

KINTANA™

# Configuring a Request Resolution System

**Version 5.0.0**

Publication Number: RequestConfig-0603A

Kintana, Inc. and all its licensors retain all ownership rights to the software programs and related documentation offered by Kintana. Use of Kintana's software is governed by the license agreement accompanying such Kintana software. The Kintana software code is a confidential trade secret of Kintana and you may not attempt to decipher or decompile Kintana software or knowingly allow others to do so. Information necessary to achieve the interoperability of the Kintana software with other programs may be obtained from Kintana upon request. The Kintana software and its documentation may not be sublicensed and may not be transferred without the prior written consent of Kintana.

Your right to copy Kintana software and this documentation is limited by copyright law. Making unauthorized copies, adaptations, or compilation works (except for archival purposes or as an essential step in the utilization of the program in conjunction with certain equipment) is prohibited and constitutes a punishable violation of the law.

THIS DOCUMENTATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND. IN NO EVENT SHALL KINTANA BE LIABLE FOR ANY LOSS OF PROFITS, LOSS OF BUSINESS, LOSS OF USE OR DATA, INTERRUPTION OF BUSINESS, OR FOR INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY KIND, ARISING FROM ANY ERROR IN THIS DOCUMENTATION.

Kintana may revise this documentation from time to time without notice.

Copyright © 1997, 1998, 1999, 2000, 2001, 2002, 2003 Kintana, Incorporated. All rights reserved.

Kintana, Kintana Deliver, Kintana Create, Kintana Drive, Kintana Dashboard, Kintana Accelerator, Kintana Demand Management (DM), Kintana Portfolio Management (PFM), Kintana Program Management Office (PMO), Kintana Enterprise Change Management (ECM), Object\*Migrator, GL\*Migrator and the Kintana logo are trademarks of Kintana, Incorporated. All other products or brand names mentioned in this document are the property of their respective owners.

Kintana Version 5.0.0

© Kintana, Incorporated 1997 - 2003

All rights reserved.

Printed in USA

**Kintana, Inc.**

1314 Chesapeake Terrace, Sunnyvale, California 94089

Telephone: (408) 543-4400

Fax: (408) 752-8460

<http://www.kintana.com>

# Contents

<b>Chapter 1</b>	
<b>Introduction</b> .....	<b>1</b>
<b>Who Should Read This Guide</b> .....	<b>2</b>
<b>How to Use This Guide</b> .....	<b>2</b>
What This Guide is NOT .....	2
<b>Additional Resources</b> .....	<b>3</b>
Kintana Documentation .....	3
<i>Kintana Business Application Guides</i> .....	4
<i>User Guides</i> .....	4
<i>Kintana Application Reference Guides</i> .....	4
<i>Kintana Instance Administration Guides</i> .....	5
<i>External System Integration Guides:</i> .....	5
<i>Kintana Solution Guides</i> .....	5
<i>Kintana Accelerator Guides</i> .....	6
Kintana Services .....	6
Kintana Education .....	6
Kintana Support .....	6
<b>Chapter 2</b>	
<b>Key Concepts</b> .....	<b>9</b>
<b>Request Resolution</b> .....	<b>9</b>
<b>Request</b> .....	<b>10</b>
<b>Request Type</b> .....	<b>11</b>
<b>Request Header Type</b> .....	<b>11</b>
<b>Field Logic</b> .....	<b>12</b>
<b>Workflow</b> .....	<b>13</b>
<b>Request/Workflow Interaction</b> .....	<b>13</b>
<b>Request Type Commands</b> .....	<b>14</b>
Special Commands .....	14
<b>Validation</b> .....	<b>15</b>
<b>Token</b> .....	<b>15</b>
<b>Security Groups</b> .....	<b>16</b>

---

<b>Participants</b> .....	<b>18</b>
<b>Integration with Kintana Products and Solutions</b> .....	<b>18</b>
Kintana Solutions .....	18
Kintana Dashboard .....	19
Kintana Drive.....	20
Kintana Deliver .....	20
Kintana Accelerators .....	20
<b>Reports</b> .....	<b>21</b>
<b>Chapter 3</b>	
<b>Using Migrators to Develop your Kintana Configurations</b> .....	<b>23</b>
<b>Using Multiple Kintana Instances - Overview</b> .....	<b>24</b>
<b>Single PRODUCTION instance is currently in use.</b> .....	<b>25</b>
New Kintana Implementation .....	25
<b>Migrating your Kintana Configurations</b> .....	<b>26</b>
How Kintana Migrators Work .....	26
Using the Kintana Migrators - Overview .....	27
Instance Requirements for Using Kintana Migrators.....	28
Kintana Requirements for PROD Instance.....	28
<b>Archiving your Kintana Configurations</b> .....	<b>29</b>
<b>Chapter 4</b>	
<b>Configuring your Request Resolution System - Process Overview</b> .....	<b>31</b>
<b>Example: Configuring a Request Resolution System</b> .....	<b>32</b>
<b>Chapter 5</b>	
<b>Gathering Process Requirements and Specifications</b> .....	<b>35</b>
<b>Identify Needed Entities</b> .....	<b>36</b>
Workflows.....	36
Request Type.....	36
Request Header Type.....	37
Security Groups .....	37
<b>Gather Requirements for Workflow</b> .....	<b>37</b>
Defining the Business Flow .....	38
Example: Defining the Business Flow .....	38
<b>Business Process Overview</b> .....	<b>38</b>
Gather Information on Each Step in the Process .....	39
Consider Using Subworkflows .....	40
Example: Using a Subworkflow .....	40
Consider Request Statuses.....	41
<b>Gather Requirements for Request Type</b> .....	<b>42</b>
Request Type Fields .....	42

Example: ACME collects information for the software change Request .....	43
Request-Workflow Interaction .....	44
Request Header Type .....	44
Request Type Commands .....	45
<b>Identify Participants and Security .....</b>	<b>45</b>
Security for Workflows .....	46
Security for Request Types .....	46
Security Around Request Fields .....	47
Configuration Security .....	47
Example: ACME Determines Participants and Security .....	47
<b>Establish Communication Points and Visibility .....</b>	<b>52</b>
Notifications on Workflow Steps .....	52
Example: ACME configures notifications .....	53
Notifications on Field Changes .....	54
<b>Chapter 6</b>	
<b>Mapping your Process into a Kintana Workflow .....</b>	<b>55</b>
<b>Building the Workflow Skeleton - Overview .....</b>	<b>55</b>
<b>Required Workflow Settings for Request Resolution Process .....</b>	<b>56</b>
<b>Create the Required Step Source .....</b>	<b>56</b>
Creating a Workflow Step Source - Overview .....	58
Workflow Step Source Configuration and Usage Restrictions .....	60
Creating a Decision Type Step .....	61
Enter the general information on the Decision step source .....	61
Select a Validation .....	63
Specify the voting requirements on the step .....	63
Specify the default timeout value .....	64
Create an Execution Type Step .....	65
Enter the general information on the Execution step source .....	66
Define the Executions .....	68
Execute the Request Type Commands .....	69
Close the Request and mark it as a Success .....	71
Close the Request and mark it as Failed .....	72
Transition (jump) to a Workflow that is Processing a Package .....	74
Receive control from a Workflow that is Processing a Package .....	74
Return from a Subworkflow to the Parent Workflow .....	74
Execute a PL/SQL function and then transition based on the result .....	75
Execute a SQL statement and then transition based on the result .....	76
Evaluate a Token and then transition based on the result .....	77
Execute a number of system level commands and then transition based on the success or failure of those commands.78	78
Select a Validation .....	81
Specify the default timeout value .....	81
<b>Configure the Step's Transition Values (Validation) .....</b>	<b>81</b>
Validations and Execution Type relationships .....	83

