least one recipient must be added for the message. 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, open the **Notifications** tab.

   See *Configuring Notifications for Workflow Steps* on page 70. The **Notifications** tab opens.

2. In the **Notifications** tab, click **New.**

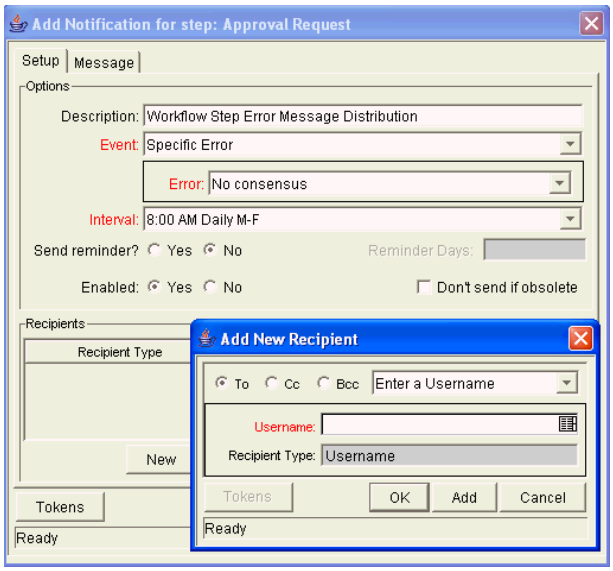
   The Add Notification for Step window opens. The Add Notification for Step window has two tabs: **Setup** and **Message.**

3. In the Add Notification for Step window, select the **Setup** tab.

   The **Setup** tab is the default tab.

4. In the **Setup** tab, click **New.**

   The Add New Recipient window opens.



5. Select how to specify the recipient from the drop-down list:

   - **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 will automatically update the Recipient Type field. For example, selecting **Enter a Security Group** will change the Recipient Type field 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. In the Add New Recipient window, click **OK.**

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

   The **Setup** tab closes. The Workflow Step window opens.

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

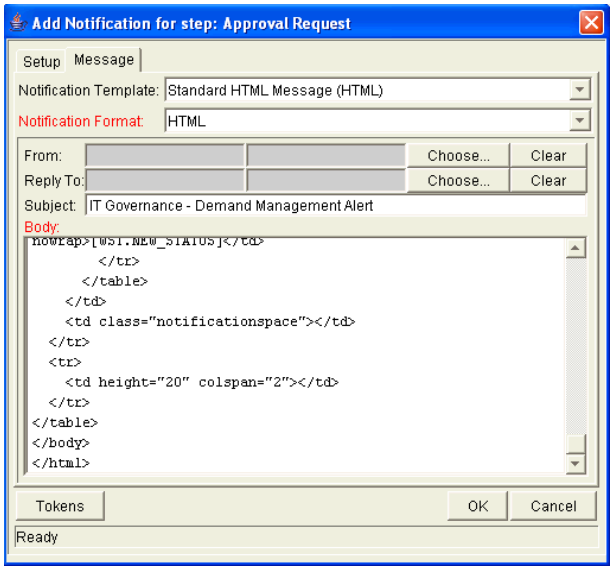
   See *Configuring Notifications for Workflow Steps* on page 70. The **Notifications** tab opens.

2. In the **Notifications** tab, click **New.**

   The Add Notification for Step window opens. The Add Notification for Step window has two tabs: **Setup** and **Message.**

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

   The **Message** tab opens.



4. Select a Notification Template from the drop-down list.

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

5. In the Notification Format field, select **HTML** or **Plain Text** from the drop-down list.

   Selecting **HTML** allows more flexibility when formatting the look and feel of the notification. The HTML code can be written and tested in any HTML editor and then pasted into the Body window.

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

7. Construct the body of the message.

   When constructing the body, consider utilizing the following:

   - Token for the URL to the Request Detail page. See *Table 3-2* on page 84 for a list of these tokens.
   - Token for the URL to the package (Workbench or standard interface). See *Table 3-2* on page 84 for a list of these distributions.
   - **Tokens in the body of the message.** Click the **Tokens** button to access the Token Builder window where tokens can be added to the message body.

- **Tokens related to specific package lines or request detail fields.** Add tokens to the Linked Token list to include tokens that resolve information related to the individual package line or request detail field.

8. In the **Message** tab, click **OK.**

   The Add Notification for Step window closes. The **Notifications** tab is enabled.

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

   The changes to the workflow are saved.

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 or Netscape Messenger), 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 long to set the timeout.

To set timeouts for a workflow step:

1. Open the Workflow Workbench.

   To open the Workflow Workbench, see *Opening the Workflow Workbench* on page 49. The Workflow Workbench window opens.

2. Open a workflow.

   The Workflow window opens.

3. Open the **Layout** tab of the Workflow window.

4. Right-click a workflow step.

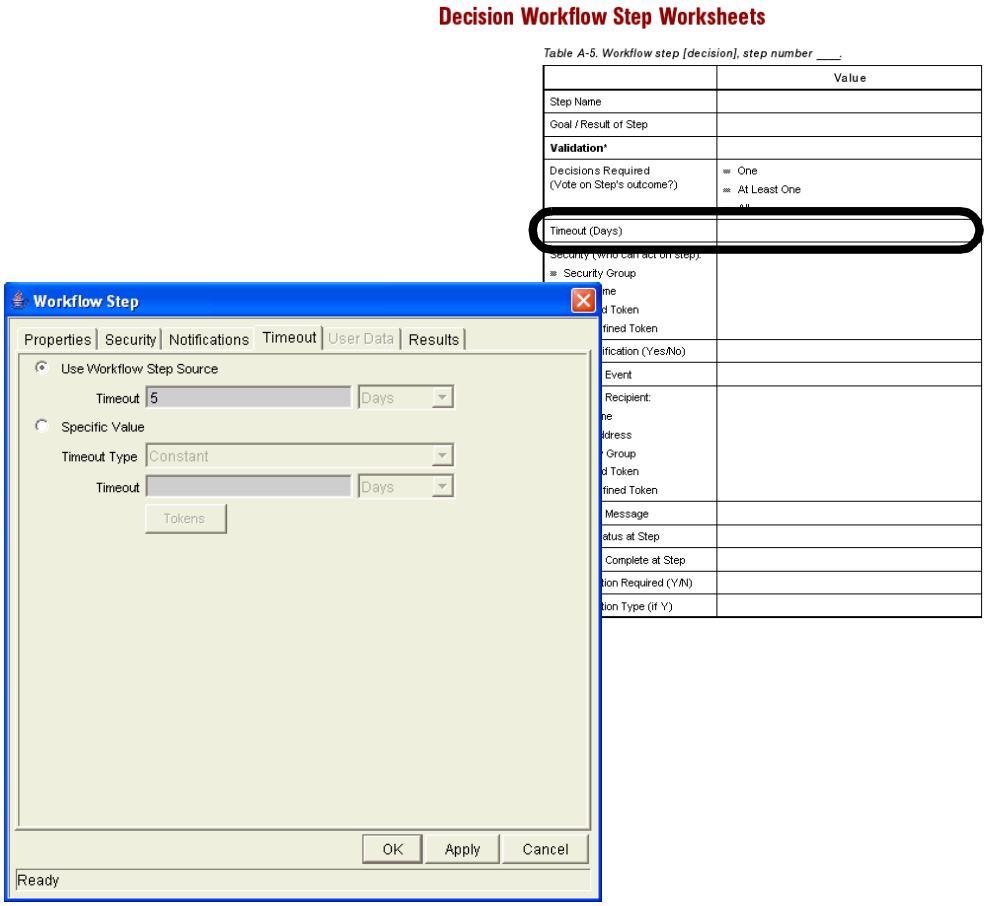
   The workflow step is highlighted. A menu window opens.

5. In the menu window, select **Edit.**

   The Workflow Step window opens.

6. In the Workflow Step window, select the **Timeout** tab.

   The **Timeout** tab opens.



7. Configure the timeout as follows:

   - **Use Workflow Step Source.** This is the default setting. The Workflow Step Source field determines the workflow step's timeout. The Timeout and Interval fields are disabled.

   - **Specific Value.** You can enter a value for the workflow step's timeout according to the Timeout Type entry.

8. In the **Timeout** tab of the Workflow Step window, click **Apply.**

   The changes are applied to the workflow.

Configuring Transitions for Workflow Steps

Transitions are the rules that logically connect workflow steps. They are added to a workflow to establish what 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 to 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 of 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.** (**Demand Management only.**) 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. Open the Workflow Workbench.

   To open the Workflow Workbench, see *Opening the Workflow Workbench* on page 49. The Workflow Workbench window opens.

2. Open a workflow.

   The Workflow window opens.

3. Open the **Layout** tab of the Workflow window.

4. Right-click a workflow step.

   The workflow step is highlighted. A menu window opens.

5. In the menu window, 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 90.

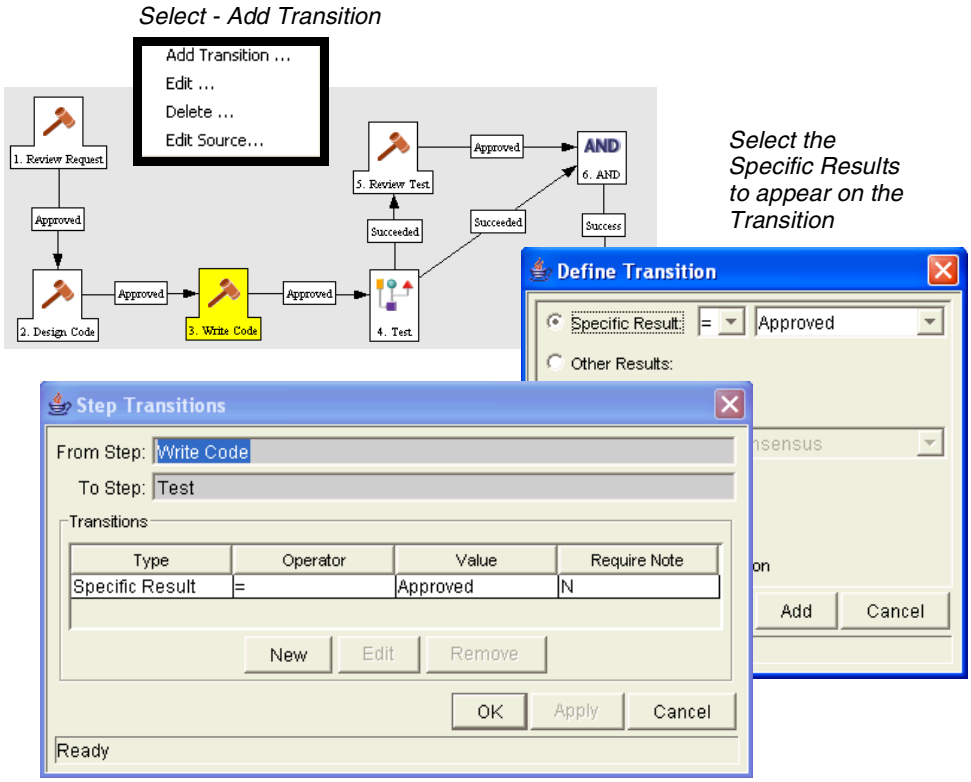
7. In the Define Transitions window, from the Specific Results drop-down list, select the appropriate transition.

8. In the Define Transition window, click **OK.**

   The Define Transition window closes. The Step Transitions window is enabled.

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

   The transition is added to the Step Transitions window. Clicking **Apply** keeps the Step Transition window open. To add another validation to the transition, click **New** and 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. At the bottom of the **Layout** tab, click **Save.**

    The changes to the workflow are saved.

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 request's Priority field 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. Open the Workflow Workbench.

   To open the Workflow Workbench, see *Opening the Workflow Workbench* on page 49. The Workflow Workbench window opens.

2. Open a workflow.

   The Workflow window opens.

3. Open the **Layout** tab of the Workflow window.

4. Configure an immediate execution workflow step.

   a. From 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 in Execution Window	Value
Workflow Scope	**Requests** for request tracking and resolution systems, **Packages** for deployment systems, **Release Distribution** for release systems.
Execution Type	**Token**
Processing Type	**Immediate**
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. In the Execution window, click **OK.**

   d. The new immediate execution workflow step is saved and the Execution window closes.

5. Add the new immediate 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.

   a. In the menu window, select **Add Transition.**

      The menu window closes. The workflow step remains highlighted.

   b. 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.

   c. In the Define Transitions window, from the Specific Results drop-down list, select the appropriate transition.

   d. In the Define Transition window, click **OK.**

      The Define Transition window closes. The Step Transitions window is enabled.

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

      The transition is added to the Step Transitions window. The Step Transitions window closes. The defined transition name is added to the transition line.

8. At the bottom of the **Layout** tab, click **Save.**

   The changes to the workflow are saved.

Adding Transitions Based on Data in Tables

It is possible to 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 list are determined by a SQL query of a database table.

As with any execution step, configure this transition to be an immediate or manual step.

Adding Transitions Based on All But One Specific Value

It is possible to transition based on all but one specified value. You can use Other Results when multiple transitions are exiting a single step. Other Results will act 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 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.

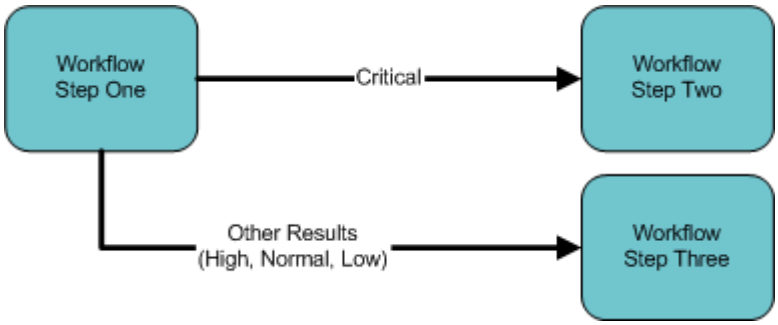


Figure 3-14. Transitions using other results

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 select **All Results.**

Consider using an **All Results** transition when kicking off a sub-process. Note that you can still define transitions based on Specific Results or errors when you select **All Results.** You can bring the process together later using an AND step.

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

It is possible to transition based on a specific error that occurs during an execution step. The business process may then be branched based on likely execution errors such as **Timeout, Command Execution,** or **Invalid Token** (see *Table 3-4*). When adding a transition, select the Specific Error radio button in the Define Transition window. From the drop-down list, select the error.

Table 3-4. Workflow transition errors

Transition Option	Meaning
Multiple Return Results	When the package level subworkflow receives multiple results from package lines that traversed through it.
No consensus	When all users of all security groups, or users linked to the workflow step need to vote, and there is no consensus.

Table 3-4. Workflow transition errors [continued]

Transition Option	Meaning
No recipients	When none of the security groups linked to the workflow step has users linked to it, no user can act on the workflow step.
Timeout	When the workflow step times out. Used for executions and decisions.
Invalid token	Invalid token used in the execution.
ORACLE error	Failed PL/SQL execution.
NULL result	No result is returned from the execution.
Invalid integer	Validation includes an invalid value in the integer field.
Invalid date	Validation includes an invalid value in the date field.
Command execution error	Execution engine has failed or has a problem.
Invalid Result	Execution or subworkflow has returned a result not included in the validation.
Parent closed	For wf_receive or wf_jump steps, a package line is expecting a message from a request that is cancelled or closed.
Child closed	For wf_receive or wf_jump steps, a request is expecting a message from a package line that is cancelled or closed.
No parent	For wf_receive or wf_jump steps, a package line is expecting a message from a request that has been deleted.
No child	For wf_receive or wf_jump steps, a request is expecting a message from a package line that has been deleted.
Multiple jump results	For wf_jump steps in a package line, different result values were used to transition to the step.

Adding Transitions Back to the Same Step

It is possible to retain the option of resetting failed execution workflow steps, rather than immediately transition along a failed path. This is often helpful when troubleshooting the execution (see *Figure 3-15*).

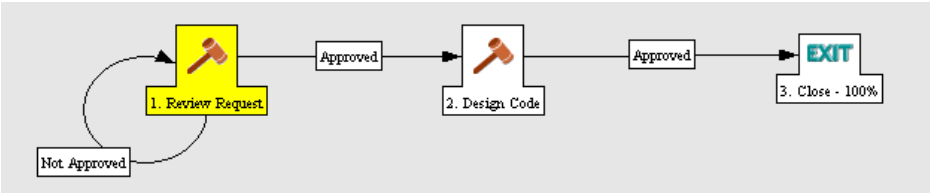


Figure 3-15. Transitioning back to the same step

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 will 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. Open the Workflow Workbench.

   To open the Workflow Workbench, see *Opening the Workflow Workbench* on page 49. The Workflow Workbench window opens.

2. Open a workflow.

   The Workflow window opens.

3. Open the **Layout** tab of the Workflow window.

4. Configure an immediate execution workflow step.

   a. From 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 in Execution Window	Value
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. In the Execution window, click **OK.**

d. The new execution workflow step is saved and the Execution window closes.

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.

   a. In the menu window, select **Add Transition.**

      The menu window closes. The workflow step remains highlighted.

   b. Select several points near the execution workflow step, 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 Define Transitions window, from the Specific Results drop-down list, select the appropriate transition.

   The Validations in the Specific Results drop-down list are the validation created for the execution workflow step. For example, select **Failed - Reset**.

d. In the Define Transition window, click **OK.**

   The Define Transition window closes. The Step Transitions window is enabled.

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

   The transition is added to the Step Transitions window. The Step Transitions window closes. The defined transition name is added to the transition line.

8. At the bottom of the **Layout** tab, click **Save.**

   The changes to the workflow are saved.

Adding Transitions Based on Previous Workflow Step Results

It is possible to 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, in 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, in the **Properties** tab of the Workflow Step window, specify the name of the workflow parameter in the Workflow Parameter field.

One step in this example process requires the user to route the request based on the type of change (code or database). The decision made at this step is then considered later in the process to correctly route rework of the specific type.

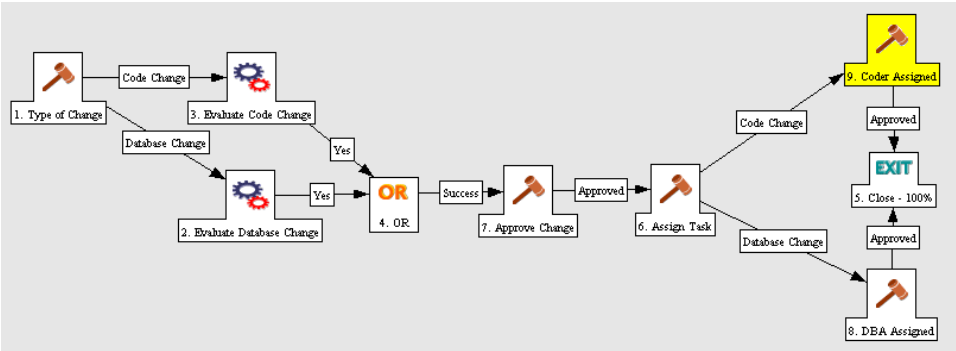


Figure 3-16. Add a transition based on a previous workflow step

To add a transition based on a previous workflow step:

1. Open the Workflow Workbench.

   To open the Workflow Workbench, see *Opening the Workflow Workbench* on page 49. The Workflow Workbench window opens.

2. Open a workflow.

   The Workflow window opens. The **Workflow** tab is displayed.

3. In the **Workflow** tab, create a Workflow Parameter.

   a. In the **Workflow** tab, in the parameters section, click **Add.**

      The Workflow Parameters window opens.

   b. Complete the fields in the Workflow Parameters window.

   c. In the Workflow Parameters window, click **OK.**

   d. The workflow parameter is saved and the Workflow Parameters window closes.

4. In the Workflow window, select the **Layout** tab.

   The **Layout** tab opens.

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

   Note that the validation used in this step should 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 as displayed in the following illustration:

c. In the Execution window, click **OK.**

d. The new execution workflow step is saved and the Execution window closes.

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

   a. Add a workflow step to the workflow.

      The Workflow Step window opens.

   b. In the Workflow Step window, in the **Properties** tab, select the workflow parameter from the Workflow Parameter drop-down list.

   c. In the **Properties** tab, click **OK.**

7. Add the steps and transitions as shown in *Figure 3-16* on page 99.

8. In the **Layout** tab, click **OK.**

   The changes to the workflow are saved.

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.

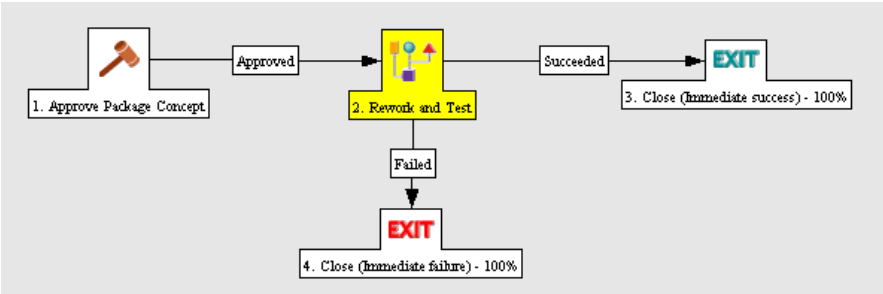


Figure 3-17. Transitioning to and from subworkflows

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. The subworkflow field cannot be edited 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 3-18* on page 102 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 workflow step's Define Transition window, the validation values are displayed as the Specific Results field.

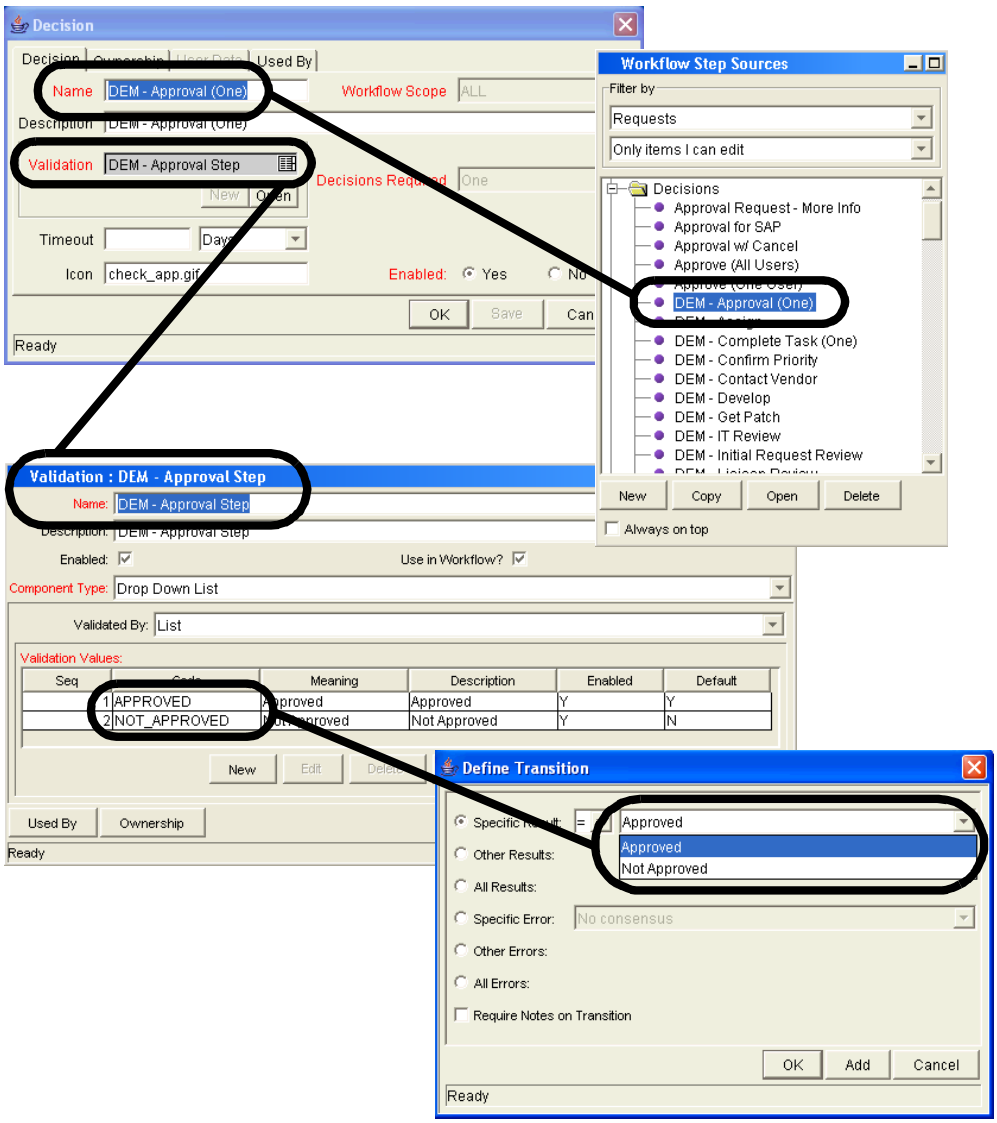


Figure 3-18. Workflow step sources and validations

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 autocomplete list) minus the WHERE clause.

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. Open the Workflow Workbench.

   To open the Workflow Workbench, see *Opening the Workflow Workbench* on page 49. The Workflow Workbench window opens.

2. Open a workflow.

   The Workflow window opens. The **Workflow** tab is displayed.

3. Click the **Layout** tab.

   The layout tab opens.

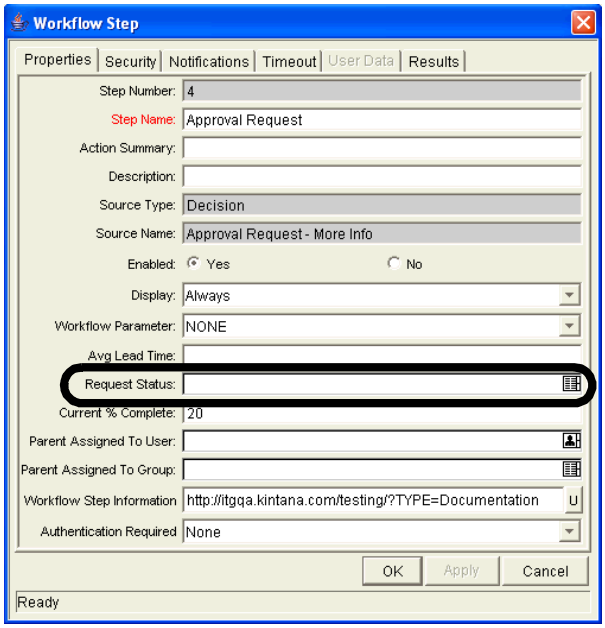
4. Right-click a workflow step.

   The workflow step is highlighted. A menu window opens.

5. In the menu window, select **Edit.**

   The menu window closes. The Workflow Step window opens.

6. Select the desired request status from the Request Status autocomplete list.



7. Repeat as needed with all necessary workflow steps.

8. In the **Layout** tab, click **OK.**

As the request progresses through this workflow, it will take on the status assigned in each workflow step.Not all workflow steps need to have a Request Status assigned. A request type retains the last-encountered Status.

Integrating Request Type Commands and Workflows

Request type commands define the execution layer within request management. While most of the resolution process for a request is analytically based, cases may arise for specific request types where system changes are required. In these cases, 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 panel (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 (Change 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 that a request is a **Predecessor** to the package. This means the package will not 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. When the Approval step is successful, 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 graphically represented in *Figure 3-19* on page 108.

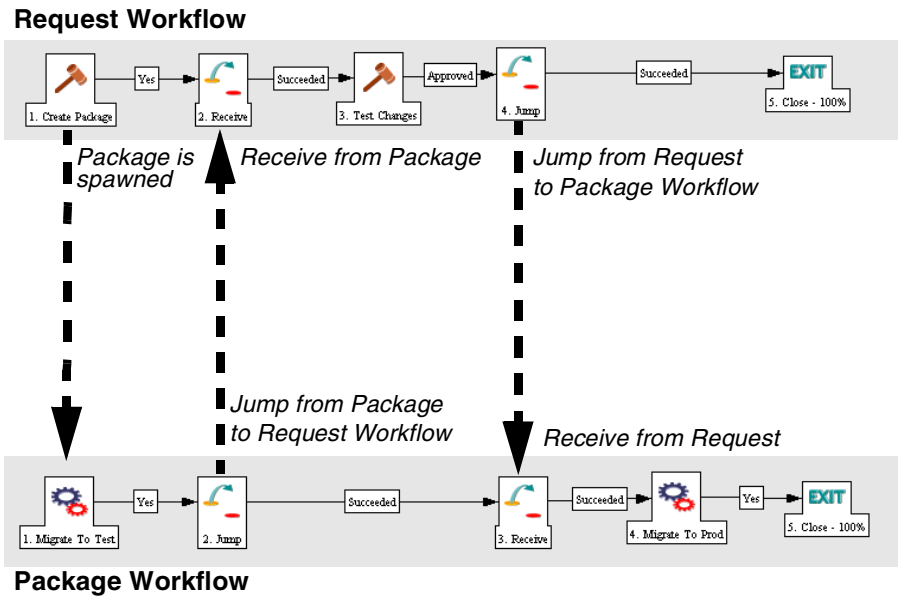


Figure 3-19. Jump/Receive workflow steps

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

To establish communication between request and package workflows:

1. Set up the **WF - Jump/Receive Step Labels** validation for use in the Workflow Step window.

   This validation is used to group a jump and receive workflow step pair. The selected **WF - Jump/Receive Step Labels** must match in the paired jump and receive Workflow Step windows. See *Setting Up WF - Jump/Receive Step Label Validations* on page 109.

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

   See *Generating Jump Step Sources* on page 111.

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

   See *Generating Receive Step Sources* on page 113.

4. Verify that both the jump and receive workflow steps specify the same **WF - Jump/Receive Step Labels.**

   See *Including Jump and Receive Workflow Steps in Workflows* on page 114.

5. Verify that the transitions exiting the jump workflow step and receive workflow steps match the possible values entering the jump workflow step.

Setting Up WF - Jump/Receive Step Label Validations

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

1. Open the Workflow Workbench.

   To open the Workflow Workbench, see *Opening the Workflow Workbench* on page 49. The Workflow Workbench window opens.

2. From the shortcut bar, select **Configuration > Validations.**

   The Validation Workbench opens.

3. In the **Query** tab of the Validation Workbench, in the Validation Name field enter **WF - Jump/Receive Step Labels.**

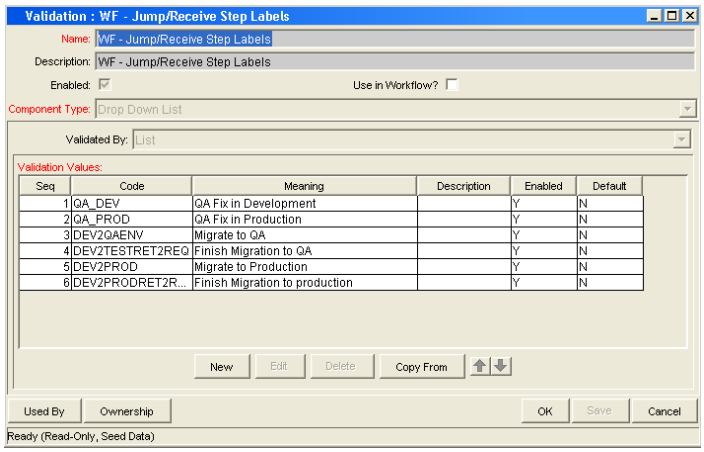
4. In the Validation Workbench, click **List.**

5. In the Validation Workbench, click the **Results** tab.

   The **WF - Jump/Receive Step Labels** is listed in the **Results** tab.

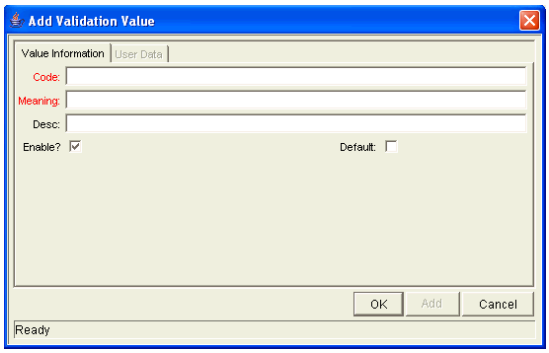
6. Highlight **WF - Jump/Receive Step Labels** and click **Open.**

   The Validation window opens.



7. Click **New** to define a new validation value that is used to link two workflows together.

   The Add Validation Value window opens.



8. In the Add Validation Value window, enter the Code, Meaning and Description.

9. In the Add Validation Value window, click **OK.**

   The Add Validation Value window closes. The Validation window is enabled.

10. In the Validation window, click **Ownership** to select which ownership groups will have the ability to edit this validation.

11. In the Validation window, click **OK** to close the Validation window.

    The changes to the validation are saved.

The new validation value is now included in the Jump/Receive Step Label drop-down list in the Workflow Step window.

For More Information

For more information concerning configuring validations, see *Commands, Tokens, and Validations Guide and Reference*.

Generating Jump Step Sources

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

1. Open the Workflow Workbench.

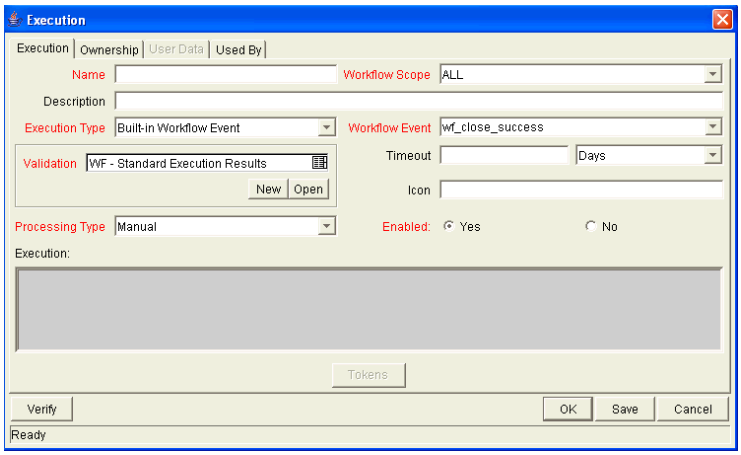
   To open the Workflow Workbench, see *Opening the Workflow Workbench* on page 49. The Workflow Workbench window opens.

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 drop-down list, depending on the desired application of the workflow.

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

5. In the Execution window, from the Execution Type drop-down list, select **Built-in Workflow Event.**

6. In the Execution window, from the Workflow Event drop-down list, select **wf_jump.**

7. In the Execution window, from the Validation drop-down list, select or create a validation which will be used 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. In the Execution window, fill in any other required or optional information, such as Name, Description, or Processing Type.

9. In the Execution window, select the **Ownership** tab.

10. Select which Ownership Groups will have the ability to edit this execution workflow step.

11. In the Execution window, 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.

Generating Receive Step Sources

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

1. Open the Workflow Workbench.

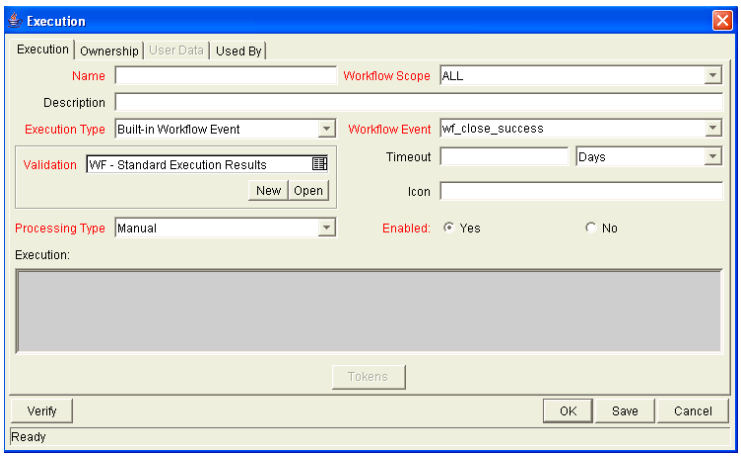
   To open the Workflow Workbench, see *Opening the Workflow Workbench* on page 49. The Workflow Workbench window opens.

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 Execution window, from the Workflow Scope drop-down list, select either **Packages** or **Requests**, depending on the desired application of the workflow.

5. In the Execution window, from the Execution Type drop-down list, select **Built-in Workflow Event.**

6. In the Execution window, from the Workflow Event drop-down list, select **wf_receive.**

7. Select or create a validation which will be used 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. In the Execution window, fill in any other required or optional information, such as Name, Description, or Processing Type.

9. In the Execution window, select the **Ownership** tab.

10. Select which Ownership Groups will have the ability to edit this execution workflow step.

11. In the Execution window, 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 receive step must have a paired jump step in another workflow.

Including Jump and Receive Workflow Steps in Workflows

To include a Jump and Receive workflow step in a workflow:

1. Open the Workflow Workbench.

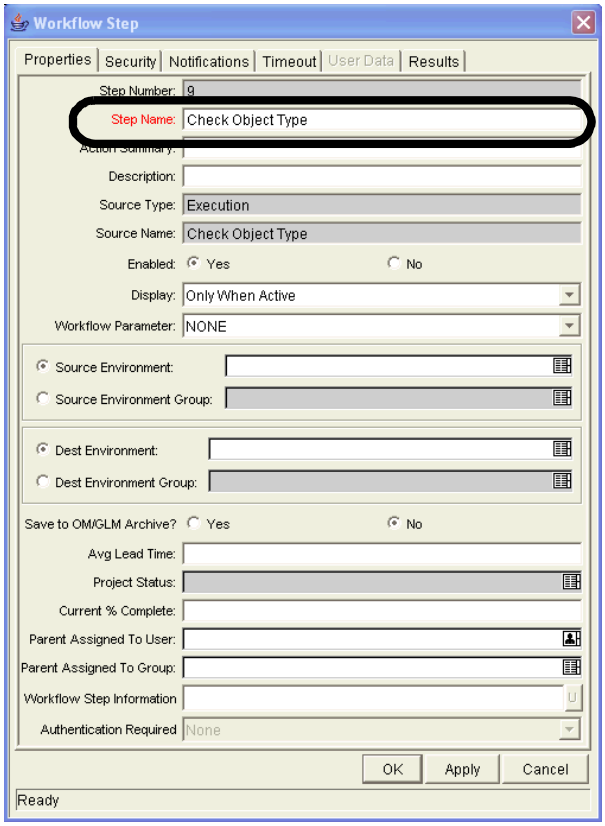
   To open the Workflow Workbench, see *Opening the Workflow Workbench* on page 49. The Workflow Workbench window opens.

2. Open a workflow.

   The Workflow window opens.

3. In the Workflow Step Sources window, in the Executions folder, drag and drop either the Jump or Receive step from the Workflow Step Sources window into the workflow's **Layout** tab.

   The Workflow Step window opens.



4. In the Workflow Step window, select an item from the Jump/Receive Step Label drop-down list.

   This item must be the same for a paired jump and receive workflow step.The Jump/Receive Step Label is the key communication link between separate workflows. The communicating jump and receive workflow steps must have a matching Jump/Receive Step Label. It is also important that the Jump/Receive Step Label is unique for any jump and receive pair.

5. In the Workflow Step window, enter any additional workflow step information.

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

Repeat this process for the other paired workflow step (jump or receive workflow step), depending on which one you configured first.

Chapter

4

Configuring Workflow Components

In This Chapter:

Overview of Workflow Step Sources

This chapter covers information concerning Demand Management workflows, Change Management workflows, and Release Management workflows.

Mercury IT Governance Center includes a number of standard workflow step sources that can be added 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:

- The validation associated with the step (and thus the list of valid transition values out of the step).
- The voting requirements of the step.
- The default timeout value for the step. Each step can be configured to have a unique timeout value.
- The 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, simply 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:

- A step source that is being used in a workflow can not be deleted.
- A validation for a step source that is being used can not be changed. If the validation needs to change, copy the step source and configure a new validation.
- The workflow step source must be Enabled before it can be added to a workflow.
- Only add step sources to a workflow when the workflow has a matching workflow scope, or the step source has a scope of **All.**
- A workflow step in a workflow that has processed a request, package line, or release can not be deleted. This would compromise data integrity. Instead of deleting the workflow step, remove all transitions to and from the workflow step and disable the workflow step.

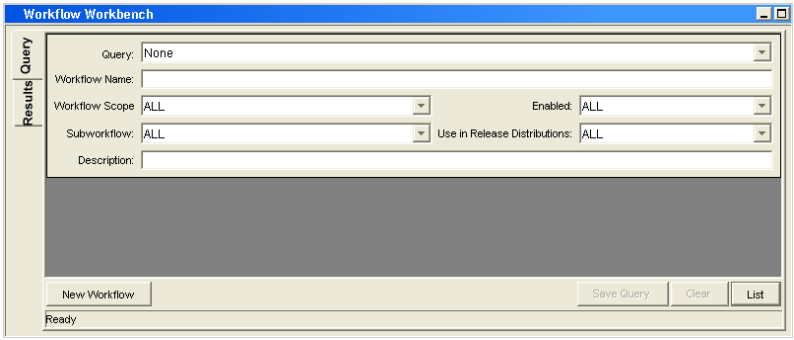
Opening the Workflow Workbench

To open the Workflow Workbench:

1. Log on to the Mercury IT Governance Center.
2. From the menu bar, select **Administration > Open Workbench.**
3. A Workbench status window opens. A few minutes later, a Warning - Security window opens.
4. In the Warning - Security window, select **Yes.**

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

   The Workflow Workbench window opens.



For More Information

For information on how to search and select an existing workflow, copy a workflow, and delete a workflow, see *Getting Started*.

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

To create a new workflow step source:

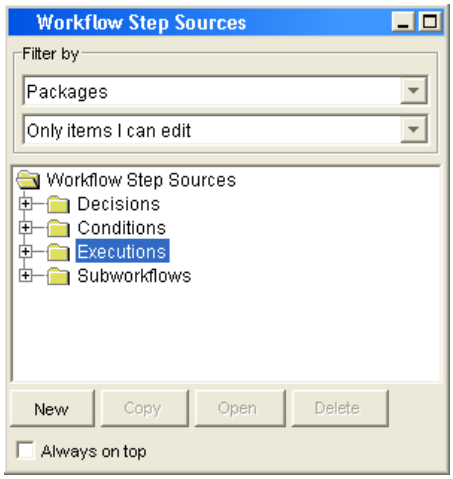
1. Open the Workflow Workbench.

   To open the Workflow Workbench, see *Opening the Workflow Workbench* on page 120. The Workflow Workbench window opens.

2. Open a workflow.

   The Workflow window opens.

3. Select the Workflow Step Sources window.

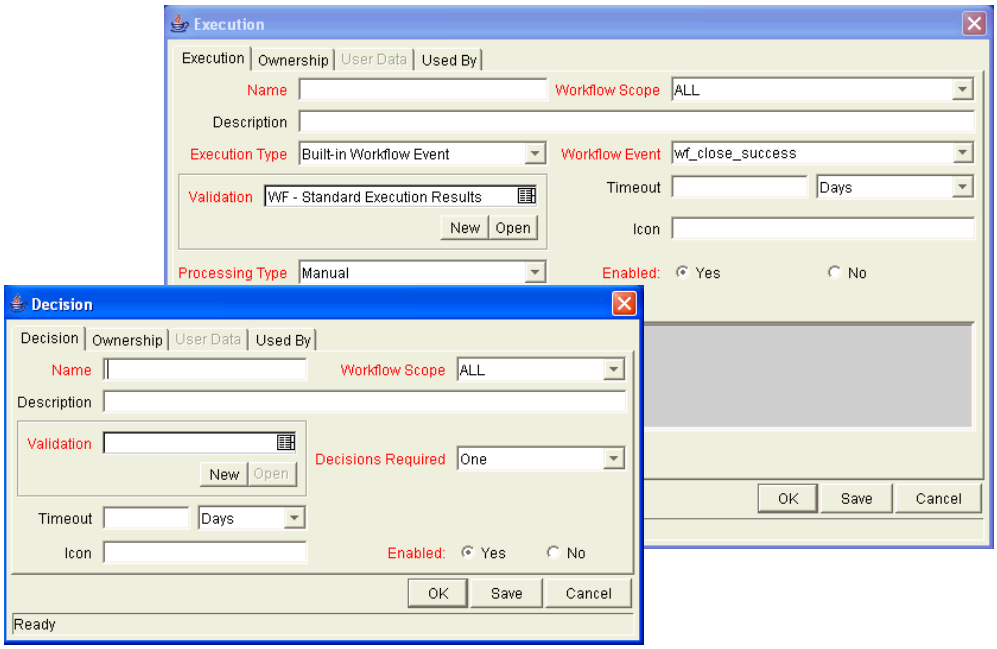


4. In Filter by, select **Requests, Packages,** or **Release Distributions,** depending on the type of workflow.

5. Select the Decisions or Executions folder.

6. At the bottom of the Workflow Step Sources window, click **New.**

   A window corresponding to the selected workflow step source type opens. For decision workflow steps, the window is Decision. For execution workflow steps, the window is Execution.



7. Enter the required information and any optional information needed to define the workflow step.

   For information on configuring a specific workflow step source, see:

   - *Creating Decision Workflow Step Sources* on page 125
   - *Creating Execution Workflow Step Sources* on page 129.

8. 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 123.

9. In the Decision or Execution window, in the Enabled field, select **Yes.**

10. In the Decision or Execution window, 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.

Configuring Ownership of Workflow Step Sources

When configuring a workflow step source, you can specify who will be able to edit the workflow step source.

To configure the ownership of a new workflow step source:

1. Open the Workflow Workbench.

   To open the Workflow Workbench, see *Opening the Workflow Workbench* on page 120. The Workflow Workbench window opens.

2. Open a workflow.

   The Workflow window opens.

3. Open a decision or execution workflow step source.

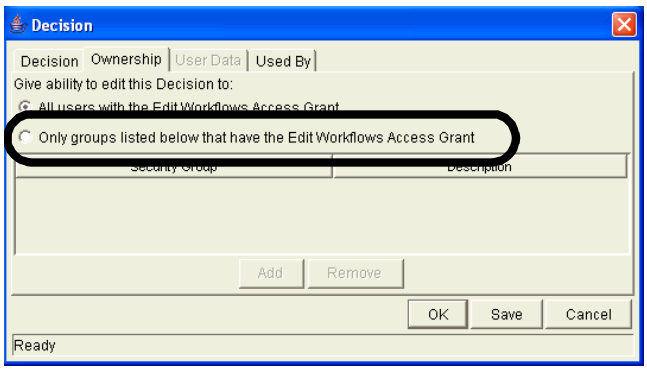
   A window corresponding to the selected workflow step source type opens. For decision workflow steps, the window is Decision. For execution workflow steps, the window is Execution.

4. In the Decision or Execution window, click the **Ownership** tab.

   The **Ownership** tab opens. Configuring the **Ownership** tab determines which security groups will have the ability to edit this Execution or Decision workflow step. The default is to allow all security groups with the Edit Workflows access grant to edit a workflow step source.

5. In the **Ownership** tab, select Only groups listed below that have the Edit Workflows access grant.

   The **Add** button is enabled.



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

   The Add Security Group window opens.

7. In the Add Security Group window, in Security Group, select a security group from the autocomplete list.

8. In the Add Security Group window, click **OK.**

   The Add Security Group window closes. The security group is added to the workflow step source. The only users who can now edit this workflow step source must belong to a listed security group and the security group must have the Edit Workflow access grant.

9. In the **Ownership** tab, 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 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.

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)	

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

To create a new decision workflow step source:

1. Open the Workflow Workbench.

   To open the Workflow Workbench, see *Opening the Workflow Workbench* on page 120. The Workflow Workbench window opens.

2. Open a workflow.

   The Workflow window opens.

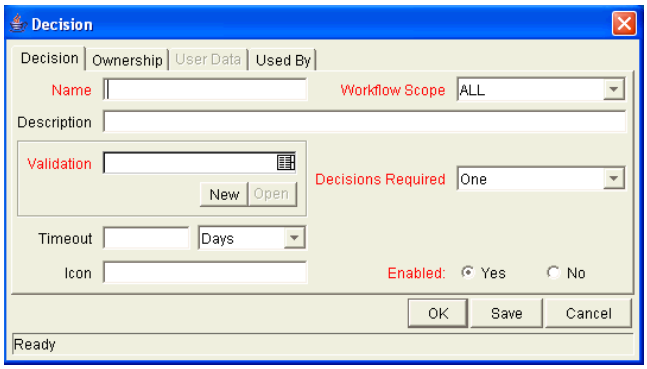
3. Select the Workflow Step Sources window.

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

5. Select the Decisions folder.

6. At the bottom of the Workflow Step Sources window, click **New.**

   The Decision window opens.



7. In the Decision window, make sure you are in the **Decision** tab.

   The **Decision** tab is the default tab showing.

8. Complete the fields in the **Decision** tab as specified in the following table:

Field	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 drop-down 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 Change Management package workflows. • **Release Distributions.** For Mercury Change Management release workflows.
Description	Description of the workflow step source.
Validation	Validations determine the transition values for the workflow step. Use the drop-down list to select a validation.

Field	Description
Decisions Required	Defines the number of decisions required for the workflow step. Use the drop-down 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. It is possible to 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.**
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.

Field	Description
Icon	A different graphic can be specified to represent steps of this source for use in the workflow **Layout** tab. The graphic needs to exist in the `icons` subdirectory in the Mercury IT Governance Server. 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.

9. In the Decision window, 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 with the Edit Workflows access grant 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 123.

10. In the Decision window, 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. In the Decision window, click the **Used By** tab.

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

12. At the bottom of the Decision window, 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.

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	

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

To create a new execution workflow step source:

1. Open the Workflow Workbench.

   To open the Workflow Workbench, see *Opening the Workflow Workbench on page 120*. The Workflow Workbench window opens.

2. Open a workflow.

   The Workflow window opens.

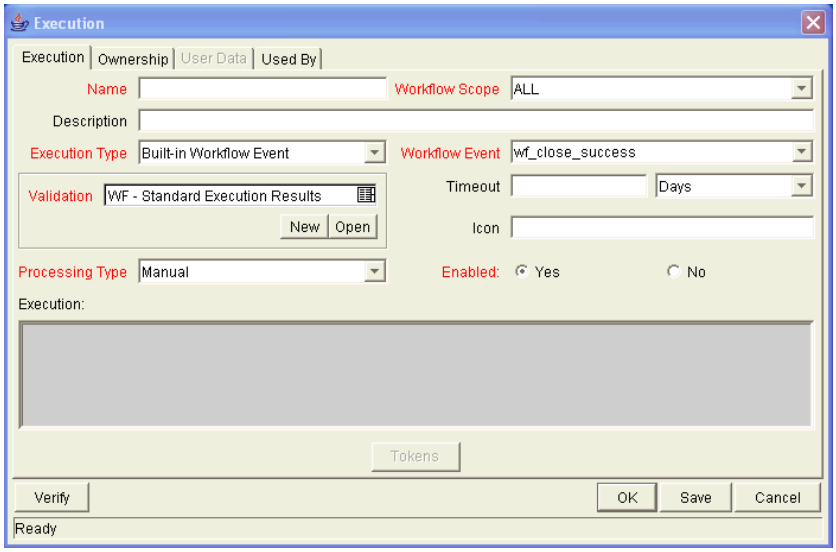
3. Select the Workflow Step Sources window.

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

5. Select the Executions folder.

6. At the bottom of the Workflow Step Sources window, click **New.**

   The Execution window opens.



7. In the Execution window, make sure you are in the **Execution** tab.

   The **Execution** tab is the default tab showing.

8. Complete the fields in the **Execution** tab as specified in the following table:

Field	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 drop-down 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 Change Management package workflows. • **Release Distributions.** For Mercury Change Management release workflows.
Description	Description of the step source.

Field	Description
Execution Type	Used to select the type of execution to be performed. Use the drop-down 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.
Workflow Event	For Execution Type **Built-in Workflow Event,** the specific event to perform must be selected. The available choices in the drop-down 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 Change Management package. • **rm_ready_for_release.** Generates a Mercury Demand Management request. • **create_package_and_wait.** Generates a Mercury Change 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 Change Management and Mercury Demand Management) Instructs the workflow to proceed to a corresponding Receive Workflow Step in another workflow. • **wf_receive.** (Mercury Change 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 Change 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.

Field	Description
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.
Processing Type	Defines when the execution is performed. Use the drop-down 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 drop-down 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 in the Mercury IT Governance server. All icons are in .gif format.
Enabled	The workflow step source must be enabled in order to add it to the workflow layout.

9. In the Execution window, 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 with the Edit Workflows access grant 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 123.

10. In the Execution window, 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. In the Execution window, click the **Used By** tab.

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

12. At the bottom of the Execution window, 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 in Execution Window	Value
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 in Execution Window	Value
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 existing or create a new validation.)
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

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

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

PL/SQL function execution workflow steps are used when a workflow 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 in Execution Window	Value
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.

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

Field in Execution Window	Value
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 in Execution Window	Value
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 in Execution Window	Value
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 that the transition will be based on.
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.**

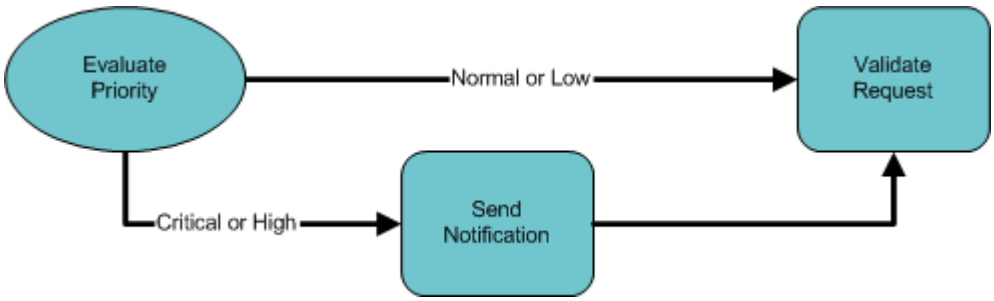


Figure 4-3. Transitioning based on a token

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 in Execution Window	Value
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.

A subworkflow can be selected from the Workflow Step Sources window and dragged onto the **Layout** tab. When the package, request, or release reaches the subworkflow step, it follows the path defined in that subworkflow. The subworkflow will either close within that workflow or return 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:

- The Workflow window contains a Sub-workflow radio button which should be set to **Yes.**
- The validation for the step leaving the subworkflow layout should match the subworkflow step in the parent workflow.

Subworkflows Returning to Demand Management Workflows

Execution workflow steps can be configured to automatically return 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. The transitions leading into the return step must match the validation established for the subworkflow step. You must verify that the validation defined for the subworkflow step is synchronized with the transitions entering the return step.

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

Field in Execution Window	Value
Name	Enter a descriptive name for the workflow step source.
Workflow Scope	**Requests**
Execution Type	**Built-in Workflow Event**
Workflow Event	**wf_return**
Processing Type	**Manual** or **Immediate**
Validation	**WF - Standard Execution Results** (This is the default selection. You can select another existing or create a new validation.)
Enabled	**Yes**

Using Workflow Parameters

Use workflow parameters to store the results of a workflow step. This value can then be used later to define a transition. The following lists the rules concerning workflow parameters:

- Workflow parameters can be referenced using the WFI.P token prefix.
- Workflow parameters can be used in PL/SQL and SQL workflow step executions.

Creating Workflow Parameters

To create a workflow parameter:

1. Open the Workflow Workbench.

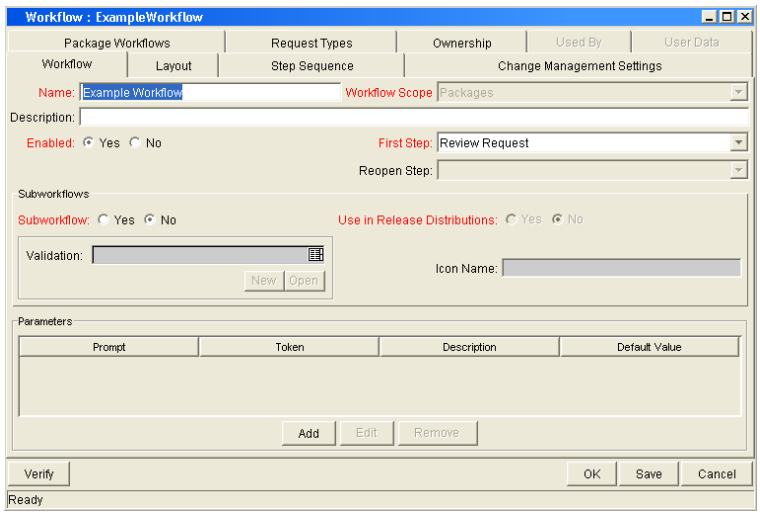
   To open the Workflow Workbench, see *Opening the Workflow Workbench on page 120*. The Workflow Workbench window opens.

2. Open a workflow.

   The Workflow window opens.

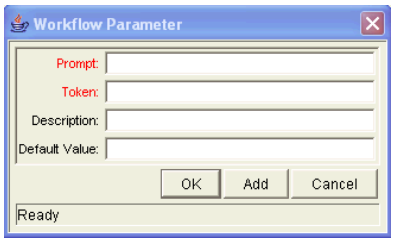
3. In the Workflow window, make sure you are in the **Workflow** tab.

   The **Workflow** tab is the default tab opened.



4. In the **Workflow** tab, click **Add.**

   The Workflow Parameter window opens.



5. Complete the fields in the Workflow Parameter window as specified in the following table:

Field	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.

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

7. The Workflow Parameter window, click **OK.**

8. The Workflow Parameter window closes. The workflow parameter appears in the Parameters section of the **Workflow** tab.

9. In the **Workflow** tab, click **OK.**

   The changes to the workflow are saved.

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. Open the Workflow Workbench.

   To open the Workflow Workbench, see *Opening the Workflow Workbench on page 120*. The Workflow Workbench window opens.

2. Open a workflow.

   The Workflow window opens.

3. In the Workflow window, make sure you are in the **Workflow** tab.

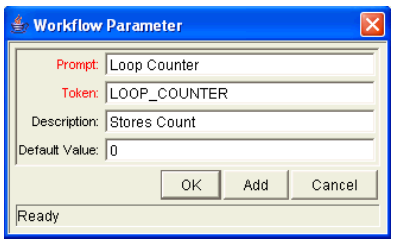
   The **Workflow** tab is the default tab opened.

4. In the **Workflow** tab, click **Add.**

   The Workflow Parameter window opens.

5. Complete the fields in the Workflow Parameter window as specified in the following table:

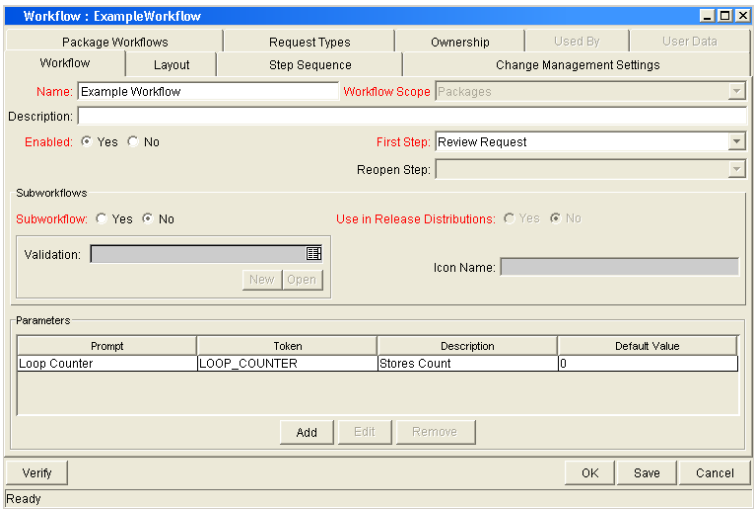
Field	Description
Prompt	Loop Counter
Token	LOOP_COUNTER
Description	Stores count.
Default Value	0



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

7. The Workflow Parameter window, click **OK.**

8. The Workflow Parameter window closes. The workflow parameter appears in the Parameters section of the **Workflow** tab.