<b>Add Steps and Transitions to the Workflow Layout .....</b>	<b>84</b>
Adding Decision Steps.....	84
Enter the general information on the Decision step .....	85
Specify the Security.....	87
Configure Notifications for the Workflow Step .....	88
Adding Execution Steps .....	89
Enter the general information on the Execution step.....	89
Specify the Security.....	92
Configure Notifications for the Workflow Step .....	93
Adding a Subworkflow .....	93
Adding Transitions Between Steps .....	94
Transition based on a specific result .....	95
Transition based on a value in a field .....	96
Transition based on data in a table.....	98
Transition based on all but one specific value .....	98
Transition based on all results.....	99
Transition based on error .....	100
Transition back to the same step .....	102
Transition based on a previous workflow step result (parameters) .....	103
<i>Example: Using a Workflow Parameter to Transition.....</i>	<i>103</i>
Transition to and from Subworkflows.....	106
Transition to and from a Package Workflow .....	106

## Chapter 7

<b>Constructing the Request Type .....</b>	<b>107</b>
<b>Creating a Request Type - Overview .....</b>	<b>107</b>
<b>Choosing a Request Header Type.....</b>	<b>109</b>
Creating a New Request Header Type .....	111
Modifying Existing Request Header Type Fields.....	112
Creating New Request Header Type Fields .....	115
Copying a Request Header Type .....	115
<b>Request Type Field Validations.....</b>	<b>117</b>
Determining the Field Type (Selecting a Validation).....	118
Available Field Types.....	118
Selecting the Validation .....	120
Building a Validation .....	121
<i>Tips for Configuring Validations.....</i>	<i>121</i>
<b>Configuring Field Behavior - Overview .....</b>	<b>122</b>
Visibility .....	122
Editability .....	124
Defaulting.....	124
Required/Reconfirm .....	125
Notifications .....	125
<b>Creating a Field.....</b>	<b>126</b>
Copying a Request Type Field .....	132

Removing Request Type Fields .....	133
Setting the Number of Maximum Fields for a Request Type .....	134
<b>Configuring Request Type Defaulting Behavior (Rules) .....</b>	<b>135</b>
Configuring Simple Default Rules .....	137
Creating a Simple Default Rule .....	138
Example: ACME Defaults Software Change Workflow .....	140
Configuring Advanced Default Rules .....	141
Creating an Advanced Default Rule .....	143
<b>Configuring Field Behavior Using Status Dependencies.....</b>	<b>146</b>
Creating Your Request Statuses .....	147
Adding and Linking a Request Status .....	148
Configuring Field Status Dependency Behavior .....	150
Status Dependencies - Visible .....	151
Status Dependencies - Required Field .....	151
Status Dependencies - Updateable Field .....	152
Status Dependencies - Reconfirm Field .....	152
Status Dependencies - Clear Field .....	152
Status Dependencies Interactions.....	153
Assigning Request Statuses to Workflow Steps .....	153
<b>Modifying the Request Type Layout .....</b>	<b>154</b>
Modifying the Request Type Field Width .....	155
Moving Fields in a Request Type .....	155
Adding Sections to the Request Type .....	156

## Chapter 8

<b>Integrating Participants into Your Request Resolution System .....</b>	<b>159</b>
<b>User Security and Participation - Overview .....</b>	<b>159</b>
<b>Establishing Security Groups.....</b>	<b>161</b>
Creating a Security Group by Specifying a List of Users .....	161
Using Kintana's Resource Management to Control User Security .....	164
<b>Setting Request Creation Security .....</b>	<b>166</b>
Enabling Users to Create Requests .....	166
Restricting Users from Selecting a Specific Workflow .....	170
Restricting Users from Selecting a Specific Request Type .....	171
<b>Setting Request Processing Security .....</b>	<b>172</b>
Providing Users with General Access to Update Requests .....	172
Enabling Users to Act on a Specific Workflow Step .....	175
Restricting Request Processing to Participants .....	176
<b>Setting Configuration Security .....</b>	<b>177</b>
Setting Ownership for Kintana Configuration Entities .....	178
Removing Access Grants.....	180

---

<b>Chapter 9</b>	
<b>Setting Up Communication Paths .....</b>	<b>183</b>
<b>Adding Notifications to Workflow Steps .....</b>	<b>184</b>
Adding a Notification to a Workflow step - Overview .....	184
Configuring When to Send a Notification .....	186
Sending a notification when a step becomes eligible .....	186
Sending a notification when a step has a specific result .....	187
Sending a notification when the step has a specific error .....	189
<i>Specific Errors for Workflow Steps .....</i>	<i>190</i>
Configuring multiple notifications for a single step .....	191
Specifying the Time the Notification is Sent .....	192
<i>Configuring the Notification Intervals .....</i>	<i>193</i>
Sending a follow up notification (reminder) .....	195
Configuring the Notification Recipients .....	196
Recipient Configuration Tips .....	198
Configuring the Notification Message .....	198
Using Tokens in the Message Body .....	200
Including URLs to Open the Request (Smart URLs) .....	200
<i>Smart URLs in HTML Formatted Messages .....</i>	<i>201</i>
<b>Setting Notifications on Request Field Changes .....</b>	<b>202</b>
<b>Configuring Your Dashboard .....</b>	<b>207</b>
Controlling User Access to Portlets .....	207
Disabling Portlets .....	207
Restricting User Access .....	209
Creating and Distributing a Default Dashboard .....	210
Creating Custom Portlets .....	211
<b>Configuring Reports .....</b>	<b>211</b>
<b>Appendix A</b>	
<b>Advanced Workflow Topics .....</b>	<b>213</b>
<b>Using Subworkflows .....</b>	<b>213</b>
Transitioning to a Subworkflow .....	214
Transitioning From a Subworkflow .....	216
<b>Package - Request Workflow Integration .....</b>	<b>218</b>
Setting Up the 'WF - Jump/Receive Step Labels' Validation .....	220
Generating a Jump Step Source .....	222
Generating a Receive Step Source .....	223
Including the Jump/Receive pair in Workflows .....	225
<b>Using Condition Steps .....</b>	<b>227</b>
AND .....	227
OR .....	228
SYNC .....	228
FIRST LINE .....	229
LAST LINE .....	231



<b>Setting the Reopen Step for Request Workflows .....</b>	<b>231</b>
<b>Modifying Workflows in Use.....</b>	<b>232</b>
Copying and Testing the Workflow .....	233
Moving Requests Out of a Step.....	233
Disabling a Workflow Step.....	234
<i>Redirecting the Workflow</i> .....	234
Setting Up Execution Steps .....	235
Modifying Workflow Step Security – Performance Consideration.....	235
Verifying Workflow Logic.....	236
<b>Using Workflow Parameters .....</b>	<b>236</b>
Creating a Workflow Parameter .....	236
Example: Building a Loop Counter .....	237
<b>Appendix B</b>	
<b>Validations .....</b>	<b>243</b>
<b>What are Validations .....</b>	<b>244</b>
<b>Validation Component Types - Overview .....</b>	<b>244</b>
<b>Creating a Validation .....</b>	<b>247</b>
User Data on the Validation Value.....	247
<b>Editing Validations .....</b>	<b>249</b>
Creating a URL to Open the Validation Window.....	250
<b>Deleting Validations .....</b>	<b>251</b>
<b>Static List Validations.....</b>	<b>251</b>
<b>Dynamic List Validations.....</b>	<b>253</b>
SQL Validation .....	253
SQL Validation Tips .....	255
Command Validation .....	255
<b>Using Auto-Complete Validations .....</b>	<b>256</b>
Validation by Command With Delimited Output .....	257
Validation by Command With Fixed Width Output.....	260
User-Defined Multi-Select Auto-Complete Fields.....	262
Example: Token Evaluation and Validation by Command with Delimited Output .....	263
Special Case - Limiting the Number of Returned Rows .....	266
<b>Using Directory and File Choosers .....</b>	<b>268</b>
Directory Chooser .....	269
File Chooser .....	269
<b>Creating 1800 Character Text Areas .....</b>	<b>271</b>
<b>Configuring the Table Component .....</b>	<b>272</b>
Define the Table Component in the Validation Workbench .....	273
Creating a Table Rule .....	276
<b>Example: Using a Table Component on an Order Form .....</b>	<b>277</b>
<i>Tokens in the Table Components</i> .....	281

Calculating Column Totals .....	281
Add the Table Component to a Request Type.....	283
<b>Package and Request Group Validations .....</b>	<b>285</b>
Package and Request Groups .....	285
Request Type Category .....	286
<b>Validation Special Characters .....</b>	<b>287</b>
<b>System Validations .....</b>	<b>287</b>
.....	303
<b>Appendix C</b>	
<b>Tokens .....</b>	<b>305</b>
<b>Chapter 10</b>	
<b>User Data Creation and Processing .....</b>	<b>307</b>
<b>Creating and Editing Kintana User Data .....</b>	<b>308</b>
Adding User Data Fields .....	309
Copying a Field's Definition .....	310
Editing User Data Fields.....	311
Configuring User Data Field Dependencies .....	312
Removing Fields .....	314
Modifying the User Data Layout .....	315
Changing Column Width .....	316
Moving a Field .....	316
Swapping Positions of Two Fields.....	317
Previewing the Layout .....	317
<b>Creating and Editing Context Sensitive User Data .....</b>	<b>318</b>
Creating Context Sensitive User Data .....	319
Defining the Context Field .....	319
Defining a Context Value .....	320
Defining the Context Sensitive Fields.....	321
Editing Context Sensitive User Data .....	321
Changing the Context Field .....	321
Changing the Context Value.....	322
Editing Context Sensitive Fields .....	323
Deleting Context Sensitive User Data.....	323
Copying Context Sensitive User Data .....	323
Example - Using Context Sensitive User Data for a Field in a Request Header Type .....	324
Setting Up the Context Sensitive User Data .....	325
Example: Configuring the Validations .....	326
Example: Modifying the SQL .....	328
Example: Resulting Behavior.....	329
<b>Project/Task User Data Roll-Up.....</b>	<b>331</b>
Creating Project/Task User Data Roll-Up.....	331
Example: Using Project/Task User Data Roll-Up.....	332

---

Editing Project/Task User Data Roll-Up .....	335
Deleting Project/Task User Data Roll-Up .....	337
Example: Creating and Using Project/Task User Data Roll-Up .....	338
<b>Referring to User Data .....</b>	<b>344</b>
<b>Migrating User Data .....</b>	<b>344</b>
Migrating User Data Values .....	344
Migrating User Data Contexts .....	345
<b>Chapter 11 .....</b>	<b>346</b>
<b>Appendix D</b>	
<b>Configuration Worksheets .....</b>	<b>347</b>
<b>Participant and Security .....</b>	<b>356</b>



# Chapter 1 Introduction

Kintana Create allows an organization to model its processes for managing technology initiatives from inception to implementation using a graphical workflow business modeler. Complex business rules can be modeled using approval methods and prioritization features that allow issues to efficiently advance through their specific workflow, routing them to relevant departments, groups or individuals. Kintana Create is designed to capture data by prompting users for information specific to their “Request,” ensuring that required information is collected and validated at the appropriate time in the process.

This document provides instructions for configuring a Request resolution system using Kintana. This includes requirements gathering, modeling your processes in a Kintana Workflow, defining a Request Type to be integrated with the Workflow, and rolling out this system to your users.

This document discusses the following topics:

- *Key Concepts*
- *Configuring your Request Resolution System - Process Overview*
- *Gathering Process Requirements and Specifications*
- *Mapping your Process into a Kintana Workflow*
- *Constructing the Request Type*
- *Integrating Participants into Your Request Resolution System*
- *Setting Up Communication Paths*
- *Rolling Out Your Request Resolution System*

## Who Should Read This Guide

This document provides details for defining, configuring, and rolling out a Request resolution system in Kintana.

This business application guide is used primarily by:

- Business or technical users who configure and maintain a Request resolution system using Kintana (Kintana Create)
- Users responsible for Workflow configuration
- Managers responsible for reporting on Requests



Note

Most Kintana configuration process is performed using the Kintana Workbench interface. Therefore, you must have a Create Power license to access the screens and windows described in this document. You must also belong to a Security Group with the correct access grants in order to define and process Requests. See "[Kintana Security Model](#)" for details.

Additionally, in order to migrate Kintana configurations from one instance to another, the user performing the migrations must have a Kintana Deliver Power License and the proper level of access to at least create and submit Packages.

## How to Use This Guide

This document provides background information and details for configuring Kintana to manage your Request resolution process. Navigate to one of the following chapter topics or use the Index to find information related to key words.

If viewing this guide online, you can press Ctrl-F on your keyboard to search for keywords.

## What This Guide is NOT

This business application guide is not meant to provide detailed information on every screen and field in Kintana, nor is it meant to provide detailed instructions on creating and submitting Requests on a day-to-day basis. For detailed screen and field information refer to the Kintana Application

Reference Guides, accessible from the Kintana Library. See [“Additional Resources”](#) on page 3 for a list of the most relevant documents.

## Additional Resources

Kintana provides the following additional resources to help you successfully implement, configure, maintain and fully utilize your Kintana installation:

- [Kintana Documentation](#)
- [Kintana Services](#)
- [Kintana Education](#)
- [Kintana Support](#)

### *Kintana Documentation*

Kintana product documentation is linked from the Kintana Library page. This page is accessed by:

- Selecting **HELP > KINTANA LIBRARY** from the Kintana Workbench menu.
- Selecting **HELP > CONTENTS AND INDEX** from the menu bar on the HTML interface. You can then click the **KINTANA LIBRARY** link to load the full list of product documents.

Kintana organizes their documents into a number of user-based categories. The following section defines the document categories and lists the documents currently available in each category.

- [Kintana Business Application Guides](#)
- [User Guides](#)
- [Kintana Application Reference Guides](#)
- [Kintana Instance Administration Guides](#)
- [External System Integration Guides:](#)
- [Kintana Solution Guides](#)
- [Kintana Accelerator Guides](#)

### **Kintana Business Application Guides**

Provides instructions for modeling your business processes in Kintana. These documents contain process overviews, implementation instructions, and detailed examples.

- Configuring a Request Resolution System (Create)
- Configuring a Deployment and Distribution System (Deliver)
- Configuring a Release Management System
- Configuring the Kintana Dashboard
- Managing Your Resources with Kintana
- Kintana Reports

### **User Guides**

Provides end-user instructions for using the Kintana products. These documents contain comprehensive processing instructions.

- Processing Packages (Deliver) User Guide
- Processing Requests (Create) User Guide
- Processing Projects (Drive) User Guide
- Navigating the Kintana Workbench:  
Provides an overview of using the Kintana Workbench
- Navigating Kintana:  
Provides an overview of using the Kintana (HTML) interface

### **Kintana Application Reference Guides**

Provides detailed reference information on other screen groups in the Kintana Workbench. Also provides overviews of Kintana's command usage and security model.