9. In the **Workflow** tab, click **OK.**

   The changes to the workflow are saved.

10. Create a new immediate SQL execution workflow step.

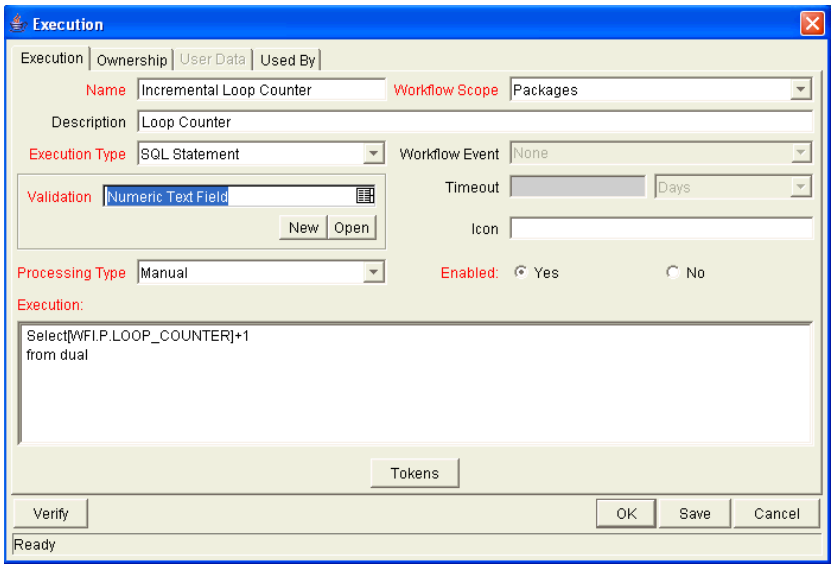
    For details on how to create an SQL execution workflow step, see *Creating Execution Workflow Step Sources* on page 129.

    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.

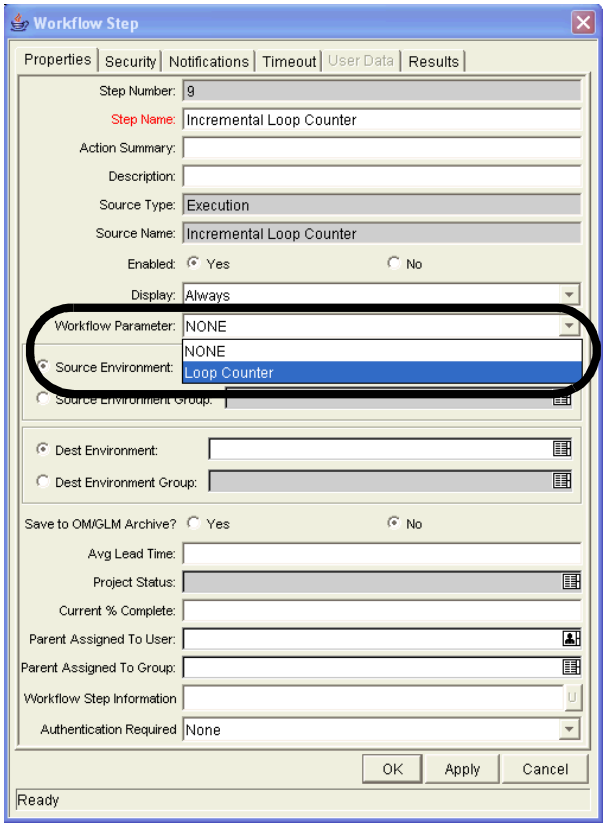
    - A validation for a **Numeric** text field is used. This allows <=, <, >=, and > comparisons to be used in transitions off this step.

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



11. 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.



It is now possible to add transitions to and from the new loop counter step. For example, the loop counter can be added to each time an execution fails. If the execution fails three times, a notification can be sent to the user. If the execution fails five times, management can be 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	All of these entities can be modified or added to a workflow in use.
Transitions Security Notifications Workflow Parameters	All of these entities can be deleted from a workflow in use.
Workflow Steps	This entity cannot be deleted from a workflow in use, but can be renamed. Transitions coming into or going out of a workflow step can be deleted, effectively removing it from the workflow.

When a workflow that is in use is modified 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 when Modifying Security

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.

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

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

Performance Considerations when Migrating Workflows

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.

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

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.
5. 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*.

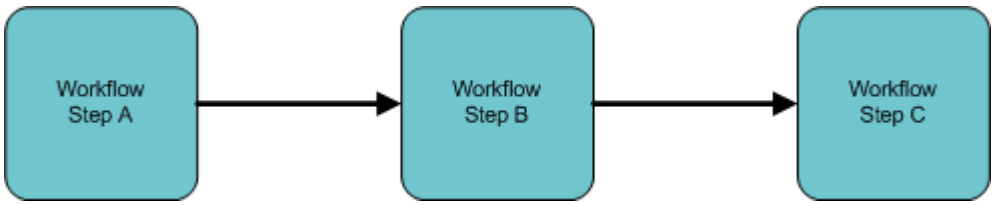


Figure 4-4. Redirecting the workflow, step 1

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*).

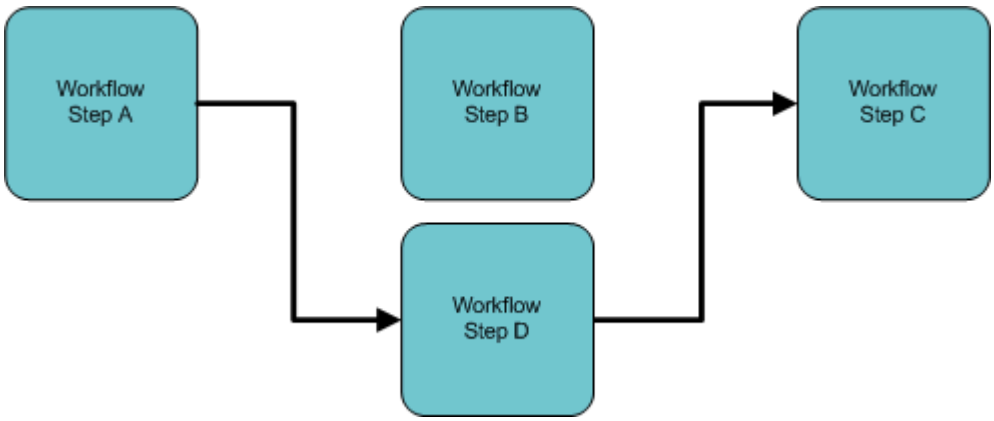


Figure 4-5. Redirecting the workflow, step 2

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

5

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. End-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*).

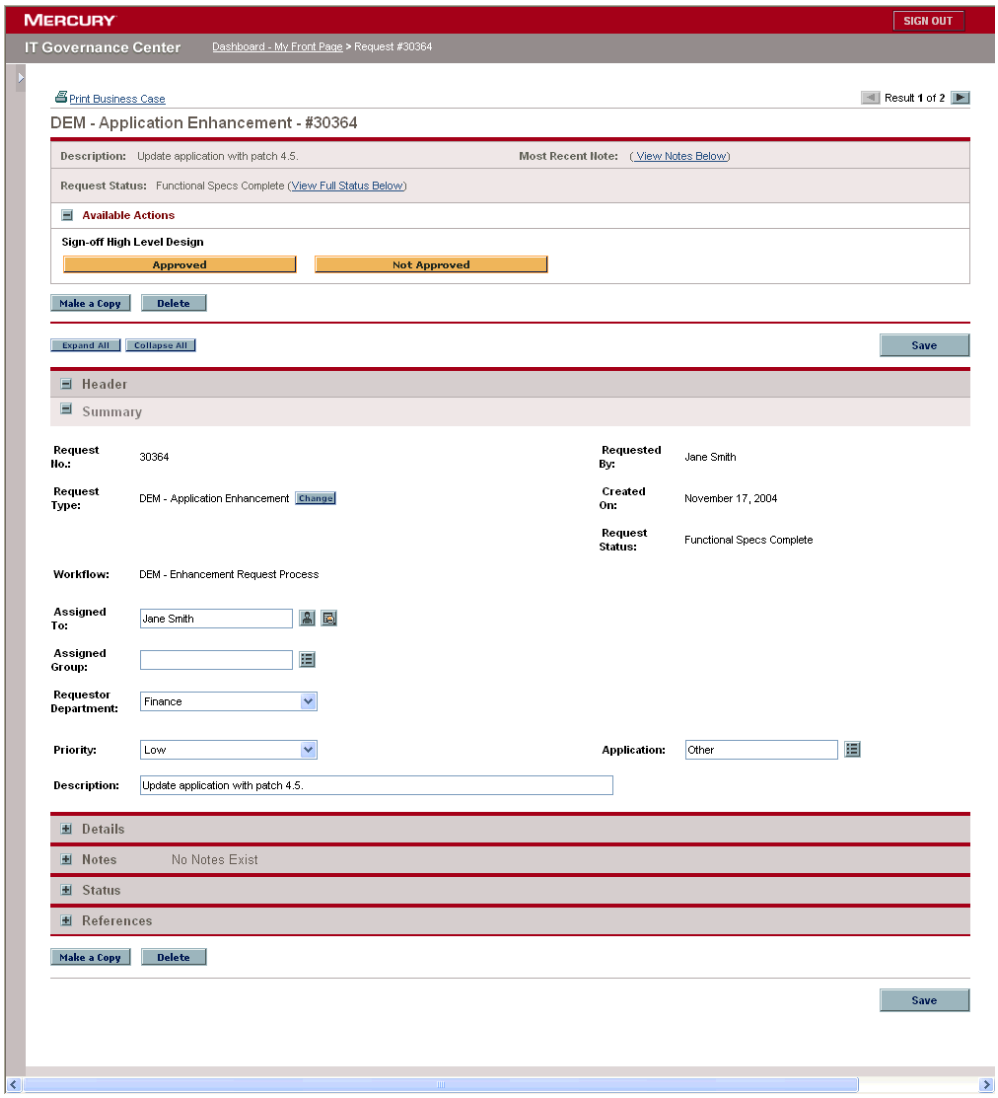


Figure 5-1. Generic request

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*).

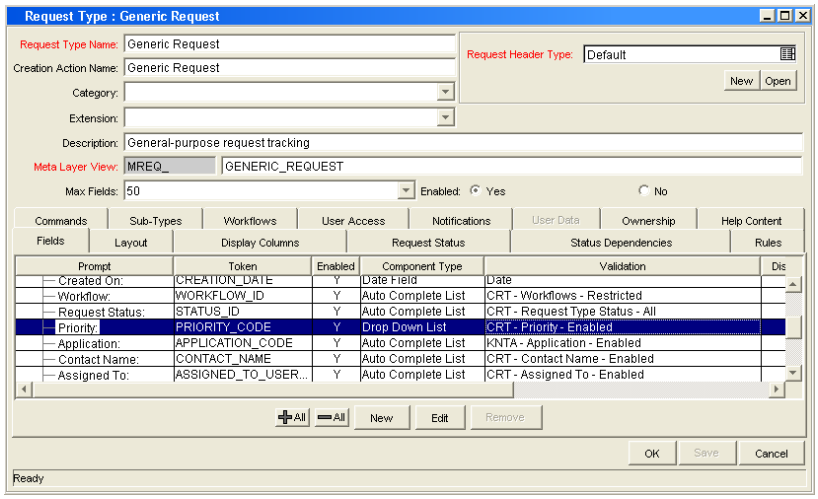


Figure 5-2. Request Type window

The following is a list of the main components of a request type:

- **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 Types* on page 165.

  - **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 added a predefined set of fields to the request type. The **Fields** tab is used to create additional fields for the request type. See *Configuring Fields for Request Types* on page 167.

- **Layout.** Once all of the fields are created for a request type, the layout of those fields can be configured using the **Layout** tab. See *Configuring Layouts for Request Types* on page 180.

- **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 187.

- **Request Status.** While processing a request, the request can acquire different statuses as it progresses along its workflow. These statuses can be used to drive field behavior, linking workflow processes to specific information in the request. See *Configuring Request Statuses for Request Types* on page 189.
- **Status Dependencies.** While processing a request, the request can acquire different statuses as it progresses along its workflow. These statuses can be used to drive field behavior. For example, a specific fields can be changed to a required field when changes are made to a request that effect the request's status. See *Configuring Status Dependencies* on page 194.
- **Rules.** Request rules can be used to set up the automatic population of a request's fields based on various dependencies. See *Configuring Rules for Request Types* on page 198.
- **Commands.** Commands can also be used to control certain behavior of request type fields. At specific workflow execution steps in a request tracking and 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** tab, which can only default based on a single parameter stored in the same request type. See *Configuring Commands for Request Types* on page 205.
- **Sub-Types.** Creates valid sub-types for the request type. For example, a bug request type might have hardware, software, and documentation sub-types. See *Configuring Sub-Types for Request Types* on page 212.
- **Workflows.** Select which workflows can work with a request type. See *Configuring Request Types to Work with Workflows* on page 215.
- **User Access.** The **User Access** tab configures participants of a request type. Participants can then be given specific 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 217.
- **Notifications.** Configure emails to be sent when specific fields in the request type are completed. See *Configuring Notifications for Request Types* on page 221.

- **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.
- **Ownership.** Configure who can edit the request type. See *Configuring Ownerships of Request Types* on page 232.
- **Help Content.** Add help content to fields, sections and request types. See *Configuring Help Contents for Request Types* on page 235.

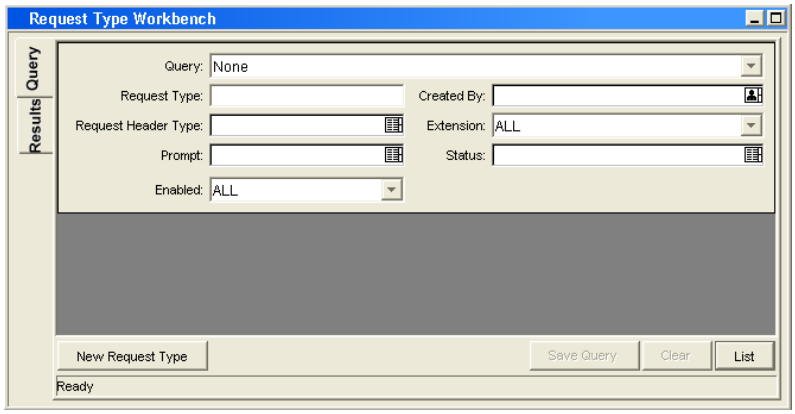
Opening the Request Type Workbench

To open the Request Type Workbench:

1. Log on to the Mercury IT Governance Center.
2. From the menu bar, select **Administration > Open Workbench.**
3. A Workbench status window opens. A few minutes later, a Warning - Security window opens.
4. In the Warning - Security window, select **Yes.**

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

   The Request Type Workbench window opens.



For More Information

For information on how to search and select an existing request type, copy a request type, and delete a request type, see *Getting Started*.

Setting Request Type Defaults

It is possible to select a default request header type and a default workflow when creating a request type. You can also select the default value for the maximum number of fields in a request type.

To set the default request header type and workflow:

1. Open the Request Type Workbench.

   To open the Request Type Workbench, see *Opening the Request Type Workbench* on page 161. The Request Type Workbench window opens. The Query tab is displayed.

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

   The **Results** tab opens. All request types are displayed.

3. In the **Results** tab, click **Setup Request Header.**

   The Request Header Setup Dialog window opens.

4. Complete the fields in the Request Header Setup Dialog window as specified in the following table:

Field	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 will be used for all new request types, unless a different request header type is specified in the individual request type.

5. In the Request Header Setup Dialog window, click **OK.**

   The selected workflow and request header type are now defaults.

To change the default number of fields for request types:

1. Open the Request Type Workbench.

   To open the Request Type Workbench, see *Opening the Request Type Workbench* on page 161. The Request Type Workbench window opens.

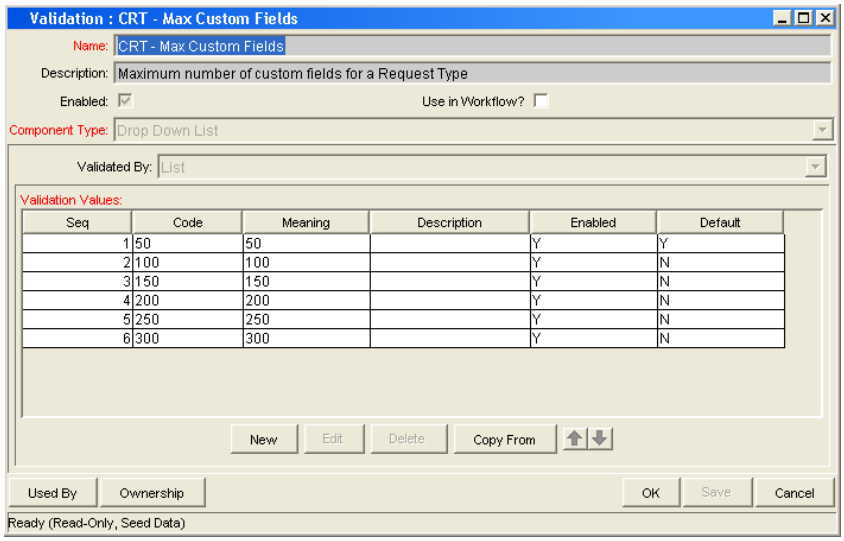
2. From the shortcut bar, select **Configuration > Validation.**

   The Validation Workbench window opens. The **Query** tab is displayed.

3. In the **Query** tab, click **List.**

   The **Results** tab opens. All validations are displayed.

4. Find and open CRT- Max Custom Fields.



5. In the Validation window, click **New.**

   The Add Validation Value window opens.

6. Complete the fields in the Add Validation Value window as specified in the following table:

Field	Description
Code	The validation value. Validation values are in increments of 50. The Code and Meaning fields must be identical.
Meaning	The validation value. Validation values are in increments of 50. The Code and Meaning fields must be identical.
Enable	Makes the validation value available to the system.
Default	Selects this validation value as the default value.

7. In the Add Validation Value window, click **OK.**

   The Add Validation Value window closes. The new validation is added to the Validation window.

8. In the Validation window, click **OK.**

   The changes to the validation are saved.

Configuring General Information for Request Types

To configure the general information for a request type:

1. Open the Request Type Workbench.

   To open the Request Type Workbench, see *Opening the Request Type Workbench* on page 161. The Request Type Workbench window opens.

2. Open a request type.

   The Request Type window opens.

3. Complete the fields in the Request Type window as specified in the following table:

Field	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 in 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 Change Management extension. Select the extension from the drop-down 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 162.

Field	Description
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 autocomplete list, or create a new request header type by clicking **New.**

4. Save the changes to the request type.

   Click **OK** to save the changes and close the Request Type window. Click **Save** to save the changes and leave the Request Type window open. Click **Cancel** to lose the changes and close the Request Type window.

Configuring Fields for Request Types

This section details how to create and configure fields for request types. This section also provides overview information concerning fields for request types.

Overview of Request Type Fields

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

Criteria for Visible Fields

Fields can be configured to be visible or hidden to the user (see *Table 5-1*).

Table 5-1. Criteria for visible fields

Criteria	Description
Field attributes	The field can be set to display or be hidden at all times. This is controlled from the Field window's **Attributes** tab. See *Creating Fields for Request Types* on page 170 for details.
Request Status	Based on the status of the request itself (linked to the workflow step), the field can be set to display or be hidden. See *Configuring Request Statuses for Request Types* on page 189 for details.
Field security	Fields can also be configured to be invisible to particular users or security groups. This is controlled from the Field window's **Security** tab. See *Creating Fields for Request Types* on page 170 for details.

Figure 5-3 on page 168 illustrates how the product determines whether a field is visible to 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.

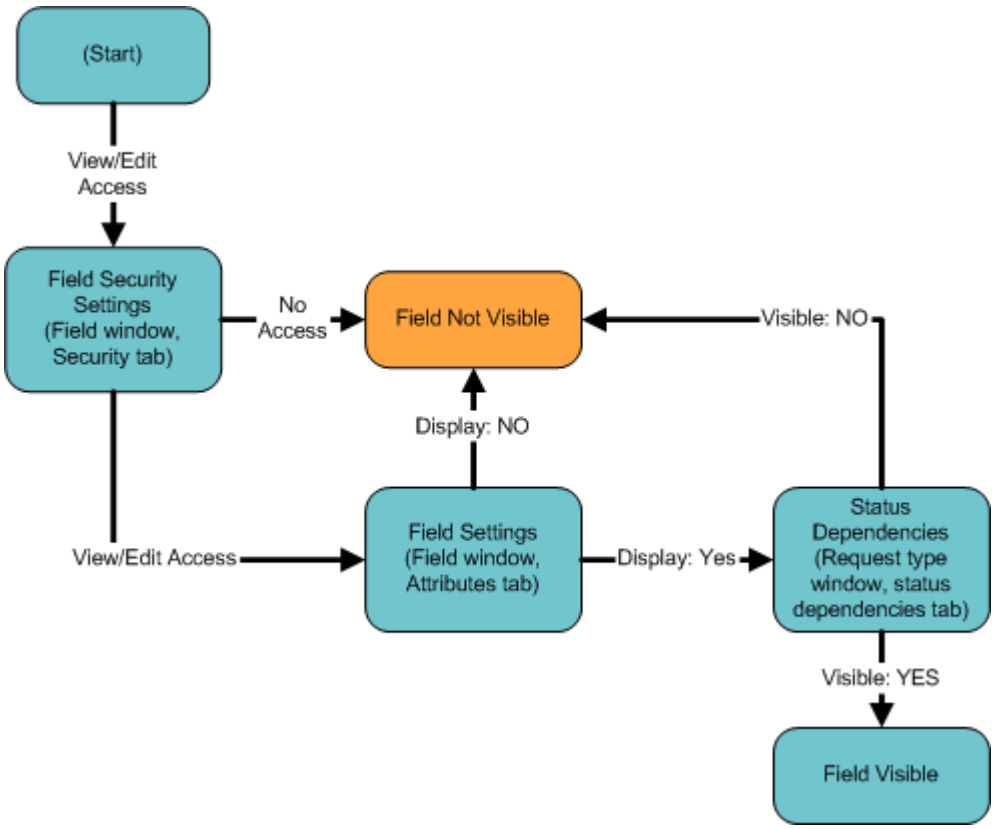


Figure 5-3. Field visibility interactions

Criteria for Editable Fields

Fields can be configured to become display-only. This makes the contents non-editable (see *Table 5-2*).

Table 5-2. Criteria for editable fields

Criteria	Description
Request Status	Based on the status of the request itself, the field can be set to become non-editable. See *Configuring Request Statuses for Request Types* on page 189 for details.
Field security	Fields can be configured to be visible but non-editable to particular users or security groups. This is controlled from the Field window's **Security** tab. See *Creating Fields for Request Types* on page 170 for details.

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 170 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 198 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

Since field behavior is often dependent 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. Open the Request Type Workbench.

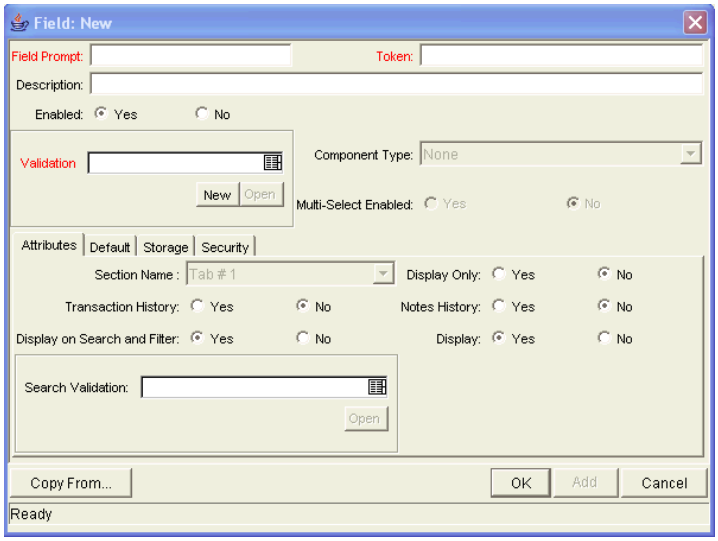
   To open the Request Type Workbench, see *Opening the Request Type Workbench* on page 161. The Request Type Workbench window opens.

2. Open a request type.

   The Request Type window opens.

3. In the Request Type window, in the **Fields** tab, click **New.**

   The Fields window opens. The **Attributes** tab is displayed.

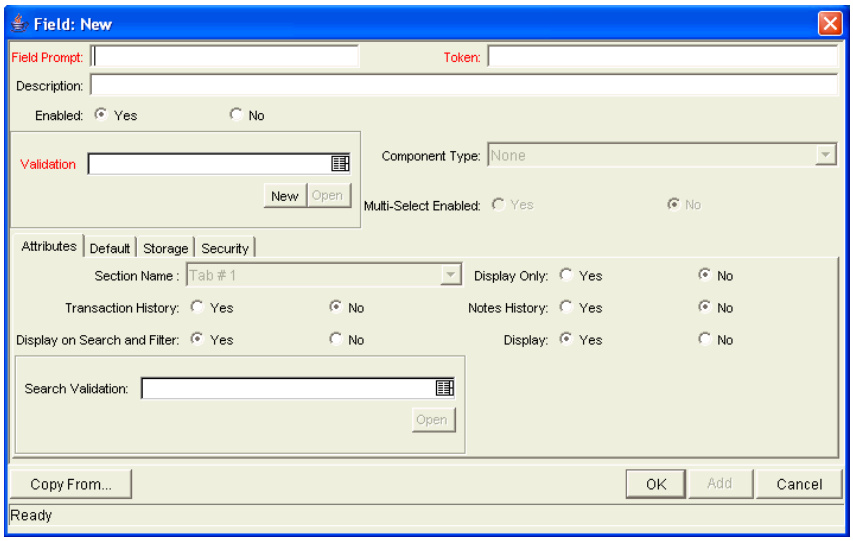


4. Complete the fields in the Fields window as specified in the following table:

Field	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 167 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 autocomplete component for the validation.

5. In the Fields window, click the **Attributes** tab

   The **Attributes** tab opens.



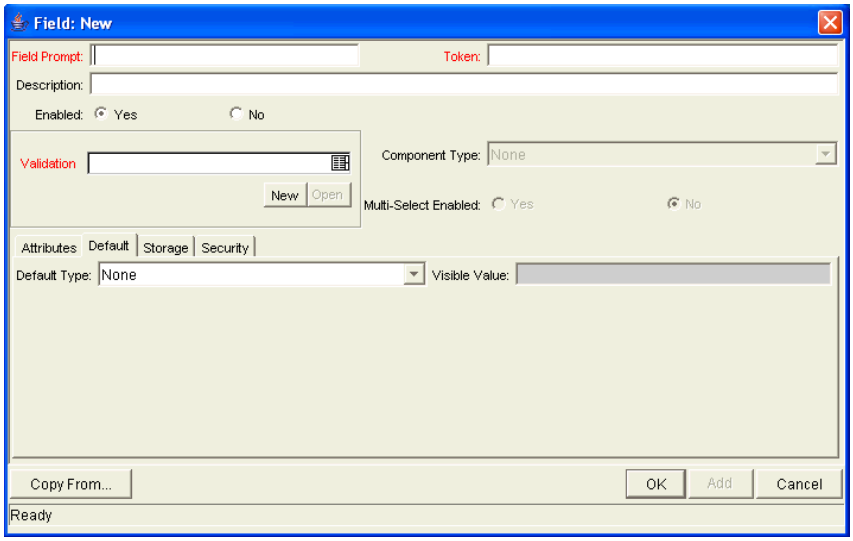
6. Complete the fields in the **Attributes** tab as specified in the following table:

Field	Description
Section Name	The area 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 will be logged in notes for the request.
Display On Search and Filter	Indicates whether or not the field will be 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 will no longer be 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 should not exceed forty fields.

7. In the Fields window, click the **Defaults** tab

   The **Defaults** tab opens.



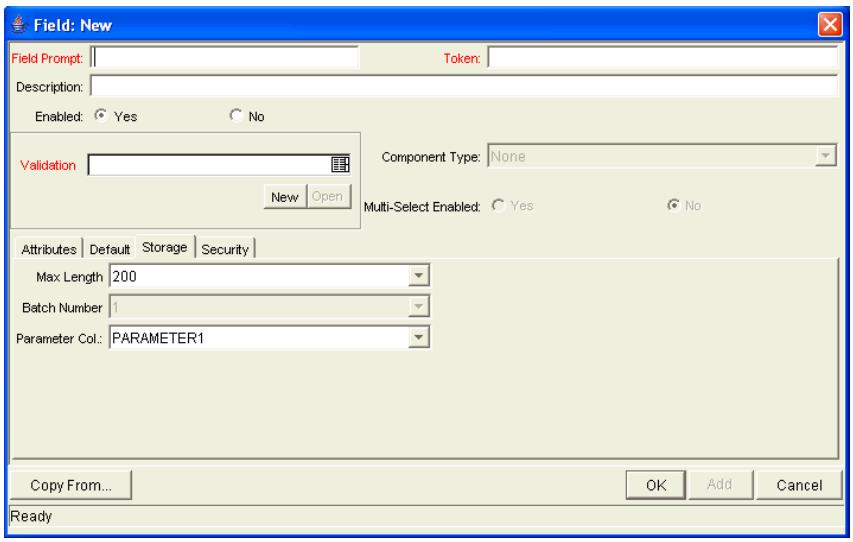
8. Complete the fields in the **Defaults** tab as specified in the following table:

Field	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, the constant value can be entered here. This value should be what the user would normally enter in the field.
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. In the Fields window, click the **Storage** tab

   The **Storage** tab opens. 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, the **Storage** tab can also be used 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 1800 in column 41 or higher.