- Reference: Using Commands in Kintana
- Reference: Kintana Security Model
- Workbench Reference: Deliver



- Workbench Reference: Configuration
- Workbench Reference: Create
- Workbench Reference: Dashboard
- Workbench Reference: Sys Admin
- Workbench Reference: Drive
- Workbench Reference: Environments

### **Kintana Instance Administration Guides**

Provides instructions for administrating the Kintana instances at your site. These documents include information on user licensing and archiving your Kintana configuration data.

- Kintana Migration
- Kintana Licensing and Security Model

### **External System Integration Guides:**

Provides information on how to use Kintana's open interface (API) to access data in other systems. Also discusses Kintana's Reporting meta-layer which can be used by third party reporting tools to access and report on Kintana data.

- Kintana Open Interface

### **Kintana Solution Guides**

Provides information on how to configure and use functionality associated with the Kintana Solutions. Each Kintana Solution provides a User Guide for instructions on end-use and a Configuration Guide for instructions on installing and configuring the Solution.

## Kintana Accelerator Guides

Provides information on how to configure and use the functionality associated with each Kintana Accelerator. Kintana Accelerator documents are only provided to customers who have purchased a site-license for that Accelerator.



Note

Kintana provides documentation updates in the Download Center section of the Kintana Web site ([http://www.kintana.com/support/download/download\\_center.htm](http://www.kintana.com/support/download/download_center.htm)).

A username and password is required to access the Download Center. These were given to your Kintana administrator at the time of product purchase. Contact your administrator for information on Kintana documentation or software updates.

## Kintana Services

Kintana is a strategic partner to its clients, assisting them in all aspects of implementing a Kintana technology chain - from pilot project to full implementation, education, project turnover, and ongoing support. Our Total Services Model tailors solution and service delivery to specific customer needs, while drawing on our own knowledgebank and best practices repository. Learn more about Kintana Services from our Web site:

<http://www.kintana.com/services/services.shtml>

## Kintana Education

Kintana has created a complete product training curriculum to help you achieve optimal results from your Kintana applications. Learn more about our Education offering from our Web site:

<http://www.kintana.com/services/education/index.shtml>

## Kintana Support

Kintana provides web-based interactive support for all products in the Kintana product suite via Contori.

<http://www.contori.com>

Login to Contori to enter and track your support issue through our quick and easy resolution system. To log in to Contori you will need a valid email address at your company and a password that will be set by you when you register at Contori.



# Chapter 2 Key Concepts

The following key concepts and definitions are used when creating a Request resolution system.

- *Request Resolution*
- *Request*
- *Request Type*
- *Request Header Type*
- *Field Logic*
- *Workflow*
- *Request/Workflow Interaction*
- *Request Type Commands*
- *Validation*
- *Token*
- *Security Groups*
- *Participants*
- *Integration with Kintana Products and Solutions*
- *Reports*

## Request Resolution

Kintana Create lets you model and enforce best practice Request management processes to accelerate Request resolution from inception through

implementation. Possible Requests that can be made are categorized by type. Each type of Request leverages an optimized workflow process tailored for your specific business rules and organization 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 you have the correct information to resolve a Request.
- 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

The Request is the fundamental work unit of Kintana Create. End-users will create Requests and then submit them along a resolution process (defined in the Workflow). The Request page contains all of the information that is typically required to complete a specific business process. Requests with similar or related functions can be grouped into Request Categories, making them easier to locate and use.

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 are prompted for all of the information necessary to bring the Request to closure. Once the basic Request information has been entered, the corresponding Workflow is automatically selected based on the Request Type.



### **A Request:**

- Is the fundamental “work unit” within Kintana Create.
- Is the repository for all of the information necessary to take a series of actions and move through a standard business process.
- Is a specific execution of a business process. Each Request is identified by a unique Request Number.

---

## Request Type

A Request Type is a general category that defines the structure of a Request in Kintana Create. Kintana Create includes such pre-defined system Request Types as the Bug Request Type and Enhancement Request Type. The fields that are used when a Request is created are customizable 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 fields' definition and behavior
- Layout
- Data/access security (who can view or edit the Request)
- Configuration security (who can alter the Request Type)
- Notifications



### **A Request Type:**

- Is the framework that defines the behavior of a Request as it moves through a business process.
- Determines the logic behind (and provides the framework for) the storage and manipulation of data within a Request.
- Represents a different process within a business. The Request Type can be defined to capture different kinds of data and follow different business and resolution processes.

## Request Header Type

Request Header Types define the collection of fields that appear in the header region of the Requests using that Request Type. The presentation and validation of these Header Data fields in a Request Header Type depends upon the type of business process from which the Request Type is gathering information. When creating or configuring a new Request Type, you associate a Request Header Type with that Request Type.

Request Header Types can also contain custom fields.



### **Request Header Type**

A Request Header Type can be thought of as a basic template for the header area that appears at the top of a Request page. Request Header Types have the following characteristics:

- Provides a framework for the storage and manipulation of Request header data.
  - Header data represents attributes common to multiple types of Requests. Header data is useful for locating and reporting certain types of Requests. Examples of Header Data are Creator, Assigned User, Description, Status, and Department.
- Label and arrange header fields in a manner most familiar to specific Business Units. For example, in a Request for novice users, the Workflow, assignment, and contact fields can be hidden.

## **Field Logic**

Many aspects of a Request field can be configured to change:

- A field value can be updated
- A field can be automatically populated
- A field can become invisible or non-editable
- A field can be required or need to be reconfirmed

These changes can in turn be triggered by many different Request-related events:

- Another field in the Request changes
- The Request status changes (based on its Workflow)
- Request Type commands are executed
- Field visibility can be controlled at the user level

The rules that govern automated Request field changes can be configured at the Request Type level or the Field level within the Request Type or Request Header Type.



# Workflow

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.

A sample workflow for an Application Enhancement is shown below:

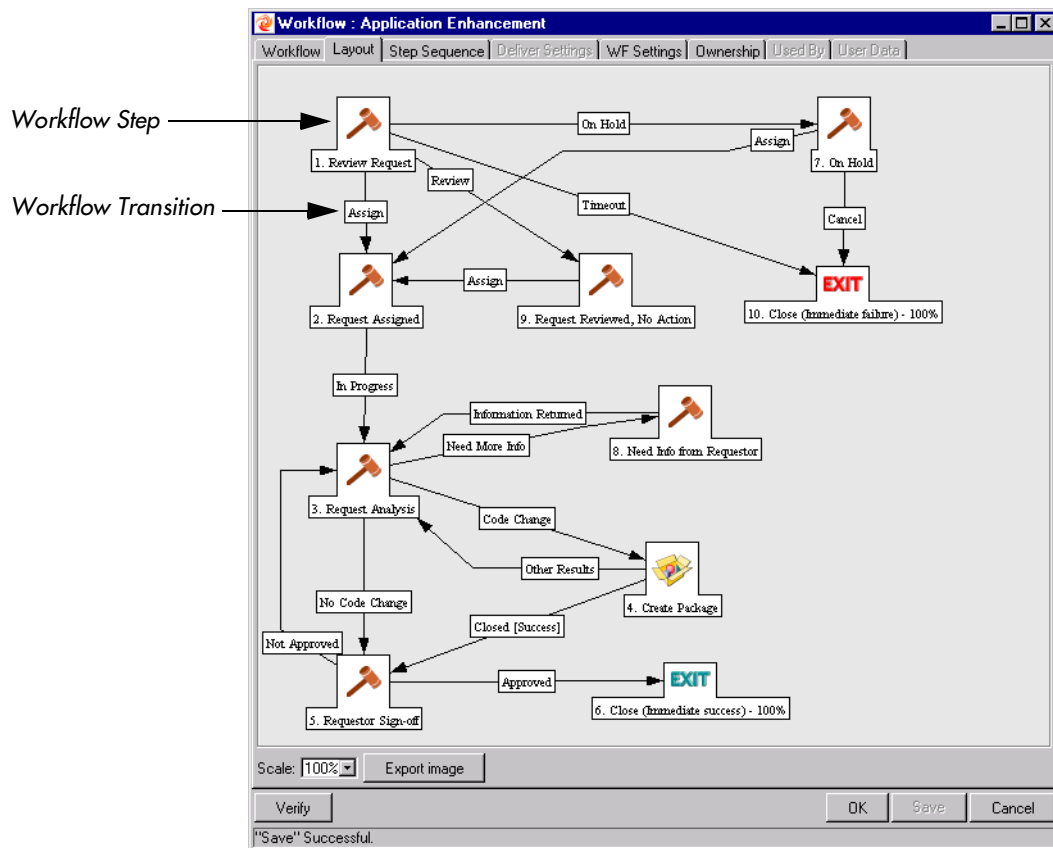


Figure 2-1 Sample Workflow

## Request/Workflow Interaction

Each Request Type has a list of possible Statuses it can take on (Assigned, On Hold, New, etc.). Each Status can be linked to a particular Workflow Step and drive field-level behavior.



A Request reaches a Workflow Step that links to a Request Status of “Assigned.” At this point, a field in the Request, ASSIGNED USER, becomes required. This field must be filled in before the Request can continue processing.

Since Request Statuses can be linked to field behavior through Status Dependencies (see “*Field Logic*” on page 12 for more information), field properties can also be altered as the Request is processed along a Workflow.

Request Type commands can also be triggered at a particular Workflow Step.

## Request Type Commands

Commands are instructions interpreted by the Kintana Execution Engine and translated into operating system commands to be dynamically executed. Commands are typically a blend between shell scripts and Kintana system Special Commands. Commands in Kintana allow the automation of an entire sequence of commands that would previously have been run manually. For example, these command sequences can automate source code compilation, check files into version control, or run a report.

Request Type commands define the execution layer within Kintana Create. 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.

## Special Commands

In order to simplify programming commands, Kintana provides a predefined set of Special Commands. These commands perform a variety of common functions, such as copying files between environments and establishing connections to environments for remote command execution. Kintana features two types of Special Commands:

- **SYSTEM SPECIAL COMMANDS** - These commands are shipped with Kintana. System Special Commands are read-only and have the naming convention “KSC\_COMMAND\_NAME.” System Special Commands always begin with “KSC\_.”

- **USER DEFINED SPECIAL COMMANDS** - These commands are user-defined and have the naming convention “SC\_COMMAND\_NAME.” User-defined Special Commands must begin with “sc\_.” User defined Special Commands can contain one or more of Kintana’s system Special Commands.

For more detailed information on Special Commands, see ["Using Commands and Tokens"](#).

## Validation

Validations determine the acceptable input values for user-defined custom fields. Validations maintain data integrity by ensuring that the correct information is entered in a field before it is saved to the database.



Validations can be used to ensure that no textual information is entered into a numeric field, or that dates are entered into a field in the proper format.

More complex Validations can be used to verify that only appropriate Kintana users are assigned to a Request. The values in selection Validations (drop down lists and auto-complete lists) can be configured by either listing the values or performing a SQL query.

Validations are used throughout Kintana:

- Every custom field generated for an Object Type, Report Type, Request Type, or User Data has a Validation.
- Every decision and execution Workflow Step has a Validation.

Every drop down list and auto-complete list in all Kintana windows are based upon a Validation. However, it is not always possible to change the Validations associated with predefined fields.

## Token

While configuring certain features in Kintana, it is often necessary to reference information in variables that is undefined until the Kintana product is actually used in a particular context. Instead of generating objects that are valid only in those specific contexts, these variables can be used to facilitate the creation of

general objects that can be applied to a variety of contexts. These variables are called **Tokens**.

There are two types of tokens used within Kintana: custom tokens and standard tokens. Standard tokens are provided with the product. Custom tokens are generated to suit specific needs. Each field of the following Kintana entities can be referenced as a token:

- Object Types
- Request Types
- Report Types
- Project Templates
- User Data
- Workflow Parameters

Tokens can be used in many Kintana entity windows:

- Object Type commands
- Request Type commands
- Validation commands and SQL statements
- Report Type commands
- Executions and notifications for a Workflow
- Workflow Step commands
- Notifications in a Report Submission
- Special Command commands
- Notifications for Tasks
- Field security
- Notifications for Request Types

## Security Groups

Security Groups are constructed to provide a set of users with specific access to screens and functions within Kintana. 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 user can view or edit, which Workflows a user can use, and which Workflow Steps a user has authority to act on. Each Kintana user can be a member of any number of Security Groups. The collection of Security Groups to which a user belongs defines that user's role within Kintana. Since users can be members of as many Security Groups as necessary, it is recommended that specific Security Groups are generated, each with a smaller range of responsibilities. Users can then be added to many different Security Groups to grant them their full range of access.

Security Groups control product access on the following levels:

- **Screen Security:**  
Each Security Group contains a list of Access Grants that determine a user's screen security. Access Grants are used to grant access to edit, view, manage or submit within a specific Security Group. By controlling the set of Access Grants for each user, specific functional roles for the user community can be defined.
- **Workflow Step Security:**  
To enforce the structure of your organization, security is controlled at the Workflow Step level. Each Workflow Step can be linked to a unique set of Security Groups. By adding or removing specific Security Groups from a Workflow Step in the Workflow window, each individual Kintana user can be limited in the set of Workflow Steps on which they have permission to act. This security level provides an extremely detailed level of control over each Kintana user's actions.
- **Request Field Security:**  
To enforce the structure of your organization, security can be controlled at the Request field level. Each field in a Request can be linked to a unique set of Security Groups. By adding or removing specific Security Groups from a field in a Request, each individual Kintana user can be limited in the set of Request fields that they have permission to see or alter.

See "[Kintana Security Model](#)" for details on configuring security and user access around your deployment and distribution system.

## Participants

Users who are involved in moving a Request through a Workflow are considered to be **Participants** in that Request. A Participant can be the:

- ASSIGNED TO user
- A member of the ASSIGNED GROUP
- The creator of the Request
- A member of a Security Group associated with any of the Workflow Steps contained in the Workflow.

You can configure Kintana so that a Request is not visible to users who are not Participants. This means users will only see Requests relevant to their business role in their organization. Additionally, users running Reports will only see information for Requests for which they are considered to be Participants.

## Integration with Kintana Products and Solutions

This document focuses on configuring the Request resolution functions within Kintana Create. With additional Kintana products and licenses, you can address the full set of challenges across your IT department. All Kintana solutions and products are designed to be seamlessly integrated to provide a complete solution to your IT management needs.

Other Kintana products include:

- [\*Kintana Solutions\*](#)
- [\*Kintana Dashboard\*](#)
- [\*Kintana Drive\*](#)
- [\*Kintana Deliver\*](#)
- [\*Kintana Accelerators\*](#)

## Kintana Solutions

Kintana's Enterprise Application for IT includes a set of proven solutions to support key IT processes and functions. Each Solution introduces specialized

business content developed to address specific business needs. Based on proven business practices, these solutions can be implemented modularly. Kintana supports the following solutions:

- **Demand Management:** ensures that all demands, regardless of type or source, are captured, evaluated, prioritized, and resolved efficiently.
- **Portfolio Management:** ensures alignment of strategic IT initiatives with business strategy.
- **Program Management Office:** ensures IT projects are delivered with very high quality and functionality, on time, within budget.
- **Enterprise Change Management:** ensures that IT delivers software for business use efficiently, with the highest quality and functionality, on time, and at low risk to production systems.