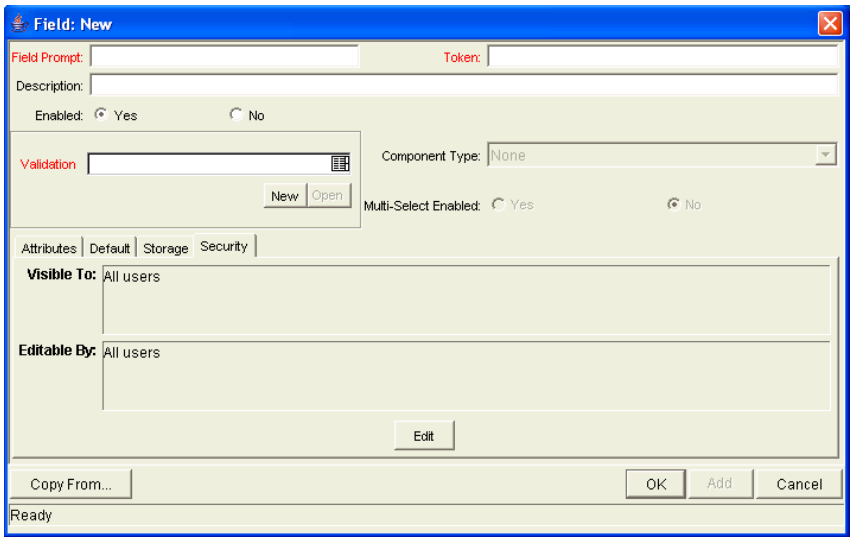


10. Complete the fields in the **Storage** tab as specified in the following table:

Field	Description
Max Length	Indicates the maximum field character length. The two possible values are 200 and 1800.
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. In the Fields window, click the **Security** tab.

The **Security** tab opens.



12. Complete the fields in the **Security** tab as specified in the following table:

Field	Description
Visible To	Lists all users, security groups, and linked tokens for which this field will be visible.
Editable By	Lists all users, security groups, and linked tokens for which this field will be editable.
Edit	Opens the Edit Field Security window, which configures the users, security groups, and linked tokens that will be able to view and/or edit this field. See *Creating Fields for Request Types* on page 170 for more detailed information.

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

The Edit Field Security window opens.

b. Deselect the Visible to all users box to begin fine tuning field properties.

c. Make a choice from the Select Users/Security Groups that can view this field drop-down list. Possible choices include **User, Security Group, Standard Token,** or **User Defined Token.**

d. After selecting a choice from the drop-down list, select the user, security group, or token from the autocomplete list.

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

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 area.

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 area. To remove viewing rights entirely, select the user, security group, or token and click **Remove.**

f. When you are finished adding users, security groups, or tokens to the This field is visible to these Users/Security Groups area, click **OK.**

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

13. At the bottom of the Field window, 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.

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) will be 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. Open the Request Type Workbench.

   To open the Request Type Workbench, see *Opening the Request Type Workbench* on page 161. The Request Type Workbench window opens.

2. Open a request type.

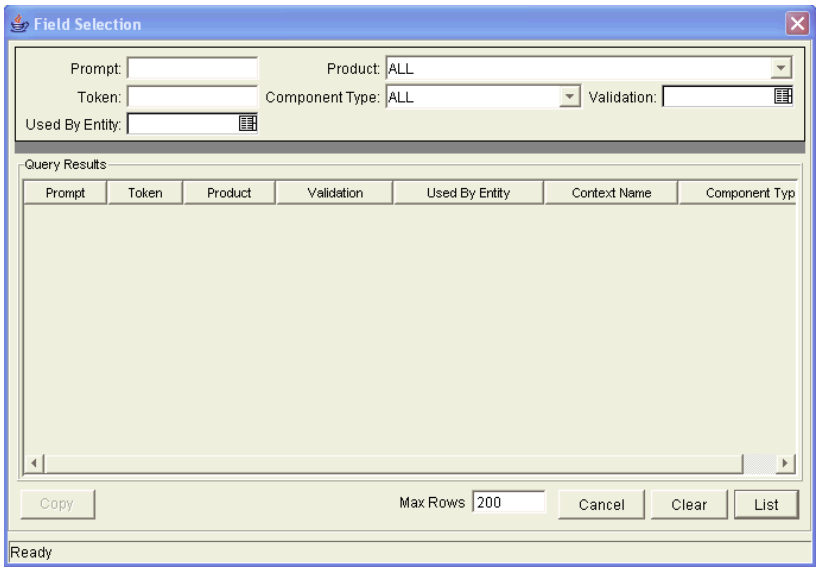
   The Request Type window opens.

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

   The Field window opens.

4. In the Field window, click **Copy From.**

   The Field Selection window opens.



5. Enter search criteria into the header fields, such as the token name or field prompt.

   More complex queries can also be performed, such as listing all fields that reference a certain validation or are used by a certain entity. Due to the large number of fields in the system, limit the list of fields by one or more of the query criteria.

6. In the Field Selection window, click **List.**

   The results are listed in the Query Results section.

7. Select the desired field, and click **Copy.**

   The Field Selection window copies the definition of the selected field into the New Field window. The Field Selection window closes.

8. In the New Field window, make any necessary modifications.

9. At the bottom of the New Field window, click **OK.**

   The changes to the request type are saved.

Removing Fields for Request Types

To remove a field from a request type:

1. Open the Request Type Workbench.

   To open the Request Type Workbench, see *Opening the Request Type Workbench* on page 161. The Request Type Workbench window opens.

2. Open a request type.

   The Request Type window opens.

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

   The field is removed.

4. In the **Field** tab, click **OK.**

   The changes to the request type are saved.

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. Open the Request Type Workbench.

   To open the Request Type Workbench, see *Opening the Request Type Workbench* on page 161. The Request Type Workbench window opens.

2. Open a request type.

   The Request Type window opens.

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

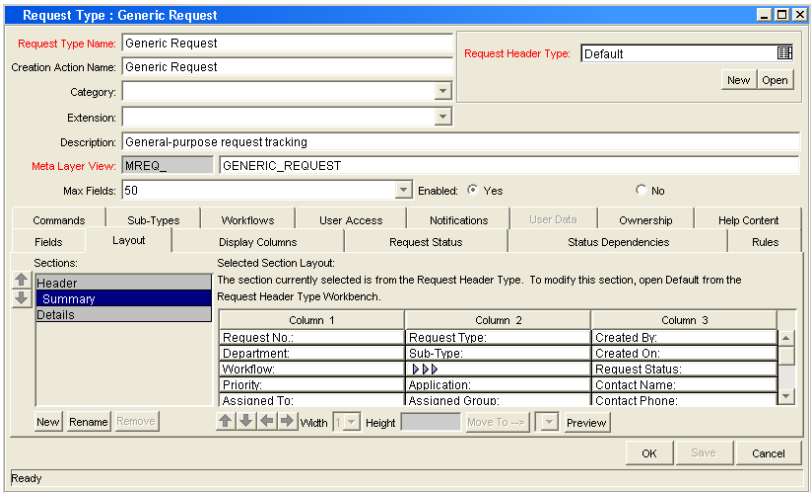
   The **Layout** tab opens.

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

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

6. At the bottom of the **Layout** tab, select a width from the Width drop-down list.

   Fields can have a width of **1, 2,** or **3.** The field's width must correspond to the column location. For example, a field located in Column 2 cannot have a width set to **3.** Additionally, for fields of component type Text Area, it is possible to 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 will be blank and non-updateable.



7. In the **Layout** tab, select **OK.**

   The changes to the request type are saved.

Moving Fields On Request Types

To move a field or a set of fields:

1. Open the Request Type Workbench.

   To open the Request Type Workbench, see *Opening the Request Type Workbench* on page 161. The Request Type Workbench window opens.

2. Open a request type.

   The Request Type window opens.

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

   The **Layout** tab opens.

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

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

6. At the bottom of the **Layout** tab, move the fields to the desired location in the layout builder, either by clicking the **Arrow** icons or using the corresponding keyboard arrow keys. If a request type has multiple sections in its field layout, fields can be moved from one section to another. To move a field to a different section:

   a. Select the field.

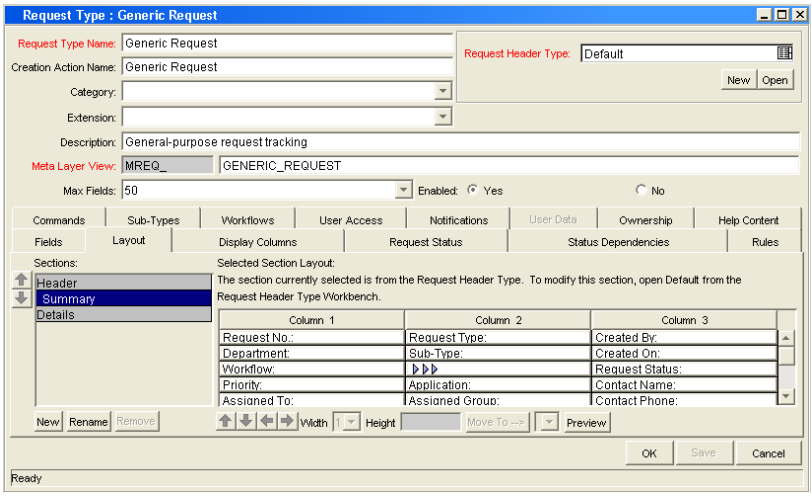
      You cannot choose a request header type field.

   b. In the fields next to the **Move To -->** button, select a section from the drop-down list.

      You cannot choose a request header type section.

   c. Click **Move To -->.**

      The field is moved to the listed section.



7. In the **Layout** tab, select **OK.**

   The changes to the request type are saved.

Adding Sections to Request Types

To add a new section to a request type:

1. Open the Request Type Workbench.

   To open the Request Type Workbench, see *Opening the Request Type Workbench* on page 161. The Request Type Workbench window opens.

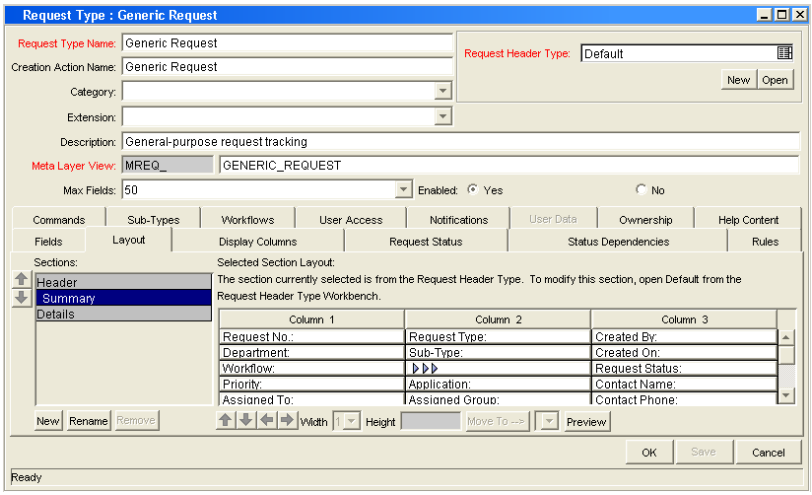
2. Open a request type.

   The Request Type window opens.

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

   The **Layout** tab opens.

4. In the **Layout** tab, at the bottom of the Sections area, click **New.**



   The Input window opens.

5. In the Input window, enter a new section name.

   Custom section names can be up to 30 characters in length.

   When requests are generated for the given request type, the new section with the defined custom fields will be visible.

6. To view what the layout will look like to the user processing the request, click **Preview.** This opens an HTML window that shows the fields as they will appear.

   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 area when a request of the given request type is viewed or edited.

   Any non-displayed fields do not affect the layout. The layout engine considers them the same as a blank field.

7. In the **Layout** tab, select **OK.**

   The changes to the request type are saved.

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. Open the Request Type Workbench.

   To open the Request Type Workbench, see *Opening the Request Type Workbench* on page 161. The Request Type Workbench window opens.

2. Open a request type.

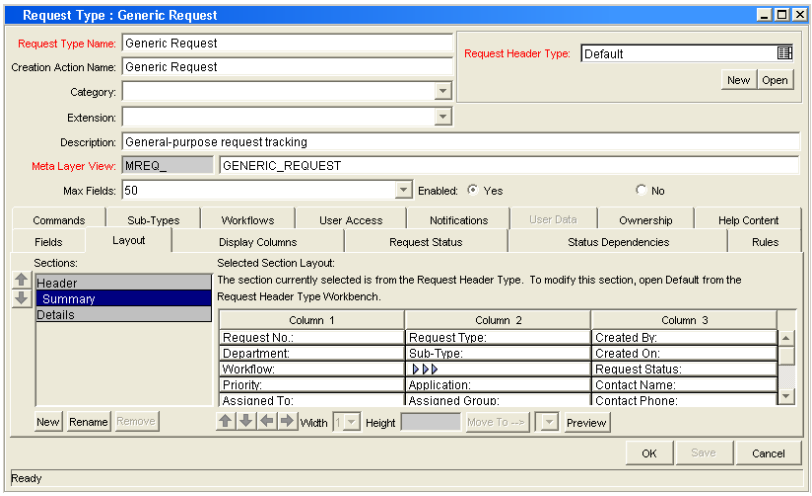
   The Request Type window opens.

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

   The **Layout** tab opens.

4. In the **Layout** tab, in the Sections area, select a Section.

5. In the **Layout** tab, at the bottom of the Sections area, click **Rename.**



The Input window opens.

6. In the Input window, enter a new section name.

   Custom section names can be up to 30 characters in length.

   When requests are generated for the given request type, the new section with the defined custom fields will be visible.

7. To view what the layout will look like to the user processing the request, click **Preview.** This opens an HTML window that shows the fields as they will appear.

   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 area when a request of the given request type is viewed or edited.

   Any non-displayed fields do not affect the layout. The layout engine considers them the same as a blank field.

8. In the **Layout** tab, select **OK.**

   The changes to the request type are saved.

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. Open the Request Type Workbench.

   To open the Request Type Workbench, see *Opening the Request Type Workbench* on page 161. The Request Type Workbench window opens.

2. Open a request type.

   The Request Type window opens.

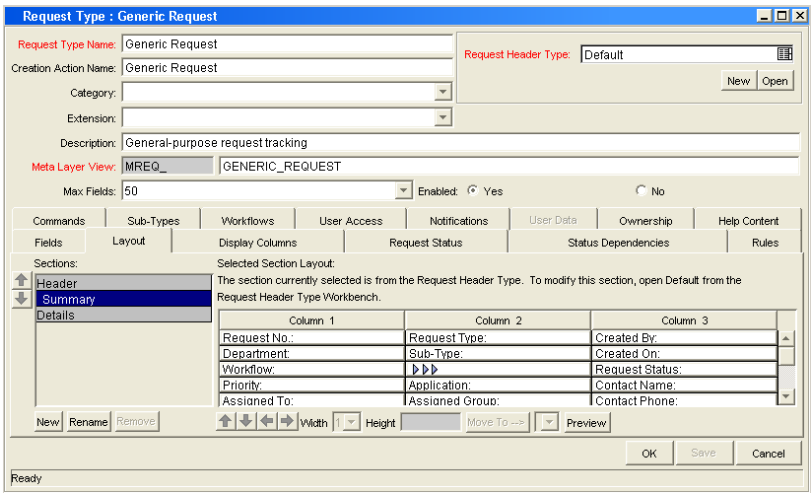
3. In the Request Type window, click the **Layout** tab.

   The **Layout** tab opens.

4. In the **Layout** tab, in the Sections area, select a Section.

5. In the **Layout** tab, at the bottom of the Sections area, click **Remove.**

   The section is removed.



6. In the **Layout** tab, select **OK.**

   The changes to the request type are saved.

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.

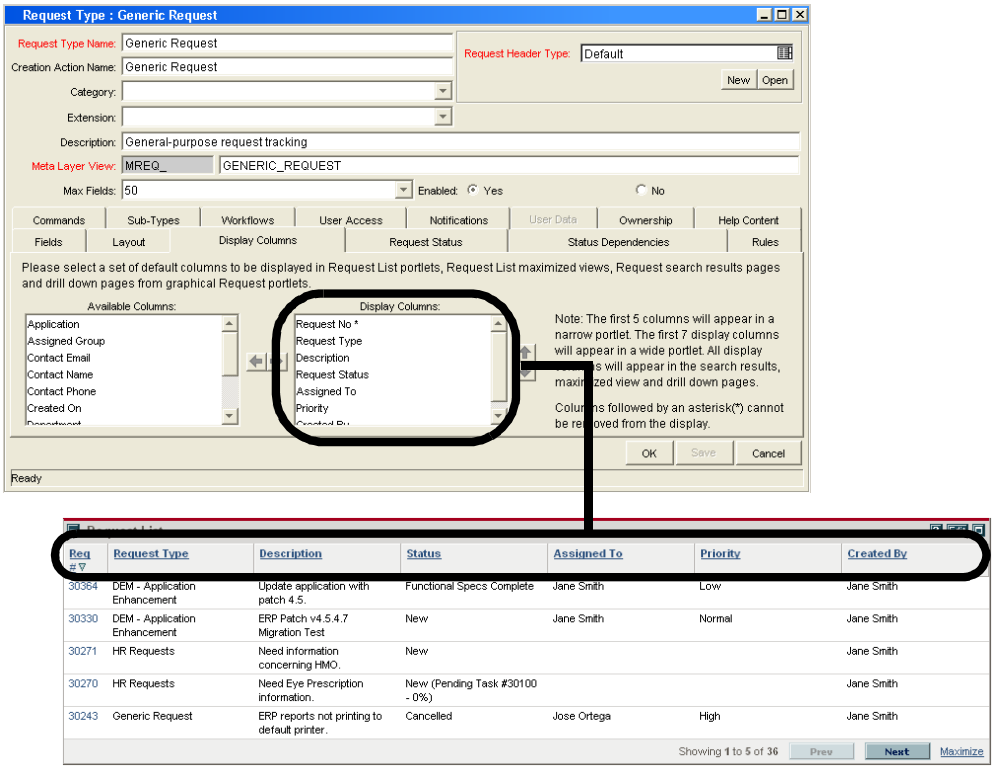


Figure 5-4. Displayed columns set in the request type window

To configure the columns that are displayed in list portlets:

1. Open the Request Type Workbench.

   To open the Request Type Workbench, see *Opening the Request Type Workbench* on page 161. The Request Type Workbench window opens.

2. Open a request type.

   The Request Type window opens.

3. In the Request Type window, click the **Display Columns** tab.

   The **Display Columns** tab opens.

4. In the Available Columns list, select the columns to display.

   Select multiple columns by pressing the Ctrl key while clicking different items.

5. Click the **Right Arrow** icon.

   The selected items are moved into the Display Columns list.

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

7. In the **Display Columns** tab, click **OK.**

   The changes to the request type are saved.

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* on page 190 shows how statuses are linked to workflow steps.

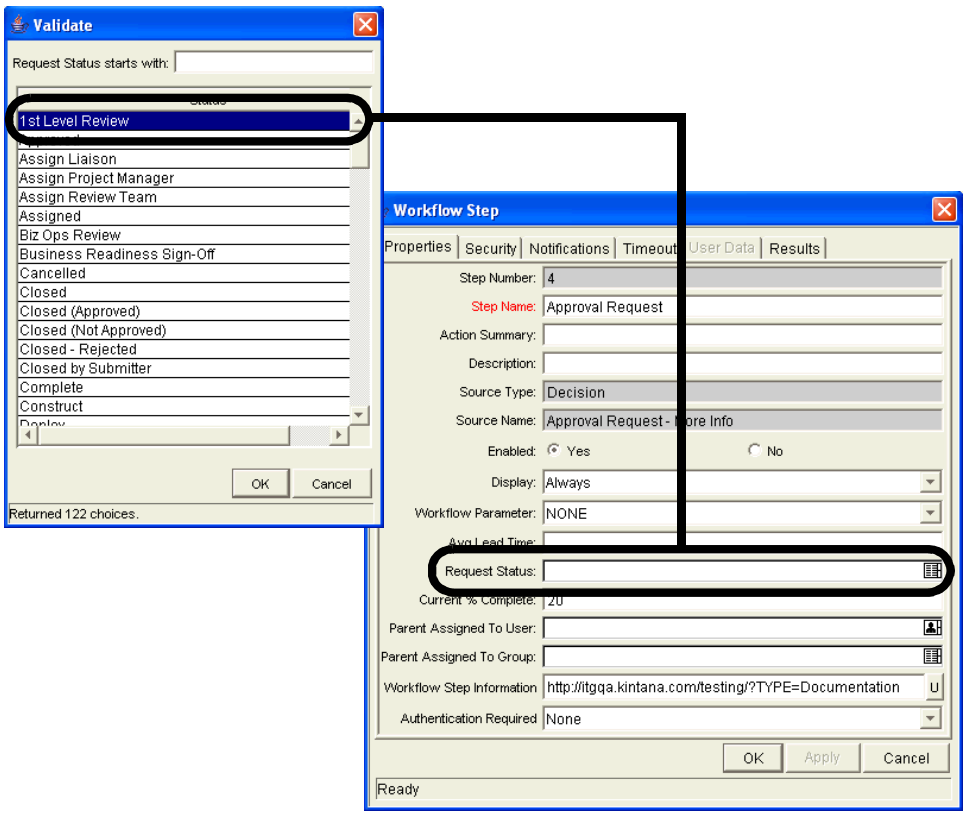


Figure 5-5. Request status specified in the workflow step window

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 194.

Before linking request statuses to workflow steps, the request type must first possess all desired statuses. The list of possible request statuses is created in the **Request Status** tab. *Figure 5-6* on page 191 shows the interface for creating request statuses.

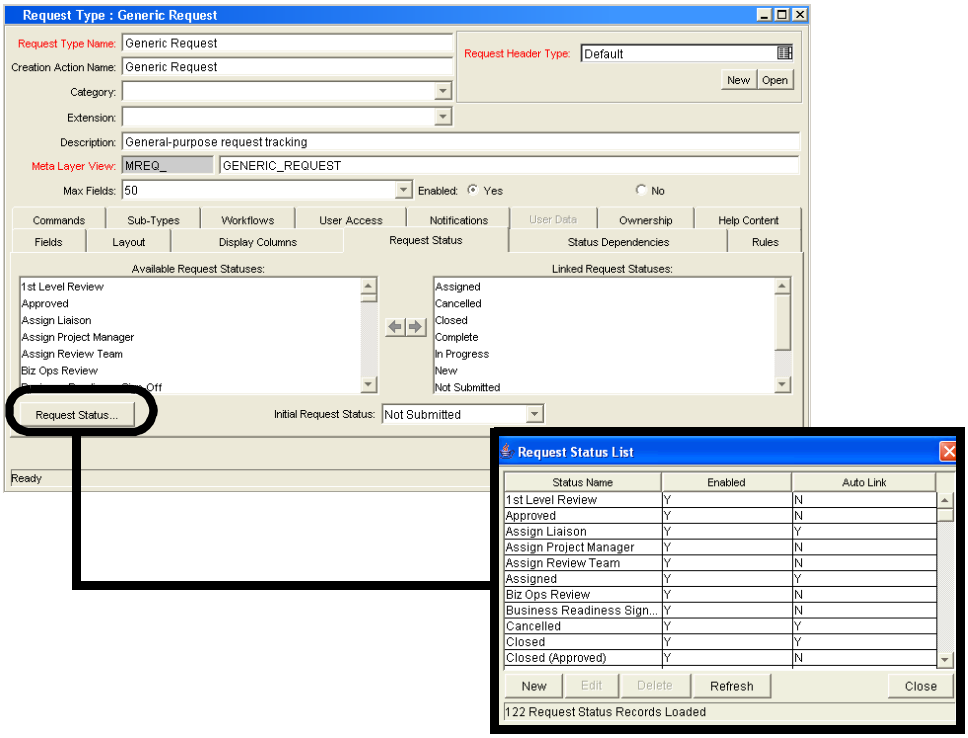


Figure 5-6. Request Status tab and Request Status List window

If a desired status does not appear in the Available Request Statuses list, it can be created. A request's initial status can be set using the Initial Request Status drop-down list.

Creating Request Statuses for Request Types

To create a new request status:

1. Open the Request Type Workbench.

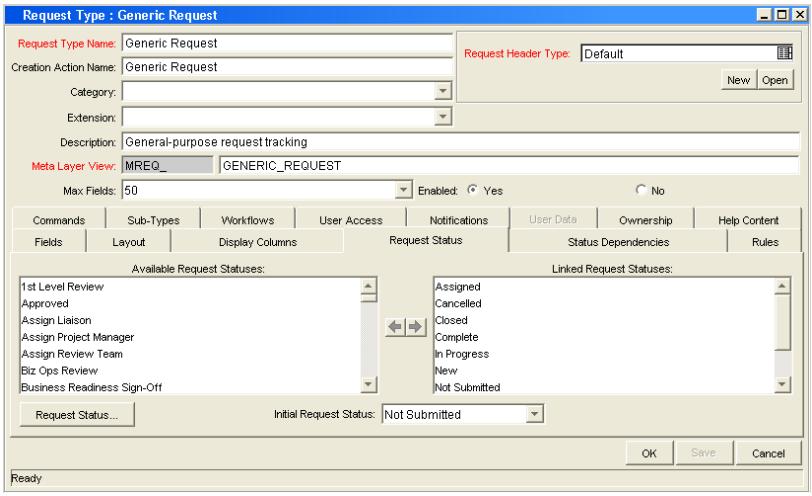
   To open the Request Type Workbench, see *Opening the Request Type Workbench* on page 161. The Request Type Workbench window opens.

2. Open a request type.

   The Request Type window opens.

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

   The **Request Status** tab opens.

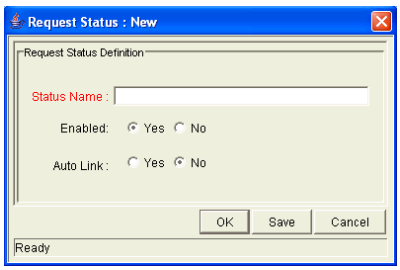


4. In the **Request Status** tab, click **Request Status.**

   The Request Status List window opens.

5. In the Request Status List window, click **New.**

   The Request Status: New window opens.



6. Complete the fields in the Request Status: New window as specified in the following table.

Field	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. In the Request Status: New window, click **OK**.

   The Request Status: New window closes. The Request Status List window opens.

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

   The Request Status List window closes. The **Request Status** tab is appears.

9. In the **Request Status** tab, click **OK.**

   The changes to the request type are saved.

Configuring Status Dependencies

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

For example, a request is not to be allowed to reach the **Assigned** request status unless the Assigned To User field has a value. Additionally, if a request is at a status of **Assigned**, a user cannot clear the Assigned To User field.

In order to make this work, the Assigned To User 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 the request's status:

1. Open the Request Type Workbench.

   To open the Request Type Workbench, see *Opening the Request Type Workbench* on page 161. The Request Type Workbench window opens.

2. Open a request type.

   The Request Type window opens.

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

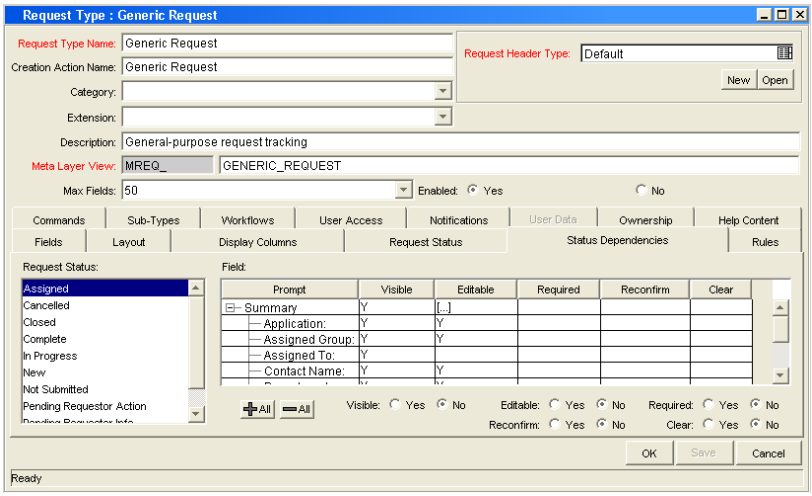
   The **Status Dependencies** tab opens.

4. In the Request Status section, select a request status.

5. In the Field section, select a field.

6. Complete the fields in the Field section as specified in the following table.

   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.



Field	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.

Field	Description
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.

7. In the **Status Dependencies** tab, click **OK.**

   The changes to the request type are added.

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	not applicable	not applicable
No	No	Yes	No	not applicable	not applicable
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.

There are two types of rules:

- **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 filled in by the user. 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 drop-down list to switch between **Simple** and **Advanced Defaults.** However, when switching between rule types, whatever work has been done in the first type will be 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 desired workflow and the rule event to **Apply On Creation,** it is possible to set the default workflow that will be used each time a request of that type is used.

To add a simple default rule to a request type:

1. Open the Request Type Workbench.

   To open the Request Type Workbench, see *Opening the Request Type Workbench* on page 161. The Request Type Workbench window opens.