Kintana Solutions are licensed separately. For more information on implementing Kintana Solutions at your site, refer to Kintana's web site (<http://www.kintana.com>).

## Kintana Dashboard

Intended for large and complex environments, Kintana Dashboard™ provides 360° visibility and control over technology-based initiatives and IT operational tasks. Configurable, role-based visual displays called “portlets” provide relevant summary information and highlight exception conditions in your Kintana-managed initiatives. Users can then drill down to any desired level of detail.

For example, a CIO may want to see the status of the major initiatives undertaken by the IT department. Instead of relying on weekly reports patched together from different sources and often compiled from out-of-date or incomplete information, he can go directly to Kintana Dashboard. The Dashboard displays the true status of the initiatives -- based on current data captured automatically as part of actually performing the work. Kintana Dashboard clearly identifies any initiative that is behind schedule, or in any other exception state, and displays the causes for the delay.

Kintana Dashboard is beneficial to all participants throughout the Technology Chain. For example, developers can use Kintana Dashboard to view all of their own action items, and end-users can consult their own Dashboards to see the status of all the Requests they have submitted.

## Kintana Drive

Kintana Drive adds a critical dimension, automated execution, to complex project management in large IT organizations. Unlike static project management tools that simply schedule the tasks, dates, and resources, Drive's automation proactively pushes project tasks to the assigned resource, links with Kintana Create and Kintana Deliver to automatically perform issue resolution and deployment tasks, and automatically updates and reports project status as tasks are completed. Project managers guide projects from concept to completion from Kintana Drive's centralized environment.

Requests (used in your Request resolution process) can be added to the Kintana Drive project plan. Dependencies can be set between Requests and Tasks on the project. This ensures that the technical aspects of the deployment process is respected by other resources on the project plan.

## Kintana Deliver

Kintana Deliver lets you automate and manage migration and deployment of application changes for packaged applications, custom applications, legacy systems, Web content, and more. Leveraging and enforcing best practice deployment processes, Kintana Deliver performs all the tasks required to install software changes correctly across your development, test, staging, and production system landscape. Developers can concentrate on developing and adapting enterprise applications rather than non-value-added tasks such as code migrations. Packages group software changes so you deploy them completely, eliminating errors. In the event of a failure, Kintana Deliver's complete audit trail helps developers pinpoint the cause of the problem and rollback changes if necessary.

## Kintana Accelerators

Kintana Accelerators simplify the complex activities required to maintain large enterprise applications like Oracle, PeopleSoft, SAP, Siebel and Web applications built using Java, Oracle and others. These applications are constantly changing as new modules are added, customizations developed, configurations modified, patches applied, etc. The changes must be done precisely across the Development, Test, Stage and Production system landscape, usually by highly paid and hard-to-find specialists. Kintana Accelerators automate these precise tasks using best practice processes designed specifically for each application.



## Reports

Kintana features two types of reports: standard reports and Decision Support System (DSS) reports. Kintana's standard reports output text that provides information on your specific entities or configurations. Kintana's DSS reports feature a graphical data display which helps evaluate key system and process performance.

See "[Kintana Reports](#)" for a complete list of the reports used in commonly used in Request resolution systems.



# Chapter 3

## Using Migrators to Develop your Kintana Configurations

Before rolling-out new or modified functionality in Kintana, you should thoroughly test the changes in a Development or Testing instance. For example, before rolling out a new SUPPORT ENHANCEMENT CONTROL process to manage enhancements to your company's internal Support software, you should test the Kintana Workflow and Request Types used to resolve Requests.

Kintana provides functionality to help with this process: the Kintana Migrators. The Kintana Migrators are used to capture and move Kintana configuration data (for example, Workflow or Request Type definitions). This allows you to share configuration data between multiple Kintana instances. You can configure and test your Kintana configurations in a TEST instance, and then migrate your configurations to the PRODUCTION instance.

This chapter provides an overview of how to use multiple instances of Kintana to configure and deploy your Kintana configurations. It explains the concepts and basic architecture of this model, but does not provide implementation details. See the following documents for additional details:

- ["Kintana System Administration Guide"](#) for instructions on setting up multiple Kintana instances.
- ["Kintana Migrators"](#) for detailed instructions on using Migrators to move Kintana configuration data.
- ["Kintana Installation Guide"](#) for instructions on installing new Kintana instances.



Note

You must have a Kintana Deliver license to use the Kintana Migrators.

This chapter represents a change management implementation recommendation. Using the concepts and procedures listed in this chapter can reduce the risk of down time when rolling-out your Kintana processes.

See "[Kintana Migrators](#)" for detailed instructions on using Kintana Migrators.

This chapter discusses the following topics:

- [Using Multiple Kintana Instances - Overview](#)
- [Migrating your Kintana Configurations](#)
- [Archiving your Kintana Configurations](#)

## Using Multiple Kintana Instances - Overview

Kintana recommends that you use multiple instances when configuring the entities and processes in the Kintana product suite. In the following sections, we will discuss the simplest multi-instance configuration, consisting of two instances: DEV (development) and PROD (production) located on different machines. You can extend the basic migration principles to support the number of Kintana instances used at your site.

There are two implementation scenarios for employing multiple Kintana instances. The process for implementing multiple Kintana instances differs depending on the following scenarios:

- **Single PRODUCTION instance is currently in use.**  
Requires you to clone the PRODUCTION instance (file system and database) to create the DEV instance.
- **New Kintana implementation.**  
Requires that you run the Kintana install multiple times.

## Single PRODUCTION instance is currently in use.

For this scenario, you need to clone the PRODUCTION instance. Each Kintana instance consists of a file system and an Oracle database. These can exist on Unix or Windows machines. Contact your Kintana System Administrator for details about your site's configuration.

To move from a single live Kintana instance to multiple instances:

1. Clone the PROD instance. This includes the file system, database, and license information. Details for this procedure are included in the Kintana System Administration Guide. You should work with your Kintana System Administrator to implement this configuration.
2. Configure any changes to Kintana in the DEV instance. This includes creating or modifying Workflows, Object Types, Validations, Security Groups, Environments, etc.
3. Configure a Package Workflow to migrate the Kintana configuration data from DEV to PROD. Kintana recommends that this process is configured in the PROD instance.
4. Migrate data from the DEV instance into the PROD instance. Again, this activity is performed from the PROD instance. Therefore, it may help you to think of migrating the data as an "import" process.

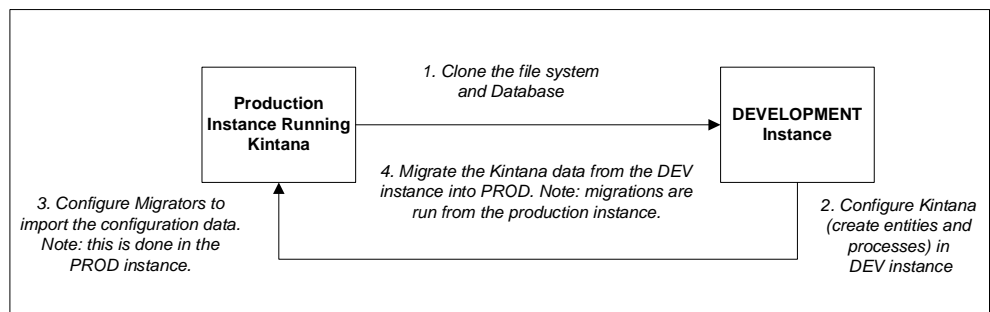


Figure 3-1 Cloning instance and configuring Kintana Migrators

## New Kintana Implementation

When first implementing Kintana at your site, you can choose to immediately set up multiple Kintana instances. One can be configured as the DEV instance,

and the other can be configured as the PROD instance. By creating two blank instances up front, you avoid the need to clone existing data from one instance into another. When this scenario is exercised, you can follow the instructions included in the "[Kintana Migrators](#)" document.