2. Open a request type.

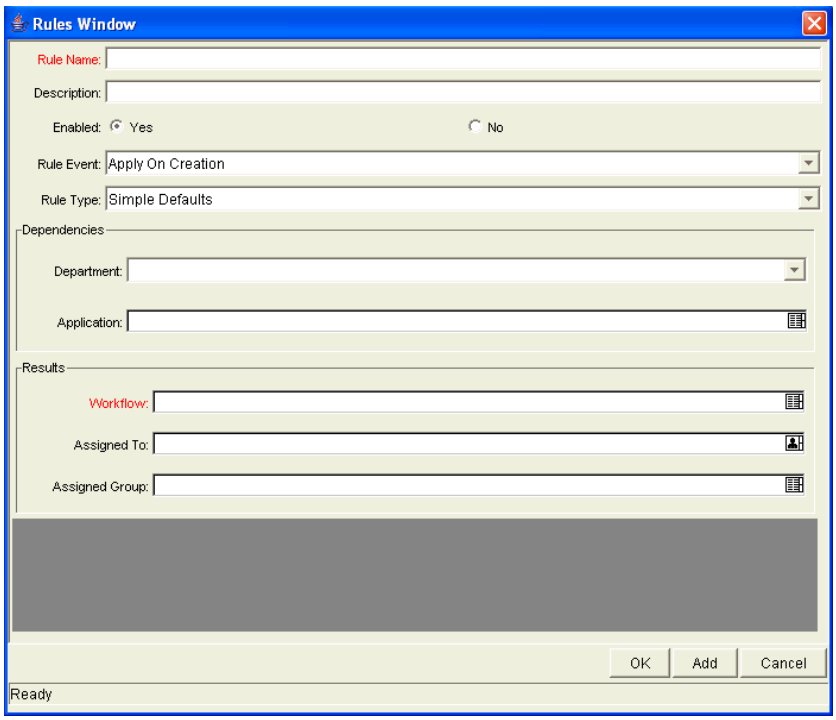
   The Request Type window opens.

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

   The **Rules** tab opens.

4. In the **Rules** tab, click **New.**

   The Request Type Rules window opens in **Simple Defaults** mode.



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

Field	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 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 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.**
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.

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

The changes made to the request type are saved. 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 specified in the rule.

If more than one rule applies for a given request, then the system uses a more specific rule. See *Creating Advanced Default Rules for Request Types* on page 201 for more detailed information.

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.

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, filling in whichever of the Results fields have been specified.
- **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 area 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 area.

When the field or fields specified in the Dependencies area are changed, any fields specified in the Results area 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 area is changed to a user-defined value. When the field is changed, any fields specified in the Results area will be automatically populated according to the first rule defined. Any other rule processing will stop 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. Open the Request Type Workbench.

   To open the Request Type Workbench, see *Opening the Request Type Workbench* on page 161. The Request Type Workbench window opens.

2. Open a request type.

   The Request Type window opens.

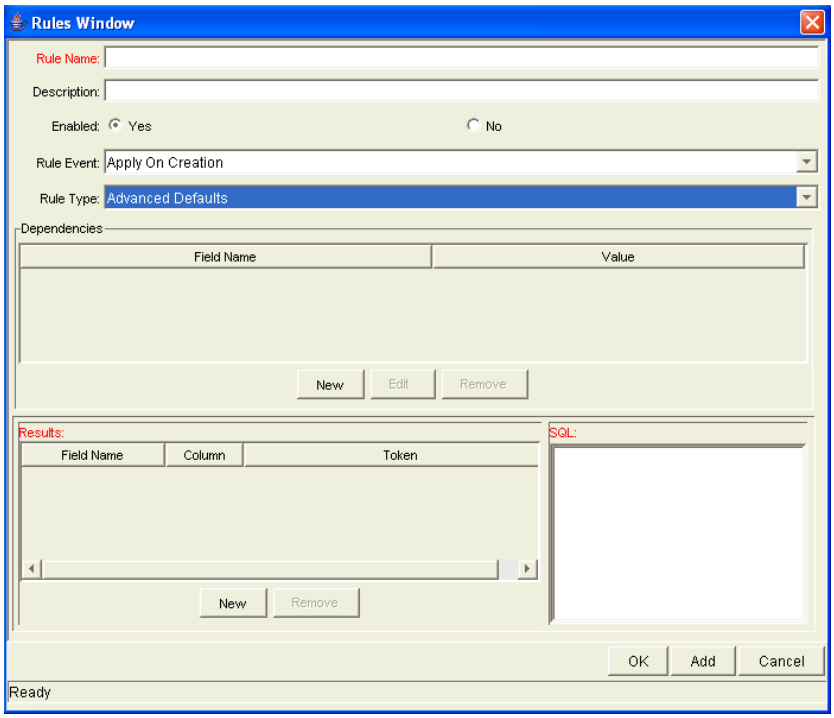
3. In the Request Type window, click the **Rules** tab.

   The **Rules** tab opens.

4. In the **Rules** tab, click **New.**

   The Request Type Rules window opens in **Simple Defaults** mode.

5. The Request Type Rules window, from the Rule Type drop-down list, select **Advanced Defaults.**



6. Complete the fields in the Rules window as specified in the following table:

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

Field	Description
Field	Selects the field from the autocomplete list. Request default rules cannot be configured to trigger from a multiple select autocomplete field. Do not choose a multiple select autocomplete 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	Description
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. In the Dependencies window, click **OK.**

   The field is added the field to the Dependencies area. The Dependencies window closes.

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

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

11. Complete the fields in the Results window as specified in the following table:

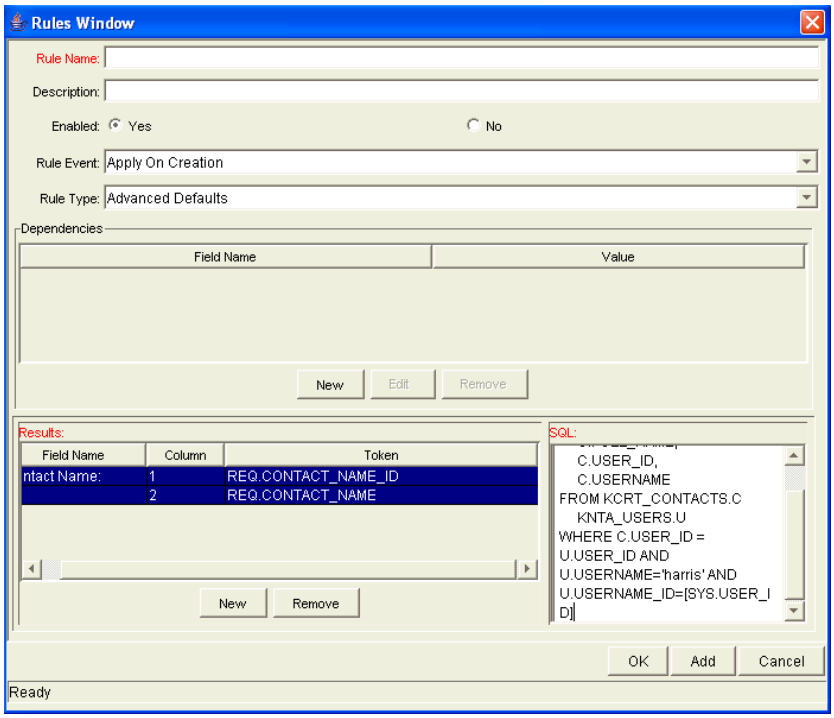
Field	Description
Field	Selects the field from the autocomplete list.
Field Type	The type of field selected, such as request header type. This field is filled automatically.
Validation Name	The field's type of validation, such as Numeric Text Field - 2 decimals. This field is filled automatically.
Visible 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. In the Results window, click **OK.**

    The field is added the field to the Results area. The Results window closes.

13. In the SQL area, define the SQL statement that will load values into the fields specified in the Results area.

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



14. In the Rules window, click **OK.**

    The Rules window closes.

15. In the **Rules** tab, click **Save.**

    The changes to the request type are saved.

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 211.

Adding Commands to Request Types

To add commands to request types:

1. Open the Request Type Workbench.

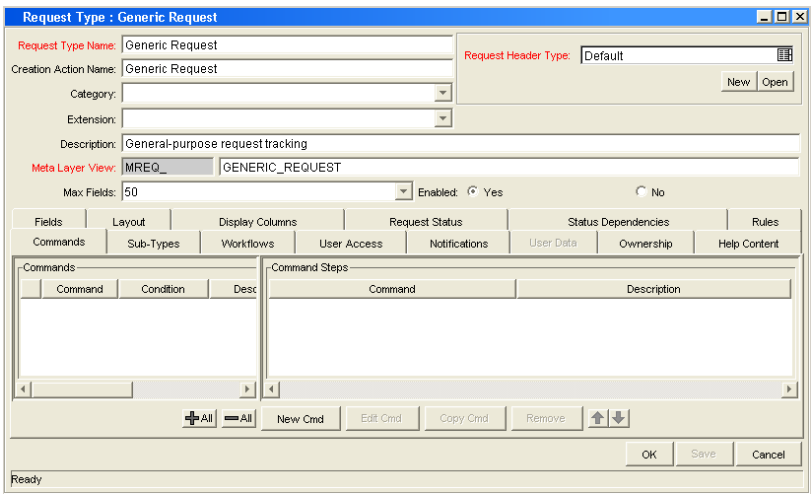
   To open the Request Type Workbench, see *Opening the Request Type Workbench* on page 161. The Request Type Workbench window opens.

2. Open a request type.

   The Request Type window opens.

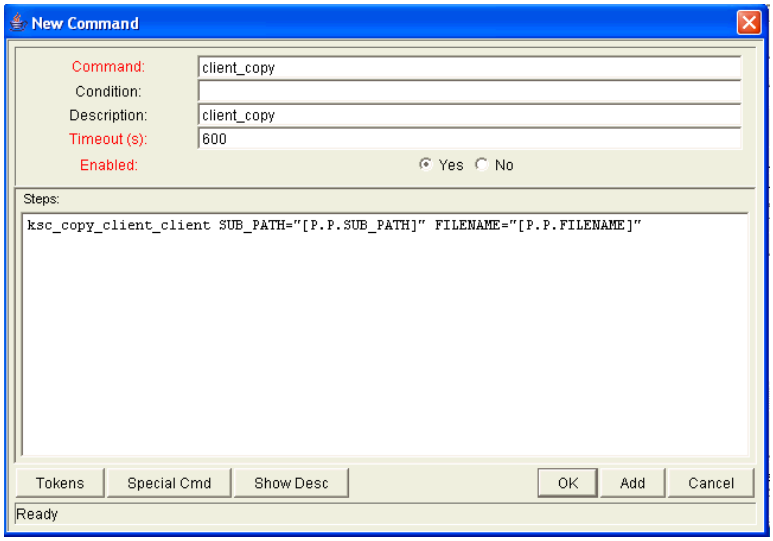
3. In the Request Type window, click the **Commands** tab.

   The **Commands** tab opens.



4. In the **Commands** tab, click **New Cmd.**

   The New Command window opens.



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

Field	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 211 for more information).
Description	A description of the command.
Timeout	The amount of time the command will be allowed to run before its process is terminated. This mechanism is used to abort commands that are hanging or taking an abnormal amount of time.
Enabled?	Indicates whether the command is enabled for execution.

6. In the New Command window, click **OK.**

   The New Command window closes. The **Commands** tab lists the new command.

7. In the **Commands** tab, click **OK.**

   The changes to the request type are added.

Editing Commands of Request Types

To edit a command on a request type:

1. Open the Request Type Workbench.

   To open the Request Type Workbench, see *Opening the Request Type Workbench* on page 161. The Request Type Workbench window opens.

2. Open a request type.

   The Request Type window opens.

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

   The **Commands** tab opens.

4. In the **Commands** tab, click **Edit Cmd.**

   The Edit Command window opens.

5. Select the command to edit.

6. Complete the fields in the Edit Command window as specified in the following table:

Field	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 211 for more information).
Description	A description of the command.
Timeout	The amount of time the command will be allowed to run before its process is terminated. This mechanism is used to abort commands that are hanging or taking an abnormal amount of time.
Enabled?	Indicates whether the command is enabled for execution.

7. In the Edit Command window, click **OK.**

   The Edit Command window closes. The **Commands** tab lists the edited command.

8. In the **Commands** tab, click **OK.**

   The changes to the request type are added.

Copying Commands in Request Types

To copy a command in a request types:

1. Open the Request Type Workbench.

   To open the Request Type Workbench, see *Opening the Request Type Workbench* on page 161. The Request Type Workbench window opens.

2. Open a request type.

   The Request Type window opens.

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

   The **Commands** tab opens.

4. Select the command to copy.

5. In the **Commands** tab, click **Copy Cmd.**

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

6. In the **Commands** tab, click **OK.**

   The changes to the request type are added.

Deleting Commands in Request Types

To copy a command in a request types:

1. Open the Request Type Workbench.

   To open the Request Type Workbench, see *Opening the Request Type Workbench* on page 161. The Request Type Workbench window opens.

2. Open a request type.

   The Request Type window opens.

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

   The **Commands** tab opens.

4. Select the command to delete.

5. In the **Commands** tab, click **Remove.**

   The command is deleted.

6. In the **Commands** tab, click **OK.**

   The changes to the request type are added.

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

Condition	Evaluates to
BLANK	Command will be executed in all situations.
'[P.P_VERSION_LABEL]' IS NOT NULL	Command will be executed if the parameter with the token P_VERSION_LABEL in the package line is not null.
'[DEST_ENV.ENVIRONMENT_NAME]' = 'Archive'	Command will be executed when the destination environment is named "Archive."
'[AS.SERVER_TYPE_CODE]' = 'UNIX'	Command will be 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 *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. Open the Request Type Workbench.

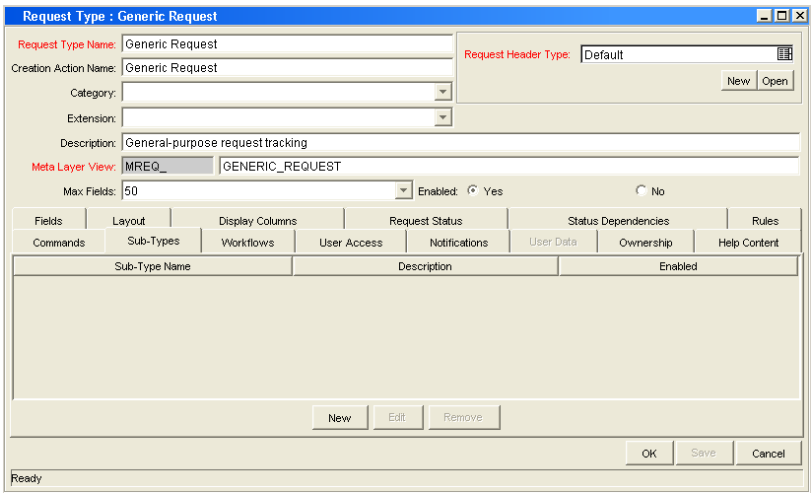
   To open the Request Type Workbench, see *Opening the Request Type Workbench* on page 161. The Request Type Workbench window opens.

2. Open a request type.

   The Request Type window opens.

3. In the Request Type window, click the **Sub-Types** tab.

   The **Sub-Types** tab opens.



4. In the **Sub-Types** tab, click **New.**

   The Request Sub-Type window opens.

5. Complete the fields in the Request Sub-Type window as specified in the following table:

Field	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. In the Request Sub-Type window, click **OK.**
7. The Request Sub-Type window closes.
8. In the **Sub-Types** tab, click **OK.**

   The changes to the request type are saved.

Editing Sub-Types for Request Types

To edit a sub-type:

1. Open the Request Type Workbench.

   To open the Request Type Workbench, see *Opening the Request Type Workbench* on page 161. The Request Type Workbench window opens.

2. Open a request type.

   The Request Type window opens.

3. In the Request Type window, click the **Sub-Types** tab.

   The **Sub-Types** tab opens.

4. In the **Sub-Types** tab, select a sub-type and click **Edit.**

   The Request Sub-Type window opens.

5. Complete the fields in the Request Sub-Type window as specified in the following table:

Field	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. In the Request Sub-Type window, click **OK.**

   The Request Sub-Type window closes.

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

   The changes to the request type are saved.

Deleting Sub-Types from Request Types

To delete sub-types from a request type:

1. Open the Request Type Workbench.

   To open the Request Type Workbench, see *Opening the Request Type Workbench* on page 161. The Request Type Workbench window opens.

2. Open a request type.

   The Request Type window opens.

3. In the Request Type window, click the **Sub-Types** tab.

   The **Sub-Types** tab opens.

4. In the **Sub-Types** tab, select a sub-type and click **Remove.**

   The sub-type is removed.

5. In the **Sub-Types** tab, click **OK.**

   The changes to the request type are saved.

Configuring Request Types to Work with Workflows

Request types can be configured to work with all workflows or only selected workflows.

Adding Workflows to Request Types

To add workflows to the request type:

1. Open the Request Type Workbench.

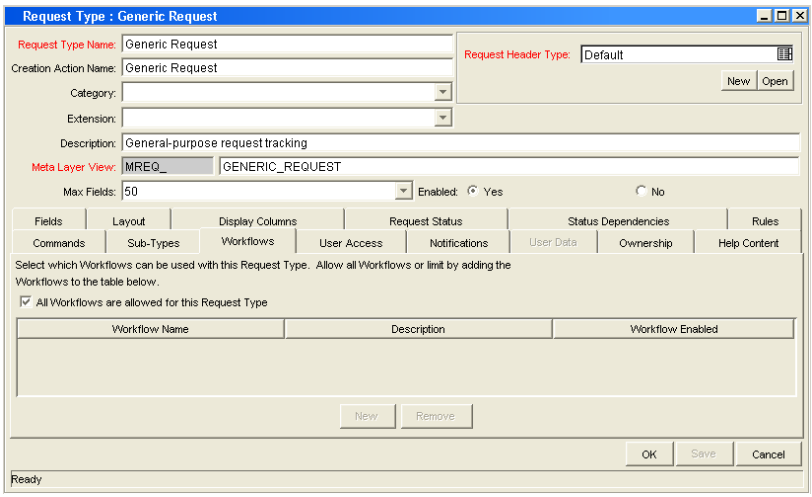
   To open the Request Type Workbench, see *Opening the Request Type Workbench* on page 161. The Request Type Workbench window opens.

2. Open a request type.

   The Request Type window opens.

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

   The **Workflows** tab opens.



4. Allow all workflows or only selected workflows.

   Selecting All Workflows are allowed for the Request Type allows all workflows to use this request type. Deselecting All Workflows are allowed for the Request Type requires specific workflows to be allowed to use this request type.

To select specific workflows for the request type:

a. Deselect All Workflows are allowed for the Request Type.

b. In the **Workflows** tab, click **New.**

The Workflow window opens.

c. In the Workflow window, in the Workflow field, select one workflow from the drop-down list and click **OK.**

The workflow is added to the Workflow window and the Workflow window closes. To add a workflow and keep the Workflow window open, click **Add.**

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

The changes to the request type are saved.

Deleting Workflows from Request Types

To delete workflows from the request type:

1. Open the Request Type Workbench.

To open the Request Type Workbench, see *Opening the Request Type Workbench* on page 161. The Request Type Workbench window opens.

2. Open a request type.

The Request Type window opens.

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

The **Workflows** tab opens.

4. In the **Workflows** tab, select a workflow to delete and click **Remove.**

The workflow is deleted.

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

The changes to the request type are saved.

Configuring Participants for Request Types

Users can be given different levels of access (use) of request types.

Adding Participants to Request Types

To add participants to the request type:

1. Open the Request Type Workbench.

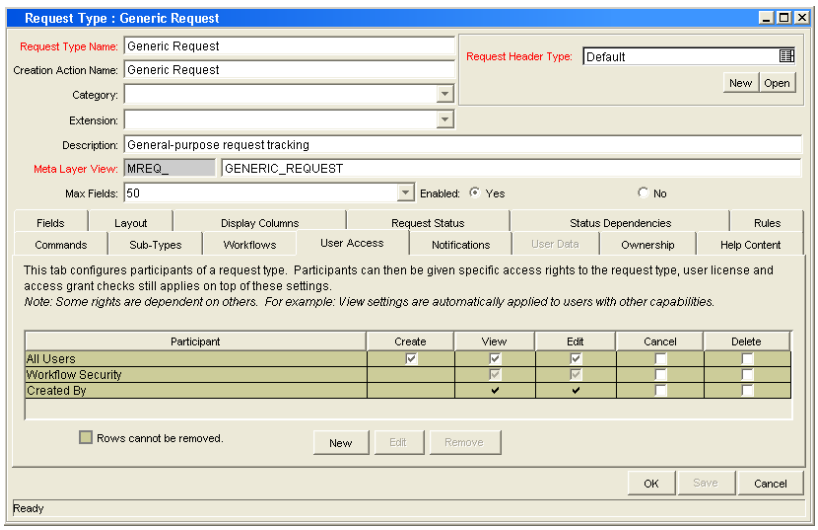
   To open the Request Type Workbench, see *Opening the Request Type Workbench* on page 161. The Request Type Workbench window opens.

2. Open a request type.

   The Request Type window opens.

3. In the Request Type window, click the **User Access** tab.

   The **User Access** tab opens.



4. In the **User Access** tab, click **New.**

   The Participant Security window opens.

5. In the Participant Security window, select the security type from the drop-down list.

   The security type options are:

   - **Enter a Security Group Name.** Select a security group to act upon the workflow step. Selecting a security group changes the name of the autocomplete 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 autocomplete field 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 autocomplete field to Standard Token. The security type is left undefined. Select a standard token from the autocomplete field. 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 autocomplete field to User Defined Token. The security type is dynamically changed to a drop-down 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.

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

6. Add the attributes for the participant.

   Attributes are attached to a participant by selecting 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. Open the Request Type Workbench.

   To open the Request Type Workbench, see *Opening the Request Type Workbench* on page 161. The Request Type Workbench window opens.

2. Open a request type.

   The Request Type window opens.

3. In the Request Type window, click the **User Access** tab.

   The **User Access** tab opens.

4. In the **User Access** tab, select a participate to edit and click **Edit.**

   The Participant Security window opens.

5. Edit the attributes for the participant.

   Attributes are attached to a participant by selecting 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. Open the Request Type Workbench.

   To open the Request Type Workbench, see *Opening the Request Type Workbench* on page 161. The Request Type Workbench window opens.

2. Open a request type.

   The Request Type window opens.

3. In the Request Type window, click the **User Access** tab.

   The **User Access** tab opens.

4. In the **User Access** tab, select a participant to delete and click **Remove.**

   The participant is deleted.

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

   The changes to the request type are saved.

Configuring Notifications for Request Types

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

Adding Notifications

To add a notification:

1. Open the Request Type Workbench.

   To open the Request Type Workbench, see *Opening the Request Type Workbench* on page 161. The Request Type Workbench window opens.

2. Open a request type.

   The Request Type window opens.

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

   The **Notifications** tab opens.

4. In the **Notifications** tab, click **New.**

   The Add Notification for Step window opens. The Add Notification for Step window has two tabs: **Setup** and **Message.**

5. Configure the **Setup** tab.

   For information on how to configure the **Setup** tab, see *Configuring Setup Tabs* on page 222.

6. Configure the **Message** tab.

   For information on how to configure the **Message** tab, see *Configuring Message Tabs* on page 226.

7. In the Add Notification for Step window, click **OK.**

   The Add Notification for Step window closes. The **Notifications** tab lists the notifications added.

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

   The changes to the request type are saved.

Configuring Setup Tabs

To configure the setup tab:

1. Open the Request Type Workbench.

   To open the Request Type Workbench, see *Opening the Request Type Workbench* on page 161. The Request Type Workbench window opens.

2. Open a request type.

   The Request Type window opens.

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

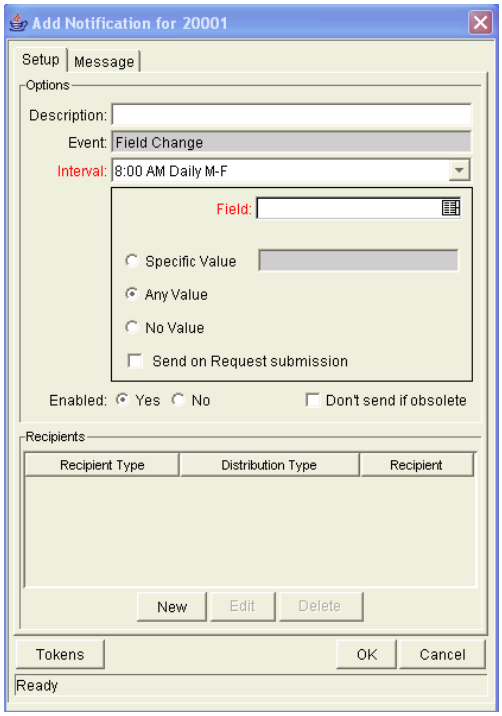
   The **Notifications** tab opens.

4. In the **Notifications** tab, click **New.**

   The Add Notification for Step window opens. The Add Notification for Step window has two tabs: **Setup** and **Message.**

5. In the Add Notification for Step window, select the **Setup** tab.

   The **Setup** tab is the default tab.



6. Configure the Options section of the **Setup** tab as specified in the following table:

Field	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	Description
Field	Selects the request type field that triggers the notification from the drop-down list. When a change occurs in the selected field, the notification will be sent.
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.

7. Configure the Recipients section of the **Setup** tab.

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

      The Add New Recipient window opens.

   b. In the Add New Recipient window, Select To, Cc, or Bcc.

   c. In the Add New Recipient window, select the recipient.

      - **Enter a Username.** Select a user as the recipient of the notification. Selecting a user changes the name of the autocomplete field 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 autocomplete field to Email Address. The security type is dynamically changed to **Email Address.**

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

- **Enter a Standard Token.** Select a standard token to act upon the workflow step. Selecting a standard token changes the name of the autocomplete field to Standard Token. The security type is left undefined. Select a standard token from the autocomplete field. 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 autocomplete field to User Defined Token. The security type is dynamically changed to a drop-down 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.

d. In the Add New Recipient window, click **OK.**

The recipient is added.

8. In the **Setup** tab, 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 tab:

1. Open the Request Type Workbench.

   To open the Request Type Workbench, see *Opening the Request Type Workbench* on page 161. The Request Type Workbench window opens.

2. Open a request type.

   The Request Type window opens.

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

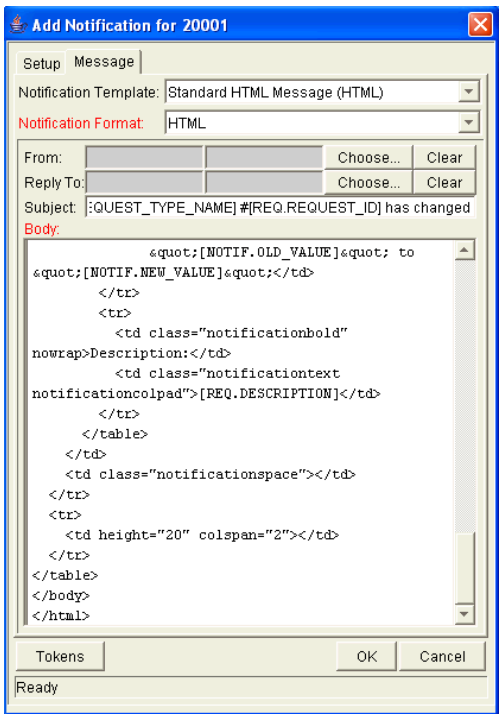
   The **Notifications** tab opens.

4. In the **Notifications** tab, click **New.**

   The Add Notification for Step window opens. The Add Notification for Step window has two tabs: **Setup** and **Message.**

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

   The **Message** tab opens.



6. Select a Notification Template from the drop-down list.

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

7. In the Notification Format field, select **HTML** or **Plain Text** from the drop-down list.

   Selecting **HTML** allows more flexibility when formatting the look and feel of the notification. The HTML code can be written and tested in any HTML editor and then pasted into the Body window.

8. Select values for the From and Reply to fields.

   a. In the From and Reply to fields, click Choose.

      The Email Header Field window opens.

   b. In the Email Header Field window, select the recipient.

      - **Enter a Username.** Select a user as the recipient of the notification. Selecting a user changes the name of the autocomplete field 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 autocomplete field 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 autocomplete field to Standard Token. The security type is left undefined. Select a standard token from the autocomplete field. 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 autocomplete field to User Defined Token. The security type is dynamically changed to a drop-down 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.

c. In the Email Header Field window, click **OK.**

The selected recipients are added to the **Message** tab.

9. Construct the body of the message.

When constructing the body, consider utilizing 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 the **Tokens** button 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 list to include tokens that resolve information related to the individual package line.

10. In the **Message** tab, click **OK.**

    The Add Notification for Step window closes. The **Notifications** tab is enabled.

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

    The changes to the request type are saved.

Editing Notifications

To edit a notification:

1. Open the Request Type Workbench.

   To open the Request Type Workbench, see *Opening the Request Type Workbench* on page 161. The Request Type Workbench window opens.

2. Open a request type.

   The Request Type window opens.

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

   The **Notifications** tab opens.

4. In the **Notifications** tab, select a notification and click **Edit.**

   The Add Notification for Step window opens. The Add Notification for Step window has two tabs: **Setup** and **Message.**

   To edit the **Setup** tab, see *Configuring Setup Tabs* on page 222.

   To edit the **Message** tab, see *Configuring Message Tabs* on page 226.

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

   The changes to the request type are saved.

Copying Notifications

To delete a notification:

1. Open the Request Type Workbench.

   To open the Request Type Workbench, see *Opening the Request Type Workbench* on page 161. The Request Type Workbench window opens.

2. Open a request type.

   The Request Type window opens.

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

   The **Notifications** tab opens.

4. In the **Notifications** tab, select a notification and click **Copy.**

   The Add Notification for Step window opens. The Add Notification for Step window has two tabs: **Setup** and **Message.**

   To edit the **Setup** tab, see *Configuring Setup Tabs* on page 222.

   To edit the **Message** tab, see *Configuring Message Tabs* on page 226.

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

   The changes to the request type are saved.

Deleting Notifications

To delete a notification:

1. Open the Request Type Workbench.

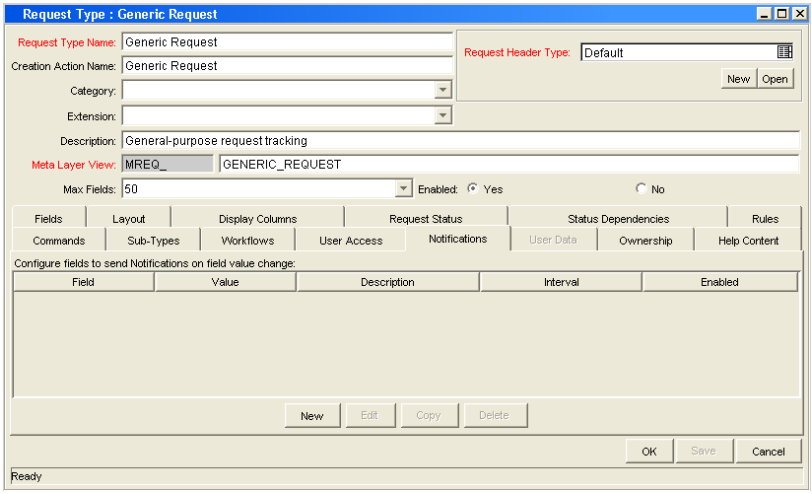
   To open the Request Type Workbench, see *Opening the Request Type Workbench* on page 161. The Request Type Workbench window opens.

2. Open a request type.

   The Request Type window opens.

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

   The **Notifications** tab opens.



4. In the **Notifications** tab, select a notification and click **Delete.**

   The notification is deleted.

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

   The changes to the request type are saved.

Configuring Ownerships of Request Types

Ownership groups are defined by adding Security groups to the **Ownership** tab. If no ownership groups are associated with the entity, the entity is considered global and any user with the Edit Access access grant for the entity can edit, copy or delete it. Refer to *Security Model Guide and Reference* for more information on access grants.

If a security group is disabled or loses the Edit Access access grant, that group will no longer be able to edit the entity.

Adding Ownerships to Request Types

To add an ownership:

1. Open the Request Type Workbench.

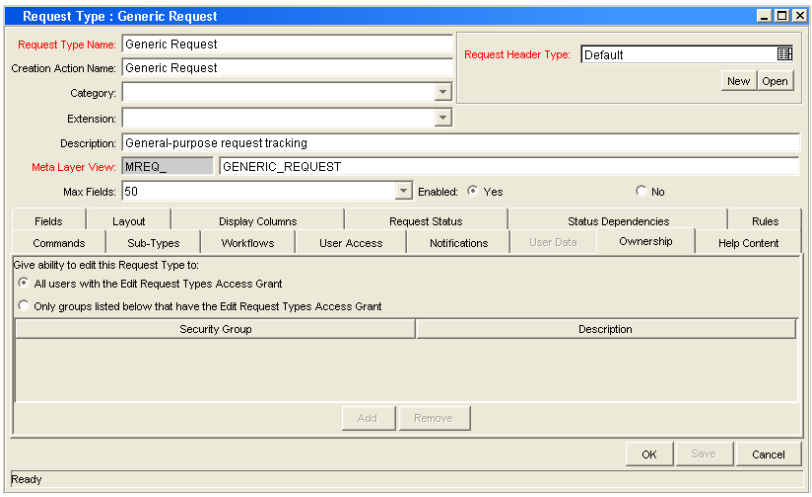
   To open the Request Type Workbench, see *Opening the Request Type Workbench* on page 161. The Request Type Workbench window opens.

2. Open a request type.

   The Request Type window opens.

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

   The **Ownership** tab opens.



4. In the **Ownership** tab, select the ownership option.

   The All user with the Edit Request Type access grant option give all users with the Edit Request Type access grant ownership of 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 Add Security Groups window, in the Security Groups field, select the security groups.

      The Validate window opens.

   c. In the Validate window, select one or more security groups and click **OK.**

      The Validate window closes. The Add Security Groups window lists the selected security groups.

   d. In the Add Security Groups window, click **OK.**

   e. The Add Security Groups window closes. The selected security groups are display in the **Ownership** tab under the Security Group column.

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

   The changes to the request type are saved.

Deleting Ownerships from Request Types

To delete an ownership:

1. Open the Request Type Workbench.

   To open the Request Type Workbench, see *Opening the Request Type Workbench* on page 161. The Request Type Workbench window opens.

2. Open a request type.

   The Request Type window opens.

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

   The **Ownership** tab opens.

4. In the **Ownership** tab, select an ownership.

   The All user with the Edit Request Type access grant option give all users with the Edit Request Type access grant ownership of 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. In the **Ownership** tab, click **Remove.**

   The ownership is deleted.

6. In the **Ownership** tab, click **OK.**

   The changes to the request type are saved.

Configuring Help Contents for Request Types

It is possible to provide accessible online information for 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. Open the Request Type Workbench.

   To open the Request Type Workbench, see *Opening the Request Type Workbench* on page 161. The Request Type Workbench window opens.

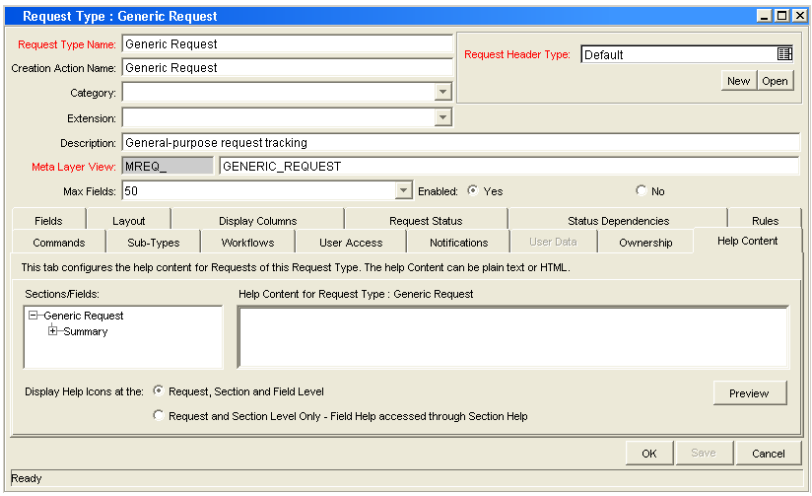
2. Open a request type.

   The Request Type window opens.

3. In the Request Type window, click the **Help Content** tab.

   The **Help Content** tab opens.

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



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

   Enter plain text or HTML-formatted text. To see what the text looks like in the actual help display, click **Preview.**

6. (Optional) Select other sections or fields to define help content for those items.

7. From the Display Help Icons at the: field, specify how the help icons will be shown 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.

8. In the **Help Content** tab, click **Save.**

   The changes to the request type are saved.

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 will be 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	Description
(REFERENCE) Default	The default request header type. Includes a percentage complete (% Complete) field.
(REFERENCE) Comprehensive	Displays all information. Consistent with previous versions of Mercury IT Governance Center.
(REFERENCE) Simple	Displays only the most essential information.
(REFERENCE) Departmental	An example request header type for simple cross-departmental requests.
(REFERENCE) Application	An example request header type for simple cross-application requests.
(REFERENCE) 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 (see *Figure 5-7*).

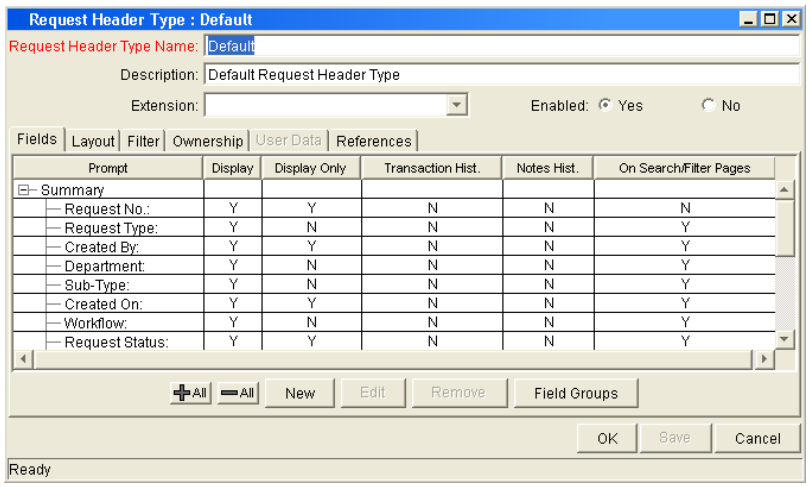


Figure 5-7. Request Header Type window

The following is a list of the main components of a request header type:

- **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 242.
- **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 167.
- **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 180.
- **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 243.
- **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 232.

- **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 239.

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 will behave just like normal fields, 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 240 lists the request header type field groups that are delivered with various Mercury IT Governance Center products.

Field groups can be added 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.

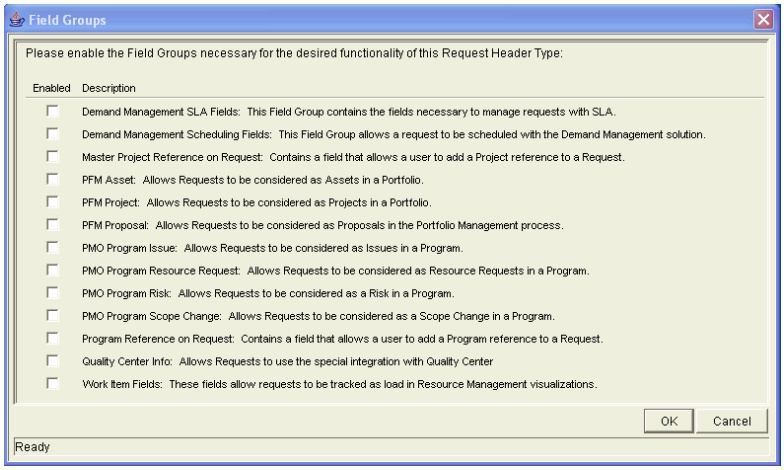


Figure 5-8. Request Header Type Field Groups window

Table 5-7. Request header type field groups

Field Group	Description
Demand Management SLA	This field group contains the fields necessary to manage requests with SLA.
Demand Management Scheduling	This field group 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 a risk in a program.
PMO Program Scope Change	Allows requests to be considered as a scope change 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.
Portfolio Management Asset	Contains the fields necessary to create a PFM asset.

Table 5-7. Request header type field groups [continued]

Field Group	Description
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 contains fields that allow requests to be scheduled as a work item.

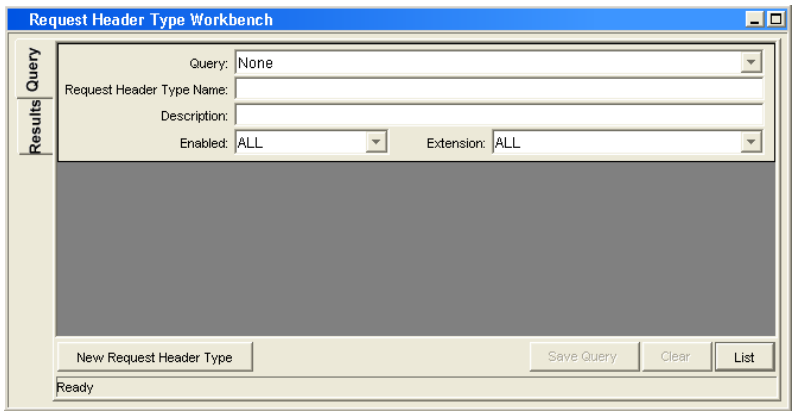
Opening the Request Header Type Workbench

To open the Request Header Type Workbench:

1. Log on to the Mercury IT Governance Center.
2. From the menu bar, select **Administration > Open Workbench.**
3. A Workbench status window opens. A few minutes later, a Warning - Security window opens.
4. In the Warning - Security window, select **Yes.**

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

   The Request Header Type Workbench window opens.



For More Information

For information on how to search and select an existing request header type, copy a request header type, and delete a request header type, see *Getting Started*.

Configuring General Information for Request Header Types

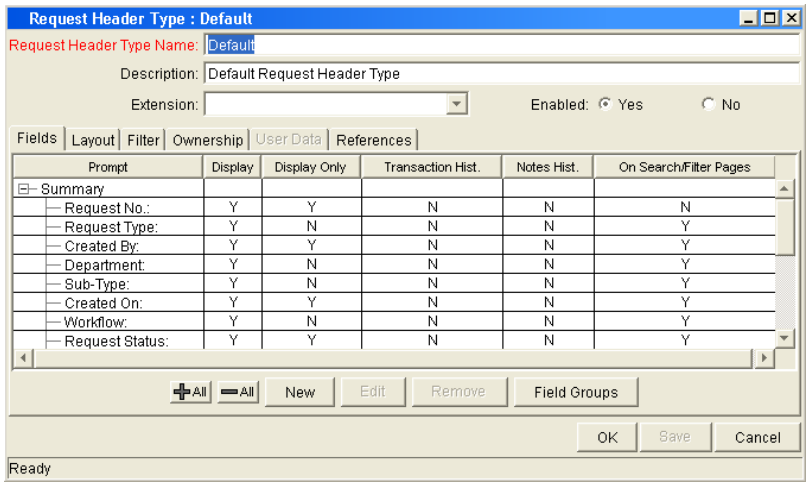
To configure the general information of a request header type:

1. Open the Request Header Type Workbench.

   To open the Request Header Type Workbench, see *Configuring Request Header Types* on page 237. The Request Header Type Workbench window opens.

2. Open a request header type.

   The Request Header Type window opens.



3. Complete the fields in the Request Header Type window as specified in the following table:

Field	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 Change Management extension. Select the extension from the drop-down 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. Save the changes to the request header type.

   Click **OK** to save the changes and close the Request Header Type window. Click **Save** to save the changes and leave the Request Header Type window open. Click **Cancel** to lose the changes and close the Request Header Type window.

Configuring Filters for Request Header Types

To configure filters for a request header type:

1. Open the Request Header Type Workbench.

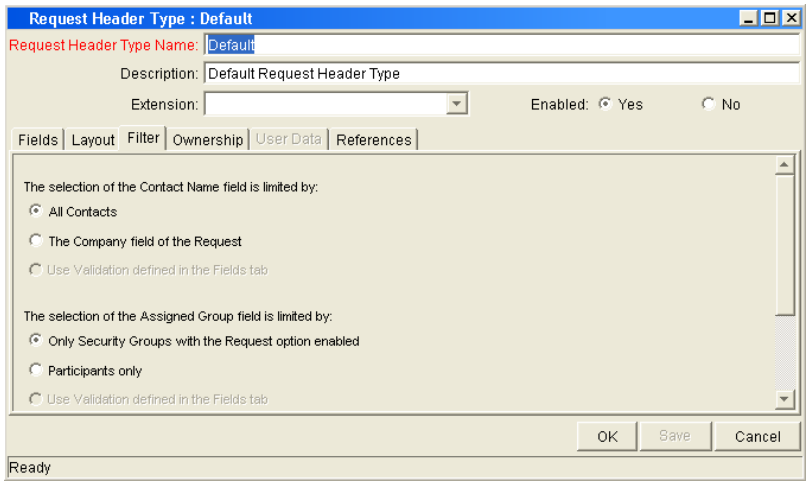
   To open the Request Header Type Workbench, see *Configuring Request Header Types* on page 237. The Request Header Type Workbench window opens.

2. Open a request header type.

   The Request Header Type window opens.

3. In the Request Header Type window, select the **Filter** tab.

   The **Filter** tab opens.



4. Complete the fields in the **Filter** tab as specified in the following table:

Field	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.

Field	Description
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.
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. In the **Filter** tab, click **OK.**

   The changes to the request header type are saved.

Chapter

6

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 system as well as external users.

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

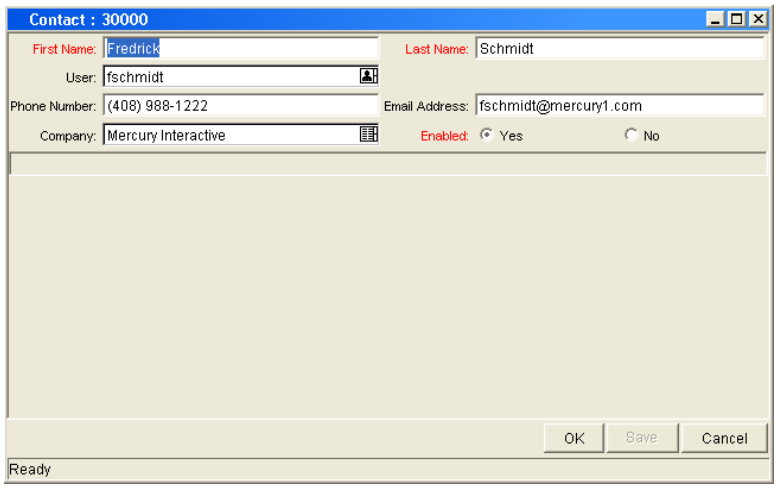


Figure 6-1. Contact window

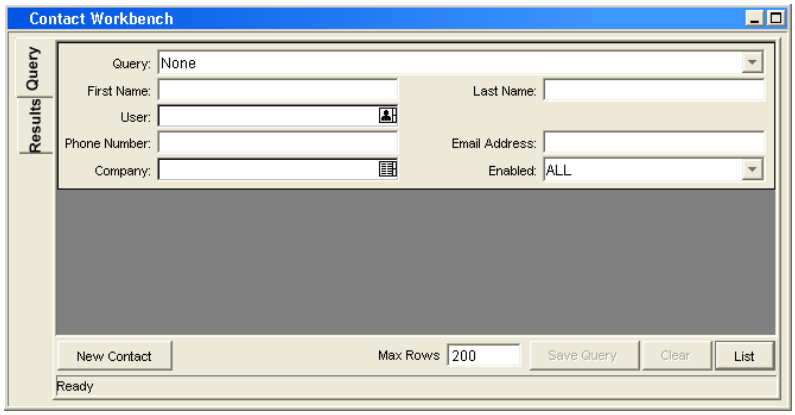
Opening the Contact Workbench

To open the Contact Workbench:

1. Log on to the Mercury IT Governance Center.
2. From the menu bar, select **Administration > Open Workbench.**
3. A Workbench status window opens. A few minutes later, a Warning - Security window opens.
4. In the Warning - Security window, select **Yes.**

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

   The Contacts Workbench opens.



For More Information

For information on how to search and select an existing contact, copy a contact, and delete a contact, see *Getting Started*.

Creating Contacts

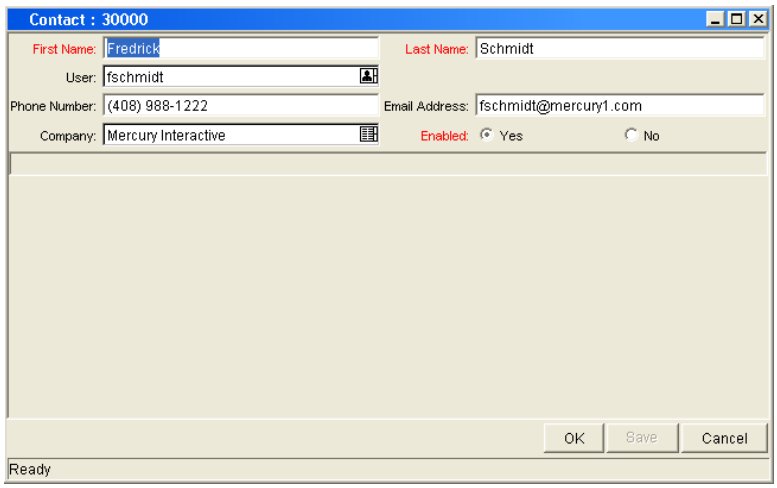
To create a new contact:

1. Open the Contacts Workbench.

   To open the Contacts Workbench, see *Opening the Contact Workbench on page 249*. The Contacts Workbench window opens.

2. In the Contacts Workbench, click **New Contact.**

   The Contact window opens.



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

Field	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 list and cannot be edited. You should select a username from the validation auto-complete list.
Phone Number	The phone number of the contact.
Email Address	The email address of the contact.

Field	Description
Company	The company employing the contact. This field is populated from **CRT - Company Validation** auto-complete list and cannot be edited. You should select a company from the validation auto-complete list.
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

7

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

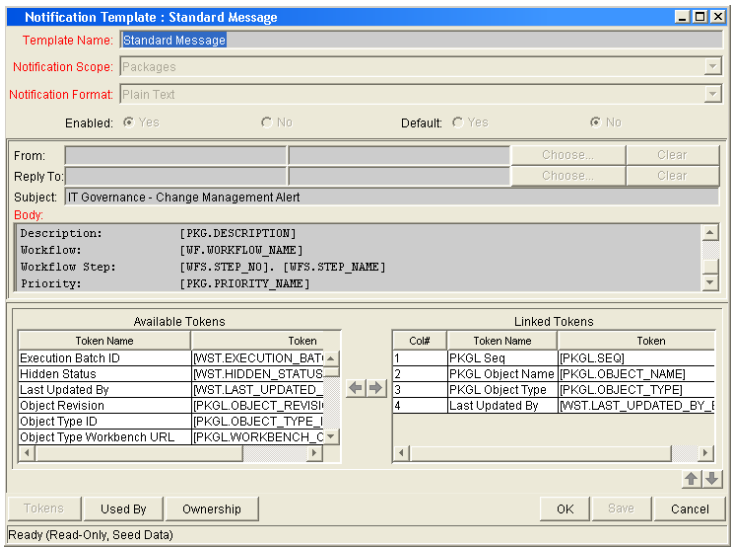


Figure 7-1. Notifications Template window

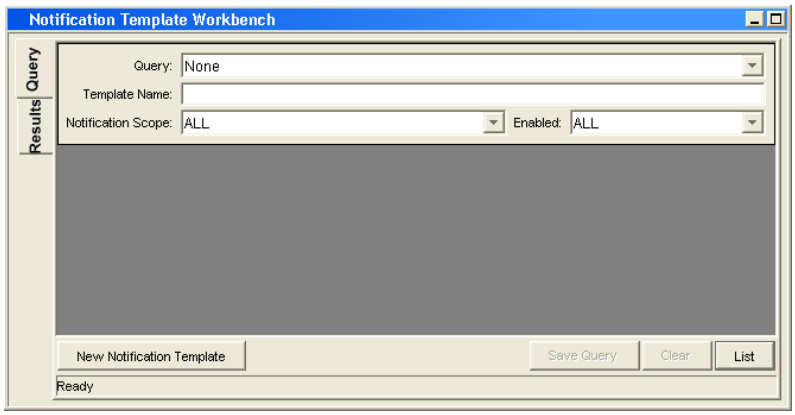
Opening the Notification Templates Workbench

To open the Notification Template Workbench:

1. Log on to the Mercury IT Governance Center.
2. From the menu bar, select **Administration > Open Workbench.**
3. A Workbench status window opens. A few minutes later, a Warning - Security window opens.
4. In the Warning - Security window, select **Yes.**

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

   The Notification Template Workbench window opens.



For More Information

For information on how to search and select an existing notification template, and copy a notification template, see *Getting Started*.

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 267.

For information on how to delete a notification template type, see *Getting Started*.

Creating Notification Templates

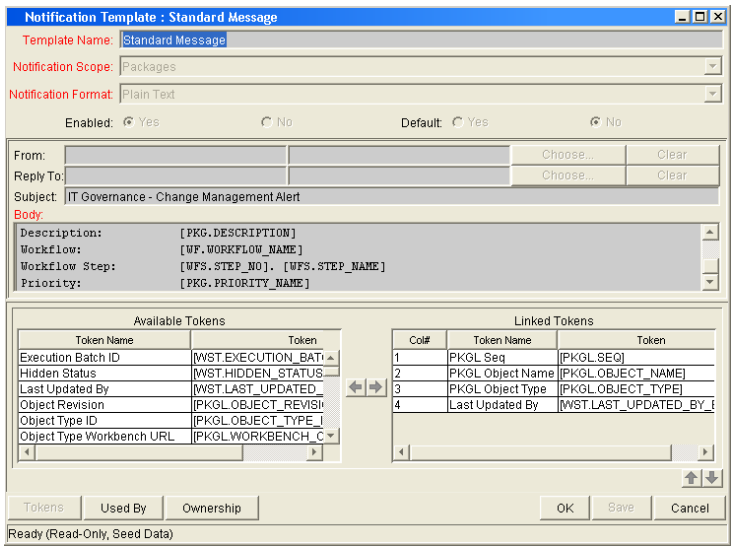
To create a new notification template:

1. Open the Notification Template Workbench.

   To open the Notification Template Workbench, see *Opening the Notification Templates Workbench* on page 255. The Notification Template Workbench window opens.

2. Click **New Notification Template.**

   A Notification Template window opens.



3. Complete the fields in the Notification Template window as specified in the following table:

Field	Description
Template Name	Enter the name of the new notification template.
Notification Scope	Include the product area where this notification template will be used. Select an entry from the drop-down list. Entries include: • 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. Select an entry from the drop-down list. Entries include: • Plain Text • HTML
Enabled	Make the notification template available to the system. Select **Yes** to make the notification available to the system.
Default	Make the notification template the default notification template for the system Select **Yes** to make the notification template the default notification template.

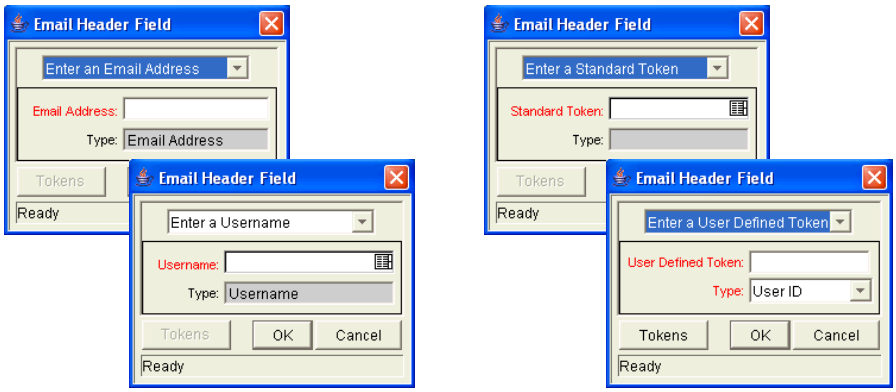
4. In the Notification Template window, enter a From address.

   a. In the Notification Template window, in From, click **Choose....**

      The Email Header Field window opens.

b. Select the recipient category from the drop-down list (**Username, Email Address, Standard Token,** or **User Defined Token**).

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 drop-down list, then it is necessary to enter an Email Address. If a **User Defined Token** is selected, click **Tokens** to bring up a full 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 with the evaluated token value.

e. In the Email Header Field window, click **OK.**

The Email Header Field window closes.

5. In the Notification Template window, enter a Reply address.

a. In the Notification Template window, in From, click **Choose....**

The Email Header Field window opens.

b. Select the recipient category from the drop-down list (**Username, Email Address, Standard Token,** or **User Defined Token**).

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 drop-down list, then it is necessary to enter an Email Address. If a **User Defined Token** is selected, click **Tokens** to bring up a full list of available tokens or type in a specific token.

c. Enter the appropriate information in the required field.

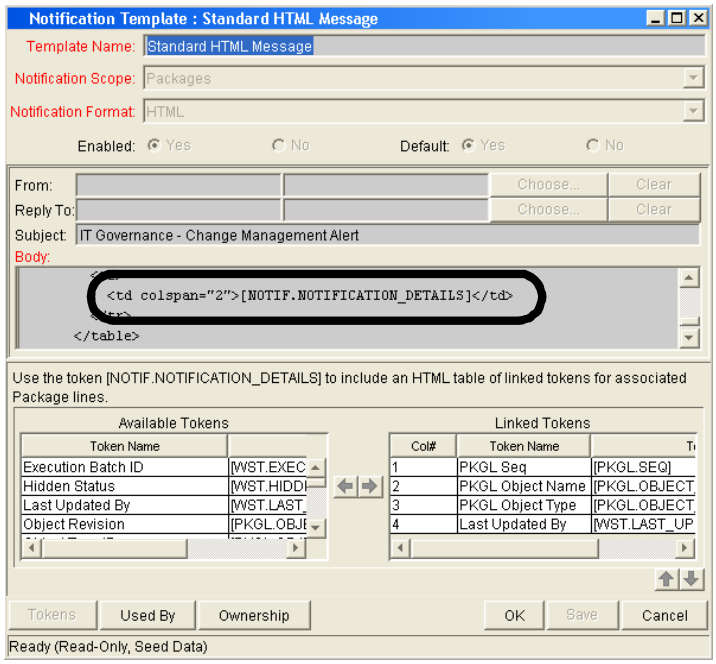
d. If a **User Defined Token** has been entered, select the token type that corresponds with the evaluated token value.

e. In the Email Header Field window, click **OK.**

The Email Header Field window closes.

6. In the Notification Template window, in Body, 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 Change Management should include the token '[NOTIF.NOTIFICATION_ DETAILS]' within the **<body>** tags to incorporate linked tokens.



7. In the Notification Template window, in Body, add tokens to the body of the text.

   If you need to add tokens to the body of the notification template:

   a. At the bottom of the Notification Template window, click **Tokens.**

      The Token Builder window opens.

   b. From the Token Builder window, select a token.

   c. In the Token Builder window, in the Token field, copy the name of the token and paste the name in the Body field.

   d. In the Token Builder window, click **Close.**

      The Token Builder window closes.

8. In the Notification Template window, configure the ownership of the notification template.

   For detailed information on how to configure the ownership of the notification template, see *Configuring Ownership of Notification Templates* on page 261.

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

   The changes to the notification template are saved.

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 access grant for the entity can edit, copy or delete it. Refer to *Security Model Guide and Reference* for more information on access grants.

If a security group is disabled or loses the Edit Access access grant, that group will no longer be able to edit the entity.