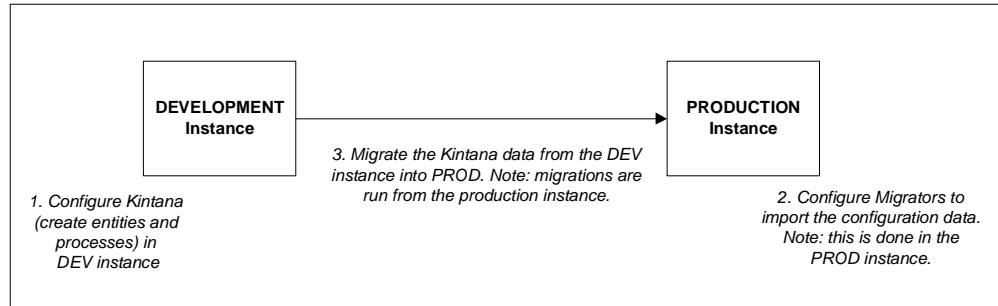


Figure 3-2 Migrating Kintana data between DEV and PROD

## Migrating your Kintana Configurations

This section provides an overview of the requirements and processes for using Migrators. These are provided to help you communicate with your Kintana Administrator (who maintains the Kintana instances and license information) and your Kintana System Administrator (who maintains the Kintana server) when developing your deployment and distribution processes in Kintana.

The following topics are discussed:

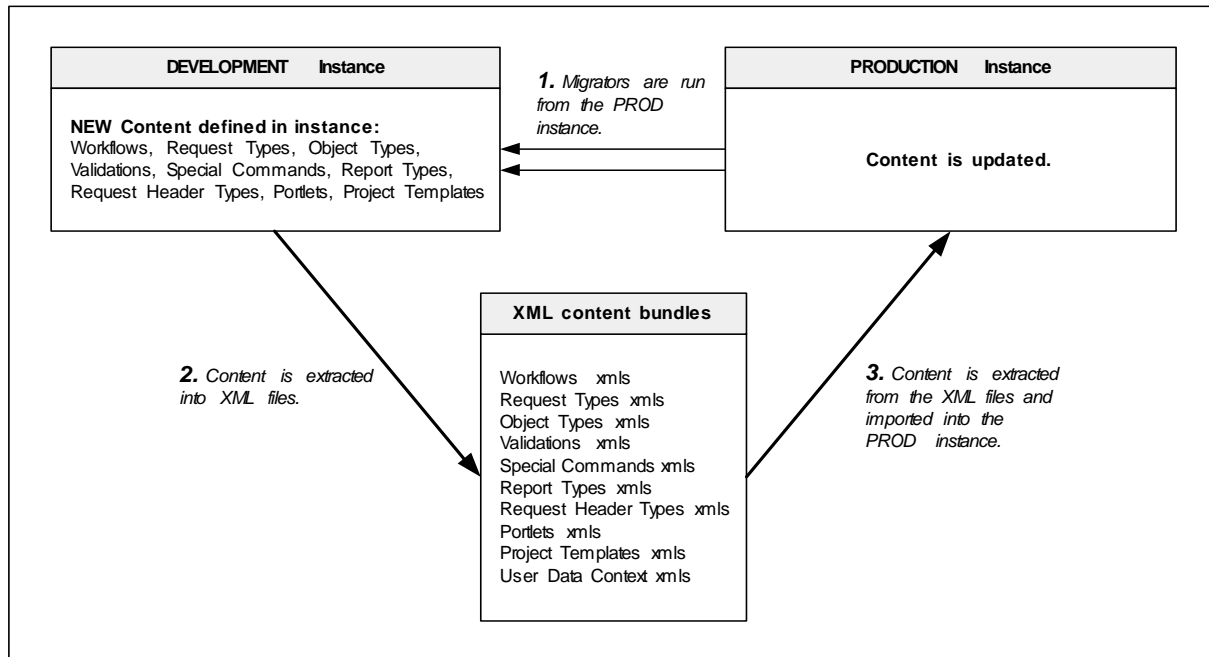
- [How Kintana Migrators Work](#)
- [Using the Kintana Migrators - Overview](#)
- [Instance Requirements for Using Kintana Migrators](#)

### How Kintana Migrators Work

Kintana Migrators are provided as Kintana Deliver Object Types. These Migrator Object Types are run through a Kintana Workflow. Each supported entity type has its own Object Type. For example, to migrate a Workflow from one instance to another, use the KINTANA WORKFLOW MIGRATOR Object Type.

Kintana Packages are used to process and audit the migration of configuration changes. When the Package (containing a Migrator object) enters the appropriate execution step in a Workflow, the Migrator's commands are executed. The commands extract the data that defines the entity into text (XML) files. These text files are then imported into the target instance.

*Figure 3-3* represents this process.



*Figure 3-3 Migrator content extraction and import overview*

## Using the Kintana Migrators - Overview

To use the Kintana Migrators to migrate Kintana configuration information between instances:

1. Configure the appropriate Environments (DEV and PROD) to represent Kintana instances involved in the migration.
2. Define a Kintana Package Workflow to model the migration process you want.
3. Build a Package using this Workflow, using the appropriate Kintana Migrator Object Types to create the Package Lines.

4. Submit the Package and migrate the Package Lines. Use the execution log generated to evaluate the results of the migration.

Detailed instructions for using the Kintana Migrators are included in the "[Kintana Migrators](#)" document.

## Instance Requirements for Using Kintana Migrators

To use the Migrators:

- You must have two working Kintana instances (DEV and PROD)
- The instances must be accessible over a network

### *Kintana Requirements for PROD Instance*

The following items must be configured in the PROD instance. This is the destination instance that will receive the configuration data from the DEV instance. See the "[Kintana Migrators](#)" for details on each of the below points.

Requirements for a successful migration:

- Environments (PROD and DEV) defined in the Environment screen in the Kintana Workbench.
- You must have at least one Kintana Workflow (SCOPE = **PACKAGES**) to run the migration. This Workflow must contain at least one execution step with the source and destination Environments configured to DEV and PROD, respectively.
- Migrator Object Types must be enabled. There is a different Object Type for each of the following entities that can be migrated: Validation, User Data, Special Command, Workflow, Report Type, Object Type, Request Type, Request Header Type, Project Template, Portlet, User Data Contexts.
- The user creating, submitting, and processing the migrations must have a Deliver power license and proper screen access. See "[Kintana Security Model](#)" for details.



## Archiving your Kintana Configurations

Kintana Migrators also let you document your processes by exporting configuration information to text files. These text files conform to the XML (eXtended Markup Language) specification and are suitable for storage in many archiving systems including source control systems. Source control check-in can be integrated into the Kintana Migrator Object Types, allowing organizations to maintain a detailed record of the specific changes made to their production Kintana configurations.

To archive your Kintana configurations:

1. Perform an Extract only using the appropriate Kintana migrator. The content is extracted into .xml files and grouped (by entity) into .zip files.
2. Check the extracts (.zip files) into your source control. Each extract contains a file (Source\_Descriptor.xml) that describes the Kintana version and date of extraction.
3. You can then import these files back into a Kintana instance (of the same Kintana version) at any point in the future.



# Chapter 4

## Configuring your Request Resolution System - Process Overview

This chapter introduces the process used to configure a Request resolution system in Kintana. It summarizes each phase of configuration. Additional details for configuration, including worksheets and checklists, are included in the following chapters.

Note

Kintana recommends that you develop your configurations in a development instance. See *“Using Migrators to Develop your Kintana Configurations”* on page 23 for an overview of using multiple Kintana instances for developing your processes in Kintana.