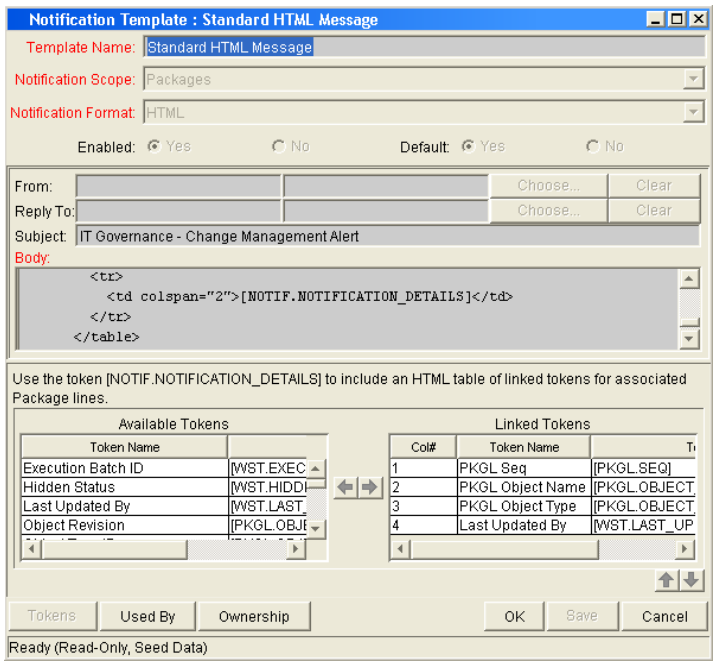
To configure the ownership of a notification template:

1. Open the Notification Template Workbench.

   To open the Notification Template Workbench, see *Opening the Notification Templates Workbench* on page 255. The Notification Template Workbench window opens.

2. Open a notification template.

   A Notification Template window opens.



3. At the bottom of the Notification Template window, click **Ownership.**

   The Ownership window opens.

4. In the **Ownership** window, select the ownership option.

   - All user with the Edit Notification Template access grant gives all users with the Edit Notification Template access grant can have ownership of the notification template.
   - Only groups listed below that have the Edit Notification Template access grant requires selected groups to be added to the ownership of the notification template.

   To select ownerships:

   a. In the **Ownership** window, deselect Only groups listed below that have the Edit Notification Template.

   b. In the **Ownership** window, click **Add.**

      The Add Security Groups window opens.

   c. In the Add Security Groups window, in the Security Groups field, select the security groups.

      The Validate window opens.

   d. In the Validate window, select one or more security groups and click **OK.**

      The Validate window closes. The Add Security Groups window lists the selected security groups.

   e. In the Add Security Groups window, click **OK.**

   f. The Add Security Groups window closes. The selected security groups are display in the **Ownership** tab under the Security Group column.

5. In the **Ownership** window, click **OK.**

   The changes to the notification template are saved.

Deleting Ownerships from Notification Templates

To delete an ownership:

1. Open the Notification Template Workbench.

   To open the Notification Template Workbench, see *Opening the Notification Templates Workbench* on page 255. The Notification Template Workbench window opens.

2. Open a notification template.

   A Notification Template window opens.

3. At the bottom of the Notification Template window, click **Ownership.**

   The Ownership window opens.

4. In the **Ownership** window, select an ownership.

   The All user with the Edit Notification Template access grant option give all users with the Edit Notification Template access grant ownership of the notification template. The Only groups listed below that have the Edit Notification Template access grant option requires selected groups to be added to the ownership of the notification template.

5. In the **Ownership** window, click **Remove.**

   The ownership is deleted.

6. In the **Ownership** window, click **OK.**

   The changes to the notification template are saved.

Configuring Notification Intervals

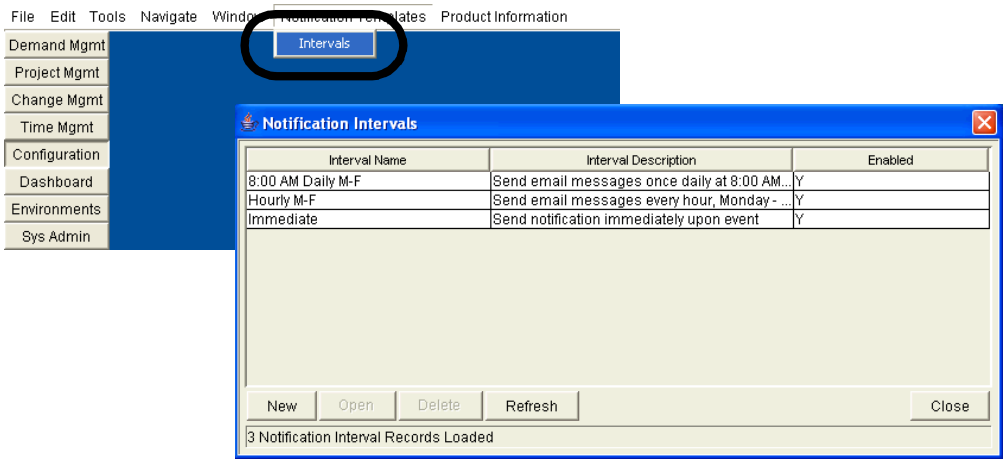
To create a new notification template:

1. Open the Notification Template Workbench.

   To open the Notification Template Workbench, see *Opening the Notification Templates Workbench* on page 255. The Notification Template Workbench window opens.

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

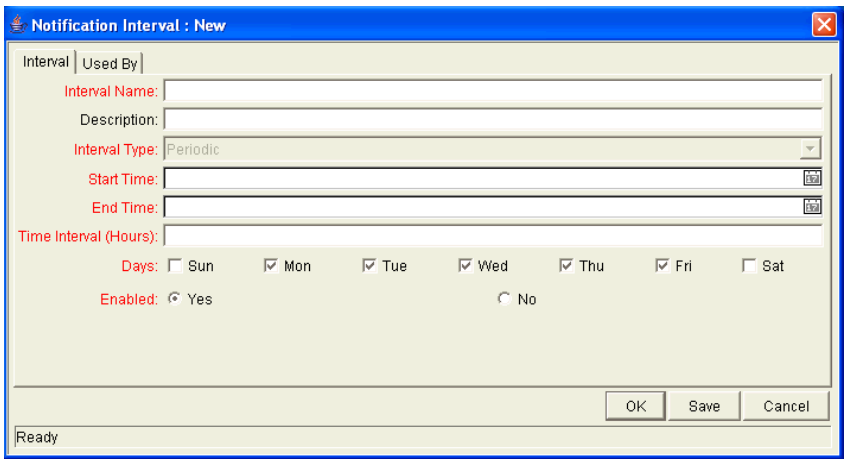
   The Notification Intervals window opens.



3. In the Notification Intervals window, click **New.**

   The the Notification Intervals window opens.

4. Complete the fields in the **Interval** tab as specified in the following table:



Field Name	Description
Interval Name	This is the name assigned to the interval.
Description	Free form description of this 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	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 this interval should execute on.
Enabled	If **Yes** is set, this interval is selectable. If **No** is set, this interval is unavailable.

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

   The Notification Interval window closes. The new interval is added to the system.

6. In the Notification Intervals window, click **Close.**

   The Notification Intervals window closes. The new notification interval can now be used in any workflow step notification.

When notifications are sent with 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 individual notification appears at the top of the email message in a Summary section.

Checking the Usage of Notification Templates

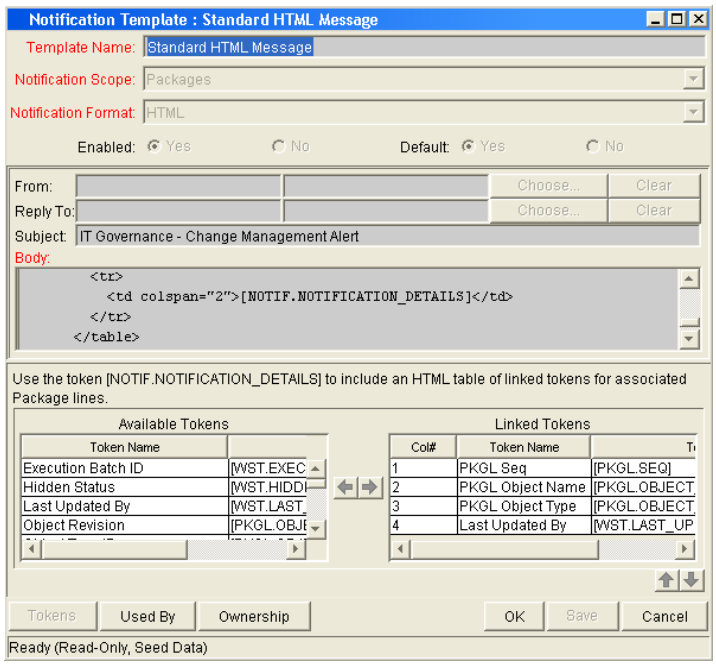
To check the usage of a notification template:

1. Open the Notification Template Workbench.

   To open the Notification Template Workbench, see *Opening the Notification Templates Workbench* on page 255. The Notification Template Workbench window opens.

2. Open notification template.

   A Notification Template window opens.



3. At the bottom of the Notification Template window, click **Used By.**

   The Used By window opens. All references to the notification template are listed.

4. In the Used By window, click **OK.**

   The Used By window closes.

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

   The Notification Template window closes.

Chapter

8

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, user data fields provide the ability to capture additional information specific to each organization. For example, you might want to include an additional field on every package. To accomplish this, you would open **Validation Value User Data** and define the extra field. Once defined, the field would appear on a validation's **User Data** tab.

User data types are configured in the User Data Workbench in the User Data Context window. *Figure 8-1* illustrates a partial list of the available user data types.

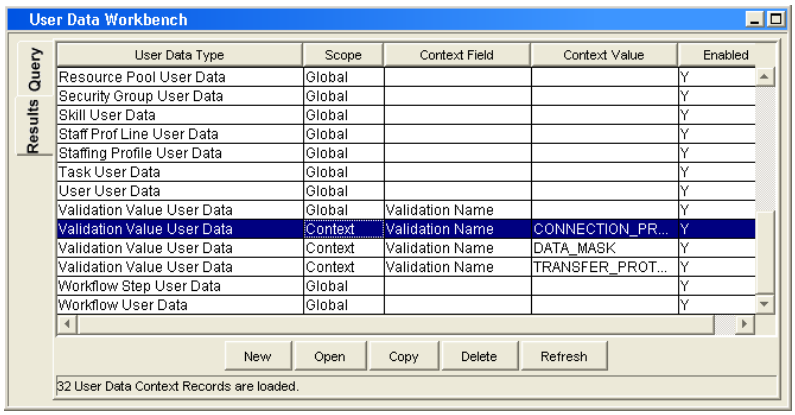


Figure 8-1. User data types

Each user data type consists of four components. All four of these components are required to fully identify a user data type. The following lists these components:

- **User Data Type.** The User Data Type field lists the name of the user data type. All available user data types are created by Mercury IT Governance Center. You can only define fields for a user data type. You cannot create a new user data type.

- **Scope.** The scope refers to the category of the user data type. There are two available scopes for user data types:
  - **Global.** The standard user data type scope. When Scope is defined as **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.
- **Context Field.** The Context Field is the name of the context sensitive field. The Context Field is only applicable to user data types with a scope of **Context.** There is only one Context Field value available for each user data type. As a result, Context Fields are filled in automatically.
- **Context Value.** The Context Value is the value (context) of the context sensitive field. The **Context Value** is only applicable to user data types with a scope of **Context.** There are multiple, pre-defined values for Context Value. You cannot create a new Context Value, you can only assign an available Context Value.

Mercury IT Governance Center can contain up to twenty user data fields that can be defined. These fields are displayed in the **User Data** tab of the defined entity. The major attributes of each of these fields, such as their graphical presentation, the validation method, and whether or not they are required can be configured.

Referring to User Data

Once a user data field has been created, it is possible to refer to it from other parts of the product by its token name, proceeded by the entity abbreviation and the UD qualifier.

Migrating User Data

For any configuration entity with user data fields the data in the user data fields is migrated along with the entity.

- If two instances have identical user data configurations, then the user data will be migrated correctly.
- If two instances do not have identical user data configurations, then the user data will be mapped into the data model according to the storage configuration in the source instance. For this reason, the two instances should be configured with the same user data fields, or the user data should be corrected after migration.
- If the user data is context sensitive, then a corresponding context sensitive configuration must exist in the destination instance, or the migration will fail.
- User data fields that have different hidden and visible values may be problematic. When the hidden value of a user data field refers to a primary key such as, Security Group ID, that can be different in the source and destination instances, then the migrator does not correct the hidden value. The user data should be corrected manually after migration.

Overview of Configuring User Data

The following is a list of the main components of User Data Context window:

- **General information.** General information includes basic information concerning the user data, such as the user data type and the user data context value. See *Configuring General Information for User Data Types* on page 274.
- **Fields.** The **Fields** tab is used to create additional fields for a user data type. See *Creating User Data Fields* on page 277.
- **Layout.** Once all of the fields are created for a user data type, the layout of those fields can be configured using the **Layout** tab. See *Configuring User Data Layouts* on page 288.

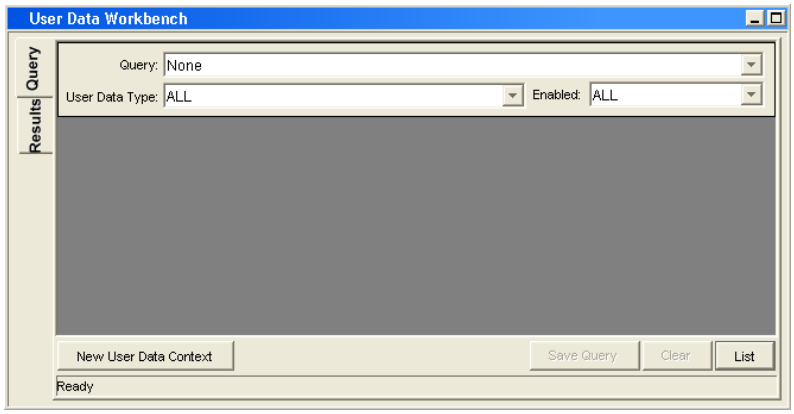
Opening the User Data Workbench

To open the User Data Workbench:

1. Log on to the Mercury IT Governance Center.
2. From the menu bar, select **Administration > Open Workbench.**
3. A Workbench status window opens. A few minutes later, a Warning - Security window opens.
4. In the Warning - Security window, select **Yes.**

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

   The User Data Workbench window opens.



For More Information

For information on how to search and select an existing user data, copy user data, and delete user data, see *Getting Started*.

Configuring General Information for User Data Types

To configure the general information for a user data type:

1. Open the User Data Workbench.

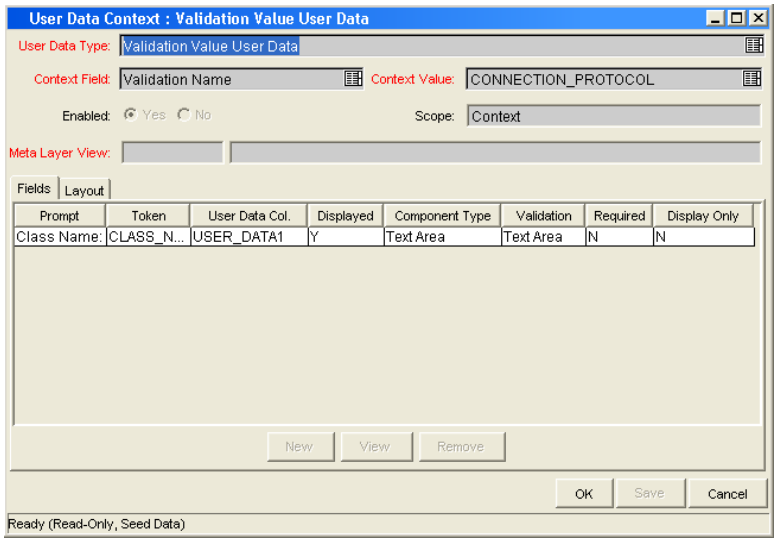
   To open the User Data Workbench, see *Opening the User Data Workbench* on page 273. The User Data Workbench window opens.

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

   When configuring a global user data type, you must open an existing user data type. For example, Skill User Data is already created by 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. In the User Data Context window, complete the fields in the User Data Context window as specified in the following table:

Field	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 drop-down 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 drop-down 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.

Field	Description
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 is defined as **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. Save the changes to the user data type.

   Click **OK** to save the changes and close the User Data Context window. Click **Save** to save the changes and leave the User Data Context window open. Click **Cancel** to lose the changes and close the User Data Context window.

Creating User Data Fields

To create a new user data field:

1. Open the User Data Workbench.

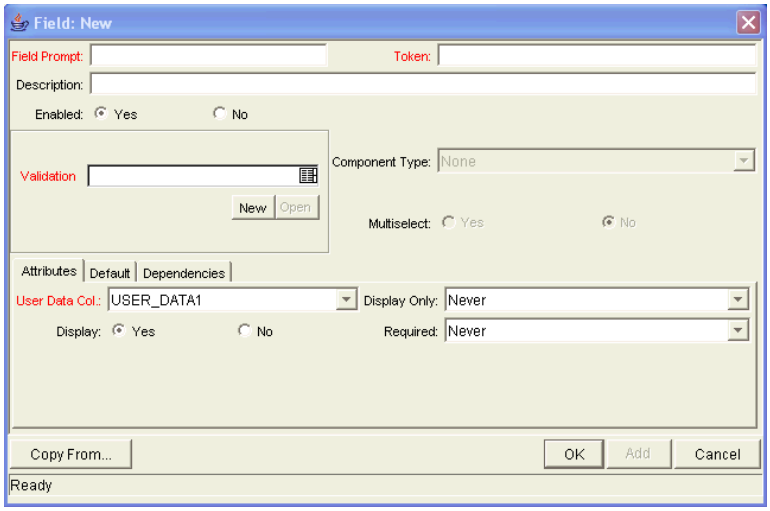
   To open the User Data Workbench, see *Opening the User Data Workbench* on page 273. The User Data Workbench window opens.

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

   The User Data Context window opens. The **Fields** tab is displayed.

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

   The Field window opens.



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

Field	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.

Field	Description
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 (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 autocomplete component for the validation.

5. In the Field window, click the **Attributes** tab.

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

Field	Description
User Data Col	Indicates the internal column that the field value will be stored in. These values will 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, therefore allowing up to 20 fields. No two fields in user data can use the same column.
Display Only	Indicates whether the field is only displayed and cannot be updated. Select **Use Dependency Rules** to use the logic defined in the **Dependencies** tab.
Display	Indicates if the user sees this field in the **User Data** tab.
Required	Indicates if the user is required to specify a value for this field. Select **Use Dependency Rules** to use the logic defined in the **Dependencies** tab.

7. In the Field window, click the **Defaults** tab.

8. Complete the fields in the **Defaults** tab as specified in the following table:

Field	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. In the Field window, click the **Dependencies** tab.

10. Complete the fields in the **Dependencies** tab as specified in the following table:

Field	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 drop-down list containing 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 drop-down list containing logical qualifiers, and a text field. To use this functionality, select **Use Dependency Rules** in the **Attributes** tab.

11. In the Field window, click the **Security** tab to define which users can view or update this field.

    Enter the information as follows:

    a. In the **Security** tab, click **Edit.**

    b. Complete the fields in the Edit Field Security window as specified in the following table:

Field / Button	Description
Visible to all users	Checking this checkbox allows all users to see the field. If this checkbox 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 checkbox is not checked, the Select User/ Security Group that can view this field area is activated. Deselecting the **Visible to all users** or **Editable by all users** checkboxes enables the Select Users/Security Groups that can view this field area of the Edit Field Security window.
Editable by all users	Checking this checkbox allows all users to edit the field. If this checkbox 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 area of the Edit Field Security window.
Enter a Security Group (drop-down list)	To select the format for specifying users to grant visibility and edit permission, use the **Enter a Security Group** drop-down list. The drop-down lists the formats to choose users. The drop-down list dynamically updates the Security Group Validate autocomplete window list. 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 drop-down list dynamically updates the Security Group field.
Security Group	Provides a field for specifying the recipient. If the Enter a Security Group drop-down list is**:** • **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. In the Edit Field Security window, click **OK.**

    The Edit Field Security window closes. The new field appears in the Field window.

13. In the Field window, click **OK.**

    The changes to the User Data Context field are saved.

Copying a Field's Definition

The **Copy From** functionality can also be utilized to streamline the process of adding fields by copying the definition of existing fields.

To copy a field's definition:

1. Open the User Data Workbench.

   To open the User Data Workbench, see *Opening the User Data Workbench* on page 273. The User Data Workbench window opens.

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

   The User Data Context window opens. The **Fields** tab is displayed.

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

   The Field window opens.

4. In the **Fields** tab, click **New.**

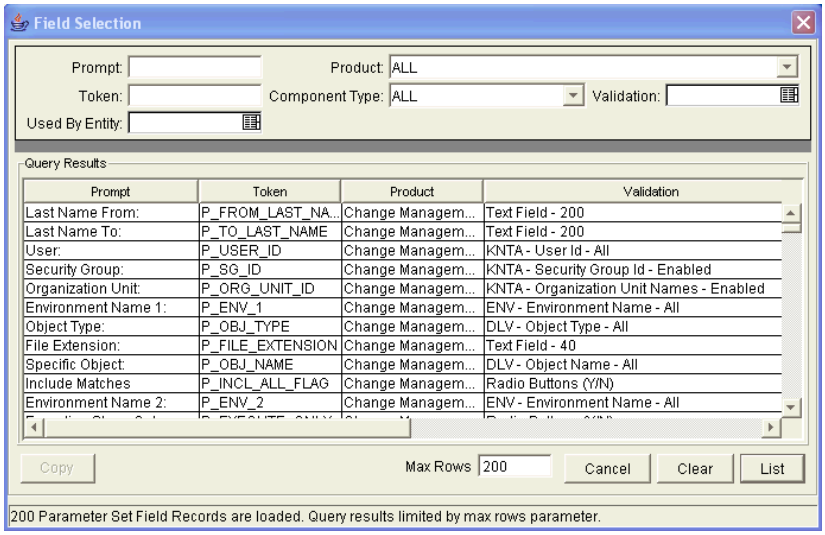
   The Field window opens.

5. Click **Copy From.**

   The Field Selection window opens.

6. In the Field Selection window, complete the fields and click **Copy From.**

   The Field Selection window refreshes with fields matching the search criteria.



7. In the Field Selection window, select a field to copy.

   Query fields by a number of criteria, such as the token name or field prompt. It is also possible to perform more complex queries such as listing all fields that reference a certain validation or are used by a certain entity.

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

   This closes the window and copies the definition of the selected field into the Field window.

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

   The changes to the user data type are saved.

Editing User Data Fields

To edit an existing field:

1. Open the User Data Workbench.

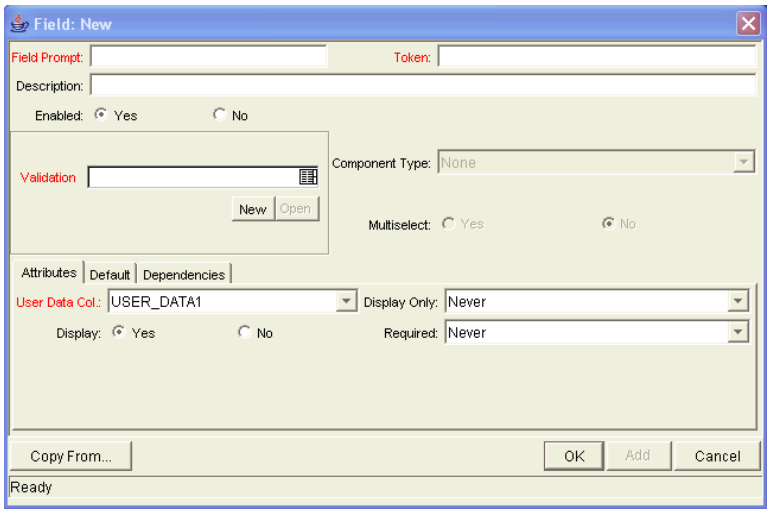
   To open the User Data Workbench, see *Opening the User Data Workbench* on page 273. The User Data Workbench window opens.

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

   The User Data Context window opens. The **Fields** tab is displayed.

3. In the User Data Context window, select the field and click **Edit.**

   The Field window opens.



4. In the Field window, make the desired changes in the header region, **Attributes** tab, **Default** tab, and **Dependencies** tab.

   For information concerning the **Attributes** tab, **Default** tab, and **Dependencies** tab, see *Creating User Data Fields* on page 277.

5. In the Field window, click **OK.**

   The Field window closes. The User Data Context field opens.

6. In the User Data Context window, click **OK.**

   The changes to the user data type are saved.

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.**

A field can be configured to:

- Clear when another field changes.
- Become read only when another field meets a logical condition, defined in *Table 8-1*.
- Become required when 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. Open the User Data Workbench.

   To open the User Data Workbench, see *Opening the User Data Workbench on page 273*. The User Data Workbench window opens.

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

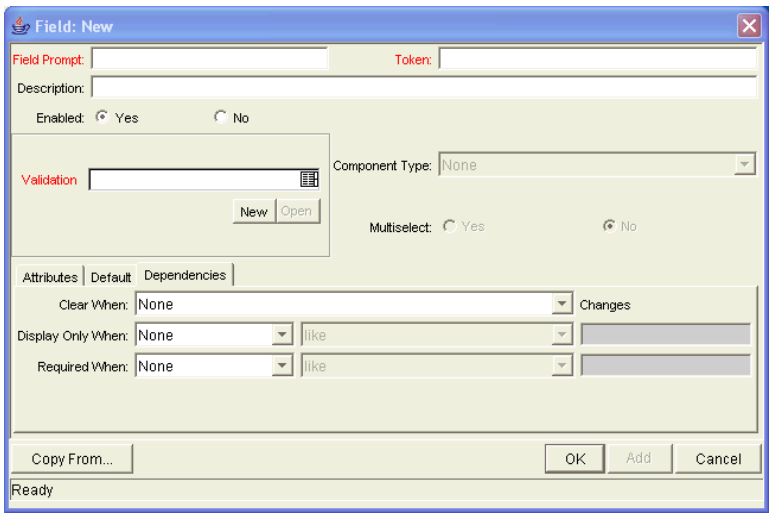
   The User Data Context window opens. The **Fields** tab is displayed.

3. In the User Data Context window, select the field and click **Edit.**

   The Field window opens.

4. In the Field window, click the **Dependencies** tab.

   The **Dependencies** tab opens.



5. In the **Dependencies** tab, set the field dependencies. It is possible to:

   - Select a field name from the Clear When drop-down list to indicate that the current field should be cleared when the selected field changes.

   - Select a field name from the Display Only When drop-down list to indicate that the current field should for display only (for example, not editable) when certain logical criteria are satisfied. This field functions with two adjacent fields. These are a drop-down list containing logical qualifier and another field which dynamically changes to a date field, drop-down list, or text field, depending on the selected field's validation.

   - Select a field name from the Required When drop-down list to indicate that the current field should be required when certain logical criteria are satisfied. This field functions with two adjacent fields. These are a drop-down list containing logical qualifier and another field which dynamically changes to a date field, drop-down list, or text field, depending on the selected field's validation.

6. In the **Dependencies** tab, click **OK.**

   The **Dependencies** tab closes. The Field window opens.

7. In the Field window, click **OK.**

   The Field window closes. The User Data Context window opens.

8. In the User Data Context window, click **OK.**

   The changes to the user data type are saved.

Removing Fields

To remove a field permanently from a user data type:

1. Open the User Data Workbench.

   To open the User Data Workbench, see *Opening the User Data Workbench* on page 273. The User Data Workbench window opens.

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

   The User Data Context window opens. The **Fields** tab is displayed.

3. In the User Data Context window, select the field and click **Remove.**

   The row is removed.

4. In the User Data Context window, click **OK.**

   The changes to the user data type are saved.

Configuring User Data Layouts

The layout of user data fields can be changed in the **Layout** tab of the User Data Context window.

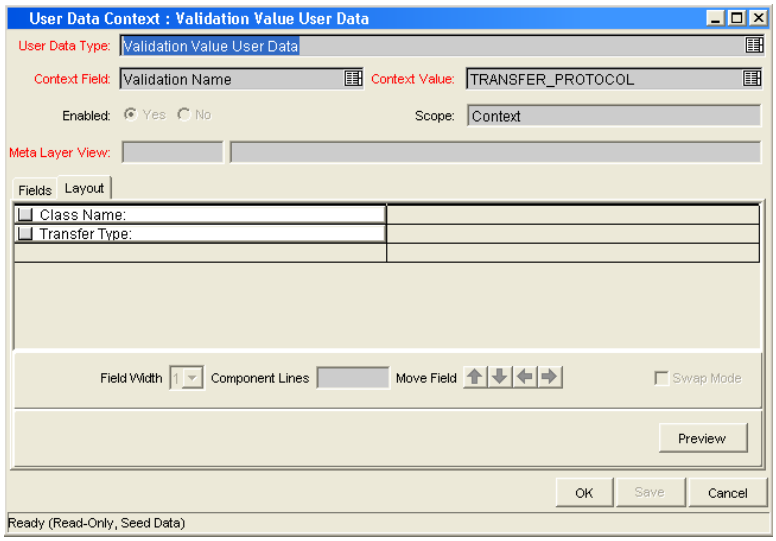


Figure 8-2. User Data window Layout tab

Changing Column Widths

To change the column width of a field:

1. Open the User Data Workbench.

   To open the User Data Workbench, see *Opening the User Data Workbench* on page 273. The User Data Workbench window opens.

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

   The User Data Context window opens. The **Fields** tab is displayed.

3. In the User Data Context window, click the **Layout** tab.

   The **Layout** tab opens.

4. In the **Layout** tab, select the field.

5. In the **Layout** tab, in Field Width, select either **1** or **2** inches.

   The Layout editor will not allow changes to be made if it conflicts with another field in the layout (for example, a field's width cannot be changed from one to two 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 will be blank and non-updateable.

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

   The changes to the user data type are saved.

Moving Fields

To move a field or a set of fields:

1. Open the User Data Workbench.

   To open the User Data Workbench, see *Opening the User Data Workbench* on page 273. The User Data Workbench window opens.

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

   The User Data Context window opens. The **Fields** tab is displayed.

3. In the User Data Context window, click the **Layout** tab.

   The **Layout** tab opens.

4. In the **Layout** tab, select the field.

   To select more than one field, press the Shift key while selecting the last field in a set. It is only possible to select a continuous set of fields.

   A field, or a set of fields, cannot be moved to an area where other fields already exist. Those other fields must be moved out of the way first.

5. At the bottom of the **Layout** tab, use the **Arrow** icons to move the fields to the desired location in the layout builder.

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

   The changes to the user data type are saved.

Swapping Positions of Two Fields

To swap the positions of two fields:

1. Open the User Data Workbench.

   To open the User Data Workbench, see *Opening the User Data Workbench* on page 273. The User Data Workbench window opens.

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

   The User Data Context window opens. The **Fields** tab is displayed.

3. In the User Data Context window, click the **Layout** tab.

   The **Layout** tab opens.

4. In the **Layout** tab, select the field.

5. In the **Layout** tab, select the Swap Mode check box.

   This causes an **S** to appear in the check box area of the selected field.

6. Once the **S** appears, double-click on the field to be swapped with.

   This causes the two fields to change positions. Following the swap, the swap mode is turned off.

7. In the **Layout** tab, click **OK.**

   The changes to the user data type are saved.

Previewing the Layout

You can check to see what the layout will look like in actual use. In the User Data Content window, in the **Layout** tab click **Preview**. This opens a small window that shows the fields as they will appear in the window, shown in *Figure 8-3*.

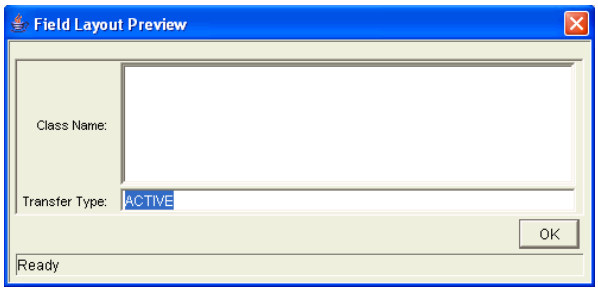


Figure 8-3. Preview mode

If all fields have a width of one column, all displayed columns will automatically span the entire available area when an entity of the given user data is being viewed or generated.

Non-displayed fields do not affect the layout. The layout engine considers them the same as a blank field.

Configuring Project and Task User Data Roll-Ups

The scope of this section is limited to Project Management.

Values from Task User Data fields can be configured to roll-up (combine and process values in a meaningful way) into parent Project User Data fields. The following types of task user data can roll up into project user data:

- Numeric fields (Text field component type with numeric data mask)
- Date fields

For each project, a Project User Data field can show a roll-up of Task User Data values using one of the following methods:

- **Average.** Shows the average of all values of a specified Task User Data field for every task under the project (numeric fields).
- **Maximum.** Shows the largest of all values of a specified Task User Data field for every task under the project (numeric and date fields).
- **Minimum.** Shows the smallest of all values of a specified Task User Data field for every task under the project (numeric and date fields).
- **Sum.** Shows the summation of all values of a specified Task User Data field for every task under the project (numeric fields).

Project and task user data roll-up can be used to capture various important aspects of a project. For example:

- Using the **Average** roll-up method, the average cost of all a project's tasks can be easily determined and automatically recalculated each time a task is updated.
- Using the **Maximum** roll-up method, the latest date out of a project's tasks can be captured.
- Using the **Minimum** roll-up method, the earliest date out of a project's tasks can be captured.
- Using the **Sum** roll-up method, the total cost of a project's tasks can be easily determined and automatically recalculated each time a Task is updated.

Example Using Project and Task User Data Roll-Up

A company needs to capture the total cost for a testing project. Total project cost in this case is to be calculated by adding the costs of individual tasks. User data fields for task cost and project total cost are each defined. The relationship is illustrated in *Figure 8-4*.

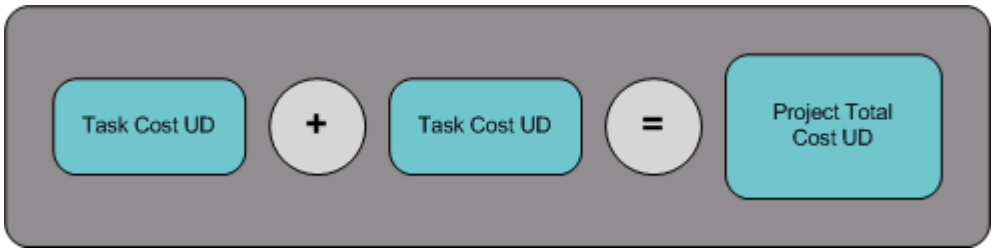
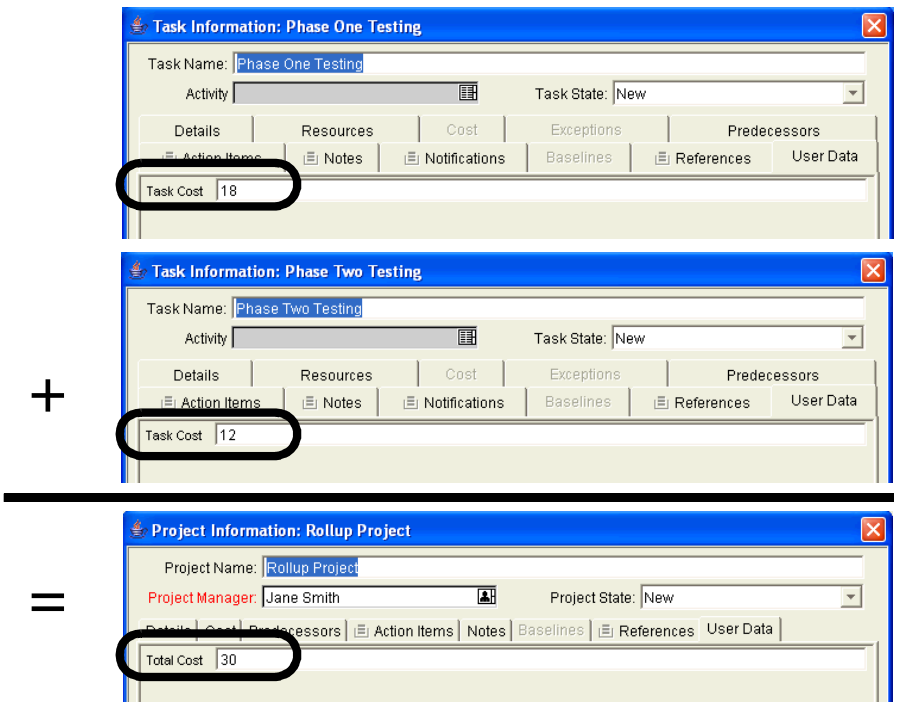


Figure 8-4. Project and task example

Each task has its own Cost User Data field (Task Cost). The values for each Task Cost User Data field are rolled up using the **Sum** roll-up method into the Project Total Cost User Data field (Total Cost). *Figure 8-4* illustrates the project's **User Data** tab and two of the project's task **User Data** tabs.



Overview of Configuring User Data Roll-Ups

User Data must be configured for the project and tasks before specifying user data roll-up methods. The following lists the main steps required to configure user data roll-ups:

To configure user data roll-ups:

1. Configure the Task User Data field.
2. Configure the Project User Data field.
3. Configure the User Data Roll-Up Method.

Configuring Task User Data for User Data Roll-Ups

Only two User Data fields of the same type can be selected for user data roll-up. For example, a Numeric text field cannot roll up into a Date field.

While a Task User Data field can have multiple user data roll-up relationships associated with it, a Project User Data field can have only one user data roll-up relationship defined.

To configure task user data for user data roll-ups:

1. Open the User Data Workbench.

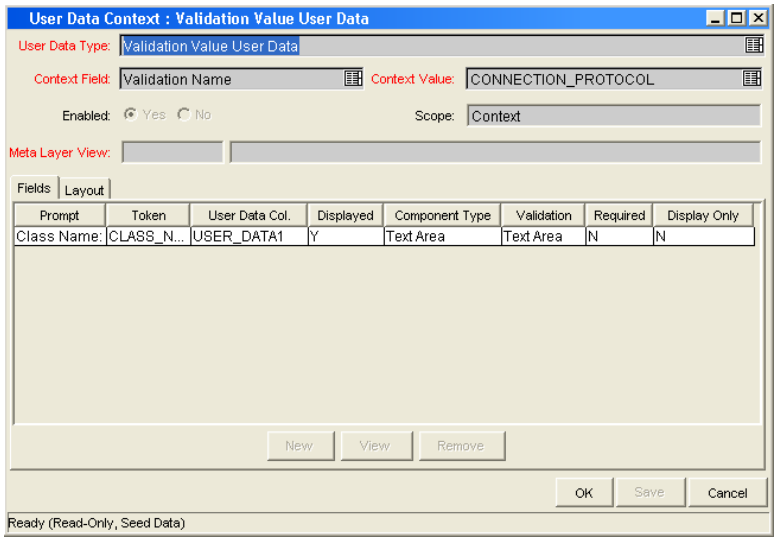
   To open the User Data Workbench, see *Opening the User Data Workbench* on page 273. The User Data Workbench window opens.

2. From the User Data Workbench, open Task User Data.

   The User Data Context window opens. The **Fields** tab is displayed.

3. In the **Fields** tab, select **New.**

   The **Fields** window opens.



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

Field	Description
Field Prompt	The prompt visible for the user data field in the request. For example: Task Cost.
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 USER_DATA_TASK_COST.
Description	A description of the user data field.
Enabled	Select **Yes.**
Validation	Selects the validation. The validation must be the same as the validation for the Project User Data field. The validation must be one of the following: • Numeric fields (Text field component type with numeric data mask) • Date fields
Component Type	Automatically set by the validation type.
Multiselect	Automatically set by the validation type.

5. In the Field window, select **OK.**

   The Field window closes. The new field is added to the User Data Context window.

6. In the User Data Context window, click **OK.**

   The changes to the user data type are saved.

Configuring Project User Data for User Data Roll-Ups

Only two User Data fields of the same type can be selected for user data roll-up. For example, a Numeric text field cannot roll up into a Date field.

While a Task User Data field can have multiple user data roll-up relationships associated with it, a Project User Data field can have only one user data roll-up relationship defined.

To configure project user data for user data roll-ups:

1. Open the User Data Workbench.

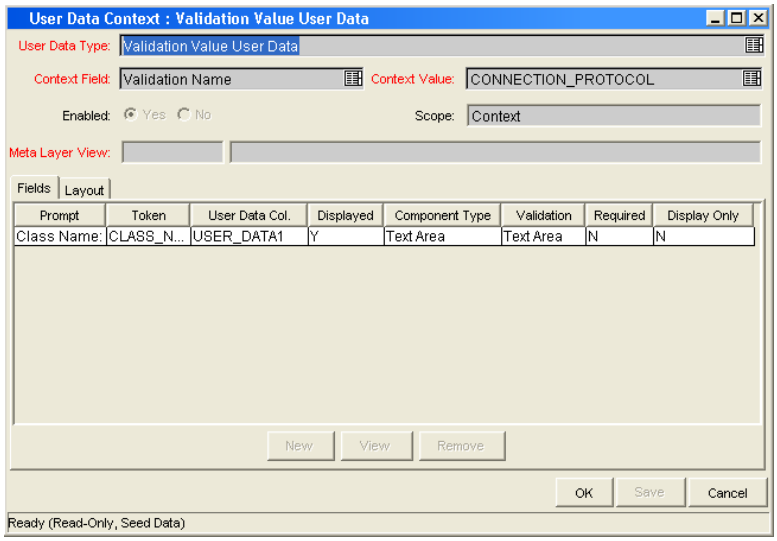
   To open the User Data Workbench, see *Opening the User Data Workbench* on page 273. The User Data Workbench window opens.

2. From the User Data Workbench, open Project User Data.

   The User Data Context window opens. The **Fields** tab is displayed.

3. In the **Fields** tab, select **New.**

   The **Fields** window opens.



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

Field	Description
Field Prompt	The prompt visible for the user data field in the request. For example: Total Cost.
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 USER_DATA_PROJECT_TOTAL_COST.
Description	A description of the user data field.
Enabled	Select **Yes.**
Validation	Selects the validation. The validation must be the same as the validation for the Task User Data field. The validation must be one of the following: • Numeric fields (Text field component type with numeric data mask) • Date fields
Component Type	Automatically set by the validation type.
Multiselect	Automatically set by the validation type.

5. In the Field window, select **OK.**

   The Field window closes. The new field is added to the User Data Context window.

6. In the User Data Context window, click **OK.**

   The changes to the user data type are saved.

Configuring User Data Roll-Ups

Only two User Data fields of the same type can be selected for user data roll-up. For example, a Numeric text field cannot roll up into a Date field.

While a Task User Data field can have multiple user data roll-up relationships associated with it, a Project User Data field can have only one user data roll-up relationship defined.

To configure user data roll-ups:

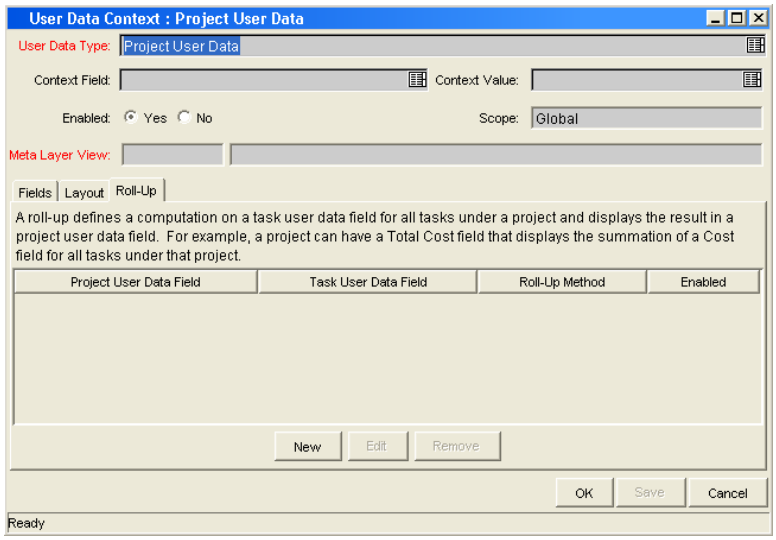
1. Open the User Data Workbench.

   To open the User Data Workbench, see *Opening the User Data Workbench* on page 273. The User Data Workbench window opens.

2. From the User Data Workbench, open Project User Data.

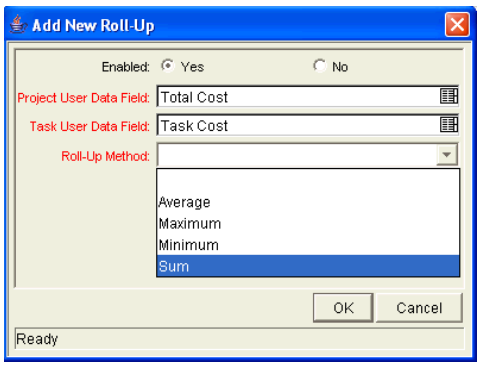
3. In the Project User Data window, select the **Roll-Up** tab.

   The **Roll-Up** tab opens.



4. In the **Roll-Up** tab, click **New.**

   The Add New Roll-Up window opens.



5. Complete the fields in the Add New Roll-Up window as specified in the following table:

Field	Description
Enabled	Makes the user data roll-up method available to the system. **Yes** makes the user data roll-up method available to the system.
Project User Data Field	Selects from the available Project User Data fields. The Project User Data field must already exist. The Project User Data field validation type must be the same as the Task User Data field validation type. For example, • Numeric Task User Data fields cannot roll-up to a date Project User Data field. • Date Task User Data fields cannot roll-up to a numeric Project User Data field.
Task User Data Field	Selects from the available Task User Data fields. The Task User Data field must already exist. The Task User Data field validation type must be the same as the Project User Data field validation type. For example, • Numeric Task User Data fields cannot roll-up to a date Project User Data field. • Date Task User Data fields cannot roll-up to a numeric Project User Data field.
Roll-Up Method	Selects the method of the user data roll-up. The following lists the types of user data roll-up: • **Average.** Shows the average of all values of a specified Task User Data field for every task under the project (numeric fields). The output is displayed in the specified Project User Data field. • **Maximum.** Shows the largest of all values of a specified Task User Data field for every task under the project (numeric and date fields). The output is displayed in the specified Project User Data field. • **Minimum.** Shows the smallest of all values of a specified Task User Data field for every task under the project (numeric and date fields). The output is displayed in the specified Project User Data field. • **Sum.** Shows the summation of all values of a specified Task User Data field for every task under the project (numeric fields). The output is displayed in the specified Project User Data field.

6. In the Add New Roll-Up window, click **OK.**

   The Add New Roll-Up window closes. The user data roll-up relationship is added to the **Roll-Up** tab.

7. In the **Roll-Up** tab, click **Save.**

   The changes to the user data type are saved.

Editing User Data Roll-Ups

To edit an existing user data roll-up:

1. Open the User Data Workbench.

   To open the User Data Workbench, see *Opening the User Data Workbench* on page 273. The User Data Workbench window opens.

2. From the User Data Workbench, open Project User Data.

3. In the Project User Data window, select the **Roll-Up** tab.

   The **Roll-Up** tab opens.

4. In the **Roll-Up** tab, select the user data roll-up and click **Edit.**

   The Edit New Roll-Up window opens.

5. In the Edit New Roll-Up window, edit the user data roll-up.

   For details on the fields of the Edit New Roll-up window, see *Configuring User Data Roll-Ups* on page 298.

6. In the Add New Roll-Up window, click **OK.**

   The Add New Roll-Up window closes.

7. In the **Roll-Up** tab, click **Save.**

   The changes to the user data type are saved.

Deleting User Data Roll-Ups

To delete an existing user data roll-up:

1. Open the User Data Workbench.

   To open the User Data Workbench, see *Opening the User Data Workbench* on page 273. The User Data Workbench window opens.

2. From the User Data Workbench, open Project User Data.

3. In the Project User Data window, select the **Roll-Up** tab.

   The **Roll-Up** tab opens.

4. In the **Roll-Up** tab, select the user data roll-up and click **Remove.**

   The user data roll-up method is removed.

5. In the **Roll-Up** tab, click **Save.**

   The changes to the user data type are saved.

Chapter

9

Rolling Out a Request Tracking and Resolution System

In This Chapter:

Testing the Request Resolution System - Checklists

This section provides a series of high-level checklists to help you validate the system before rolling it out to the users.

General Configuration Checklist

The following items have to be configured to enable your request tracking and resolution system. See the referenced sections in the Notes column for additional details and instructions on configuring each of the entities.

Table 9-1. General configuration checklist

Done	Entity Defined?	Notes
	Workflow	One or more workflows that will be used to process the requests must be defined. See the following sections for details on workflow construction: • *Configuring Workflows* on page 43 • *Configuring Workflow Components* on page 117
	Request Types	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. See the following sections for details on request type construction: • *Configuring Request Types and Request Header Types* on page 155 • *Commands, Tokens, and Validations Guide and Reference*
	Security Groups/User Access	Define the security groups used to control different aspects of the deployment process: request creation, request processing, and request resolution system configuration. See the following sections for details on security group and user participant definition: • *Security Model Guide and Reference*
	Dashboard/Portlets	Decide which portlets can be added to the Dashboard. If none of the default system portlets suit your business needs, construct your own custom portlets. See the following sections for details on portlet construction and Default Dashboard creation: • *Configuring the Standard Interface*

Table 9-1. General configuration checklist [continued]

Done	Entity Defined?	Notes
	Notifications	Define the notifications used in your process. See the following sections for details: • *Configuring Notification Templates* on page 253
	User Data	Define the user data fields used in your process. See the following sections for details: • *Configuring User Data* on page 269

Workflow Checklist

Table 9-2. Workflow configuration checklist

Done	Workflow Check Item	Notes
	Business process is modeled on the Workflow	Execution, decision and condition steps have been added to the **Layout** tab on the Workflow window. See the following sections for details: • *Configuring Workflows* on page 43
	Decision steps set	See the following sections for details: • *Configuring Workflows* on page 43 • *Configuring Workflow Components* on page 117
	Timeouts are set	Timeout values have been placed on how long workflow steps can remain in a single state, and timeouts have added to command executions. This ensures that the process is not delayed from lack of user action or complications during executions. See the following sections for details: • *Configuring Workflows* on page 43 • *Configuring Workflow Components* on page 117
	Automatic transitions are properly set	Ensure that the request will not become “stuck” in a step. This can happen when the results of an execution or query yield a result that is not linked to a transition out of the step. See the following sections for details: • *Configuring Workflows* on page 43
	Manual transitions are set	Ensure that the step has a transition path for each available decision result. See the following sections for details: • *Configuring Workflows* on page 43 • *Configuring Workflow Components* on page 117
	Notifications are set on appropriate Workflow steps	Configure notifications to be sent at specific points in the process. See the following sections for details: • *Configuring Workflows* on page 43 • *Configuring Workflow Components* on page 117
	Includes a Close step.	The process should conclude with a “Closed” request. See the following for details: • *Configuring Workflows* on page 43
	Verify the Workflow	Use the workflow’s **Verify** tool to ensure that serious configuration errors were not made. The workflow verification tool checks for the possible configuration errors described in *Table 9-3* on page 307. See the following for details: • *Configuring Workflows* on page 43

Table 9-3. Workflow logical guidelines

Guideline	Returns	Reason
Workflow should have at least one step.	Error	No processing can be done if the workflow has no steps.
Workflow should have at least one Close step.	Error	The request cannot be closed without a Close step in the workflow.
Each enabled workflow step should have at least one incoming transition	Error	It is not possible to flow to a workflow step without an incoming transition.
Each decision step should have at least one security group, user or token defined in the **Security** tab.	Error	No one is authorized to act on the step without a security group.
Each manual execution step should have at least one security group, user or token defined in the **Security** tab.	Error	No one is authorized to act on the step without a security group.
First workflow step should not be a condition.	Error	Workflow processing may not be correct if the first step is a condition.
A condition step should not have a transition to itself.	Error	A condition with a transition to itself could cause the workflow to run indefinitely.
Transition value is not a valid validation value (error).	Error	The validation value has changed since the transition has been made.
Close steps should not have a transition on **Success** or **Failure.** Return steps should have no outgoing transitions.	Error	The request will not close if a transition exists on **Success.**
An immediate execution step should not have a transition to itself on **Success** or **Failure.**	Error	The workflow could loop indefinitely.
Other Values and **All Values** transitions should not exist at the same step.	Warning	Other Values transition is always ignored if an **All Values** transition exists.
Each workflow step should have at least one outbound transition.	Warning	The branch of the workflow stops indefinitely without closing the request.
Each value from a list-validated validation should have an outbound transition.	Warning	There are validation values that do not have transitions defined.
Step with text or numeric validation should have an **Other Values** or **All Values** transition.	Warning	Since text and numeric validations are not limited, an **Other Values** or **All Values** transition should be defined.
All steps should be enabled.	Warning	Disabled steps cannot be used by a request.

Table 9-3. Workflow logical guidelines [continued]

Guideline	Returns	Reason
AND or OR condition step should have at least two incoming transitions.	Warning	An AND or OR condition with only one incoming transition will always immediately be true and have no effect.
Subworkflow should have at least one Return step.	Error	Should include a Return step.
Notifications with reminders should not be set on results that have transitions.	Error	Transition into the Return step does not match the validation.
Close step in subworkflow will close entire request.	Warning	Has a Close step.
Top level workflow should not have a Return step.	Error	Only subworkflows have a Return step.
Request status for a step not linked to a request type that uses this workflow.	Warning	Request cannot handle status.

Request Type Checklist

Table 9-4. Request type configuration checklist

Done	Request Type Check Item	Notes
	Request Header Type associated	A request header type should be associated with the request type. If no satisfactory request header type exists, a new one can be created. See the following sections for details: • *Configuring Request Types and Request Header Types* on page 155
	Fields defined	Fields are required to define the request. Ensure the correct parameters are used to describe the request to be processed. See the following sections for details: • *Configuring Request Types and Request Header Types* on page 155 • *Commands, Tokens, and Validations Guide and Reference*
	Request Rules defined	Rules can be set for the automatic population of fields in the request. See the following sections for details: • *Configuring Request Types and Request Header Types* on page 155
	Request Statuses 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. See the following sections for details: • *Configuring Request Types and Request Header Types* on page 155
	Status Dependencies set	Request fields can be configured to be hidden, required, read-only, cleared, or reconfirmed based on the status of the request. See the following sections for details: • *Configuring Request Types and Request Header Types* on page 155
	Request security set	It is possible to exercise a great deal of control over who can participate in your request resolution process. See the following sections for details: • *Configuring Request Types and Request Header Types* on page 155 • *Security Model Guide and Reference*
	Request field security set	Request fields can be configured to be hidden to particular users or security groups. See the following sections for details: • *Configuring Request Types and Request Header Types* on page 155 • *Security Model Guide and Reference*

Table 9-4. Request type configuration checklist [continued]

Done	Request Type Check Item	Notes
	Notification	Notifications can be configured to be sent automatically at various points in your process. See the following sections for details: • *Configuring Notification Templates* on page 253
	User data fields	User data fields can be used to track specific information across a Workbench entity. See the following sections for details: • *Configuring User Data* on page 269

Security/User Access Checklist

Table 9-5. Security/User access configuration checklist

Done	Security/User Access Check Item	Notes
	Created Security Groups (for access to screens and functions)	Security groups have been created to grant access to certain screens and functions. See the following sections for details: • *Security Model Guide and Reference*
	Created Security Groups (for association with workflow steps)	security groups have been created to allow users to act on a specific workflow step. See the following sections for details: • *Security Model Guide and Reference*
	Set security on Request Creation	All available options have been set for restricting who can create and submit requests. See the following sections for details: • *Security Model Guide and Reference* • *Configuring Request Types and Request Header Types* on page 155
	Set security on Request processing	All available options have been set for restricting who can process requests. See the following sections for details: • *Security Model Guide and Reference* • *Configuring Request Types and Request Header Types* on page 155
	Set security on Request field visibility	All available options have been set for restricting who can view or edit particular request fields. See the following sections for details: • *Configuring Request Types and Request Header Types* on page 155 • *Security Model Guide and Reference*
	Set security on Request resolution system configuration	Specified who can modify the request resolution process. This includes editing the workflow, request type, security groups, and so on. See the following sections for details: • *Configuring Request Types and Request Header Types* on page 155 • *Security Model Guide and Reference*

Dashboard/Portlet Checklist

Table 9-6. Dashboard/Portlet configuration checklist

Done	Dashboard Check Item	Notes
	Configure the display columns in request types	Advanced users can configure which columns are displayed in a request. See the following sections for details: *Configuring Request Types and Request Header Types* on page 155
	Created custom portlets to display desired data	Advanced users with a knowledge of SQL programming can create their own Dashboard. See the following sections for details: • *Configuring the Standard Interface*
	Enable portlets for use on the Dashboard	See the following sections for details: • *Configuring the Standard Interface*
	Specify which users can add use certain portlets	See the following sections for details: • *Configuring the Standard Interface*
	Create a default Dashboard or distribute a Dashboard to users.	See the following sections for details: • *Configuring the Standard Interface*

Cross Entity Checklist

Table 9-7. Cross entity configuration checklist

Done	Entities	Configuration Considerations
	Request Header Type and Request Type	The following items should be coordinated between the request header type and request type: • Decide which request header type will be used with the request type.
	Workflow and Request Type	The following items should be coordinated between the workflow and request type: • Decide which request type statuses correspond to which workflow steps. • Decide which workflow steps will execute the request type commands, if any. • workflow step source validations and request type field validations are in agreement. This is required when transitioning based on a field value (using token, SQL or PL/SQL execution types) • Allow the request type use for the workflow (set in the workflow window - **Request Types** tab). • Allow the workflow to be used by the request type (set in the request type window - **workflows** tab).
	Workflow and Security Groups	The following items should be coordinated between the workflow and security groups: • Associate security groups with workflow steps. Users in the included groups can act on the step. • Set workflow and workflow step ownership.
	Security Groups and other entities (request types, environments, and so on.)	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.

Enabling Entities and User Access

Each entity used in the request resolution process includes an Enabled parameter. This parameter needs to set to **Yes** in order to provide general access. Ensure that the following entities are enabled in the system:

- Workflows (including subworkflows)
- Request Types
- Request Header Types
- Security Groups
- Users
- Portlets

Educating Your Users

The final step in rolling out the system is training the users. This includes the educating the users on the following activities:

- **Basic product use.** Creating, processing, and reporting on requests.
- **Process-specific training.** Ensure that each of the users understands the general request process. Plan a formal roll-out meeting or publish documents on the configurations and processes at the site.
- **User Responsibilities.** Ensure that each user understands their individual role in the process. Also, take advantage of the product's email notification functionality. The notifications can be very specific, instructing individual users with their required actions.

Appendix A
Worksheets

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				
* Type = Workflow Step Type: Decision (D), Execution (E), Condition (C), Subworkflow (S)				

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, auto-complete, drop-down list, and so on.)*	
Validation Definition (list of values or SQL)	

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

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

Decision Workflow Step Worksheets

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

	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, drop-down list, and so on.)*	
Validation Definition (list of values or SQL)	

Subworkflow Workflow Step Worksheets

Table A-7. Workflow step [subworkflow], step number ____

	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	
Notification Recipient: • Username • Email Address • Security Group • Standard Token • User Defined Token	
Notification Message	
Request Status at Step	
Request % Complete at Step	
Authentication Required (Y/N) Authentication Type (if Y)	

Table A-8. Workflow step [subworkflow], step number ____ validation

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

Request Type Configuration Sheets

Table A-9. Request type information

	Value
Request Type Name	
Associated Request Header Type	
Description	

Table A-10. Request type field information

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

Table A-11. Request type commands

Goal of Commands	
Command Steps	
Conditions (When to execute)	

Table A-12. Request type statuses

Status	Corresponds to Workflow Step

Table A-13. Request type field information

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 to populate Field with:	

Table A-14. Field validation information

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

Table A-15. Request header type information

	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

V

W