When configuring a Request resolution system or process in Kintana, you will follow the process described below.

1. *Gathering Process Requirements and Specifications*

Before you begin configuring the Kintana product to manage your Request resolution process, you need to collect specific related information. This includes information on your business process, any information needed to process the Request, any automated executions that need to occur during the process, and the communication devices surrounding the process.

2. *Mapping your Process into a Kintana Workflow*

Using the information gathered in the *Gathering Process Requirements and Specifications* chapter, you will build your Workflow in Kintana. This section includes instructions on setting up required Workflow step sources, creating Validations to be used by the transitions, and adding steps and transitions to your Workflow.

3. *Constructing the Request Type*

Using the information gathered in the *Gathering Process Requirements and Specifications* chapter, build your Request Type(s) in Kintana. This includes creating and configuring Request Type fields and field logic.

The final step in constructing the Request Type is to link Request Statuses to Workflow Steps.

#### 4. *Integrating Participants into Your Request Resolution System*

After the Workflow is constructed, construct security around the process. Ultimately, you need to specify who can do what within your process. This includes: who can create and process Requests, who can act on a particular Workflow step, and who can alter the process (Workflow, Request Types, etc.).

#### 5. *Setting Up Communication Paths*

Kintana also includes a number of features that enable high visibility into Requests in your Request resolution process. This includes instructions for creating notifications for Workflow steps, configuring portlets to provide additional real-time visibility, and configuring and running reports.

## Example: Configuring a Request Resolution System

This section provides a business case example for configuring a Request resolution system in Kintana. This example is used throughout this document to discuss Kintana configuration techniques.

### **Example: Request Resolution System for ACME Company**

The financial group at ACME Company uses a proprietary software solution to manage the many aspects of ACME's finances:

- Billing
- Accounts Payable
- Accounts Receivable
- Fixed Asset Management
- Inventory
- Payroll

- Reporting
- Cash Management
- etc.

This software solution is still relatively new and is used by hundreds of employees every day. Almost every day, someone thinks of an idea for new or enhanced functionality to the financial system that they put to their manager. Depending on the quality of the suggestion, a manager has the option of submitting a request for new or enhanced functionality to the financial system. Initially, the request must contain the following information:

- Whether the request is for new or enhanced functionality
- The module to be enhanced (Billing, Inventory, Payroll, etc.)
- The priority of the request
- Who originated the request
- A description of the new or enhanced functionality
- Any supporting documents (detailed proposals, Web sites, etc.)

If the request is approved, more information must be gathered:

- The user who approved the request
- Estimated time to completion
- The group assigned to build the new or enhanced functionality
- Any supporting documents (design documents, test plans, etc.)

During its lifecycle, the request must be approved or at least viewed by the following users or groups:

- The manager who originated the request
- The financial group director (budget approval)
- An IT analyst (feasibility, time estimates)
- The IT manager (approval to build)
- Developers (building the new/enhanced functionality)



# Chapter 5

## Gathering Process Requirements and Specifications

Before you begin configuring the Kintana product to manage your Request resolution process, you need to collect specific related information. This includes information on the:

- **Business process:**  
What are the steps in the process and which steps need to be reviewed and approved?
- **Information to gather:**  
What information needs to be gathered to make
- **Participants who will create and process Request:**  
What level of security do you want to place on this system?
- **Communication devices surrounding the process:**  
Do you want to communicate using notifications, the Kintana Dashboard, or reports.

You need to consider all of the above topics when configuring a new process in Kintana. This chapter discusses the information that you will need and provides examples to help you manage this information. This chapter includes the following sections:

- *Identify Needed Entities*
- *Gather Requirements for Workflow*
- *Gather Requirements for Request Type*
- *Identify Participants and Security*
- *Establish Communication Points and Visibility*

## Identify Needed Entities

The first step to configuring your Request resolution process is to determine requirements for the Kintana entities that will be needed. The following entities are necessary for a Request resolution system:

- [Workflows](#)
- [Request Type](#)
- [Request Header Type](#)
- [Security Groups](#)

## Workflows

The Workflow is the main process element in a Request resolution system. It provides the framework along which a Request moves towards resolution. Elements needed for a Workflow include:

- Process steps — The steps in the Request resolution process.
- Transitions between steps — Transitions connect the process steps and define where a Request goes. This translates into the Workflow Step's Validation.
- Request Type Statuses — These are defined by the [Request Type](#). Request Type Statuses link to individual process steps and can change Request field attributes.

For more detailed information on Workflow requirements, see [“Gather Requirements for Workflow”](#) on page 37.

## Request Type

A Request Type defines the structure of a Request in Kintana Create. The fields that are used when a Request is created are customized based on the Request Type. Necessary elements for a Request Type include:

- Fields — Request Type fields capture information necessary for Request resolution.
- Field behavior — Request Type fields can be configured with default values, auto-population rules, and attribute changes.
- Statuses



For more detailed information on Request Type requirements, see [“Gather Requirements for Request Type”](#) on page 42.

## Request Header Type

A Request Header Type defines a standard set of fields to be associated with a Request Type. Request Header Types can contain Field Groups, which are collections of fields pre-configured for Kintana functionality like Resource Management, or the Program Management Office Solution. For more detailed information on Request Header Type requirements, see [“Request Header Type”](#) on page 44.

## Security Groups

Security Groups are sets of users with specific access to screens and functions within Kintana. A user's Security Group memberships determine which windows user can view or edit, which Workflows a user can use, and which Workflow Steps a user has authority to act on. Security Groups should be considered for the following Request resolution entities:

- Workflows
- Request Types
- Request Type fields

For more detailed information on Security Group requirements, see [“Identify Participants and Security”](#) on page 45.

## Gather Requirements for Workflow

The first step to configuring your 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 Steps, when (if at all) to execute commands, and what's the path (transitions) between steps in the process.

The following sections discuss the specific information that you need to gather to help you later with your Kintana Workflow configuration:

- [Defining the Business Flow](#)
- [Gather Information on Each Step in the Process](#)

- *Consider Using Subworkflows*

## Defining the Business Flow

Map your business process. This consists of all of the steps (decisions, conditions, points of execution if any) and transitions needed to resolve your Requests. It is helpful to graphically map these processes. The following example provides an illustration of the design issues that you should consider.

### *Example: Defining the Business Flow*

ACME Company needs to configure a resolution process for processing change requests for their Financial Applications system. This system consists of over ten modules (billing, accounts payable, accounts receivable, fixed asset management, inventory, reporting, payroll, cash management, etc.). ACME's IT group needs to create a process that can address the complications related to evaluating and approving changes to this system, with an eye toward deploying the changes.

#### **Business Process Overview**

ACME 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 ACME's Financial group. If the priority of the Request is **HIGH** or **CRITICAL**, a manager will need to be informed right away.
2. **Validate:**  
The Request is reviewed. 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.
4. **Schedule:**  
If the Request is approved, the work needed to create the enhancement must be scheduled. If ACME lacks the necessary resources at the moment, the Request should be put on hold.
5. **Develop:**  
The requested enhancement is developed by ACME's IT group.

**6. Deploy:**

The finished enhancement is deployed to ACME’s Financial group.

After investigating their Request resolution process, ACME identified a process change that needed to be made:

- Requirement:*  
 When a Request is submitted, its PRIORITY should be evaluated. If the Request is **CRITICAL** or **HIGH**, the Financial manager and members of the ACME’s IT group should be informed.
- Result:*  
 To address this requirement, ACME added an Execution step that would evaluate the Request’s PRIORITY. If the Request is **CRITICAL** or **HIGH**, it is routed to another Execution step that sends a Notification to the appropriate users.

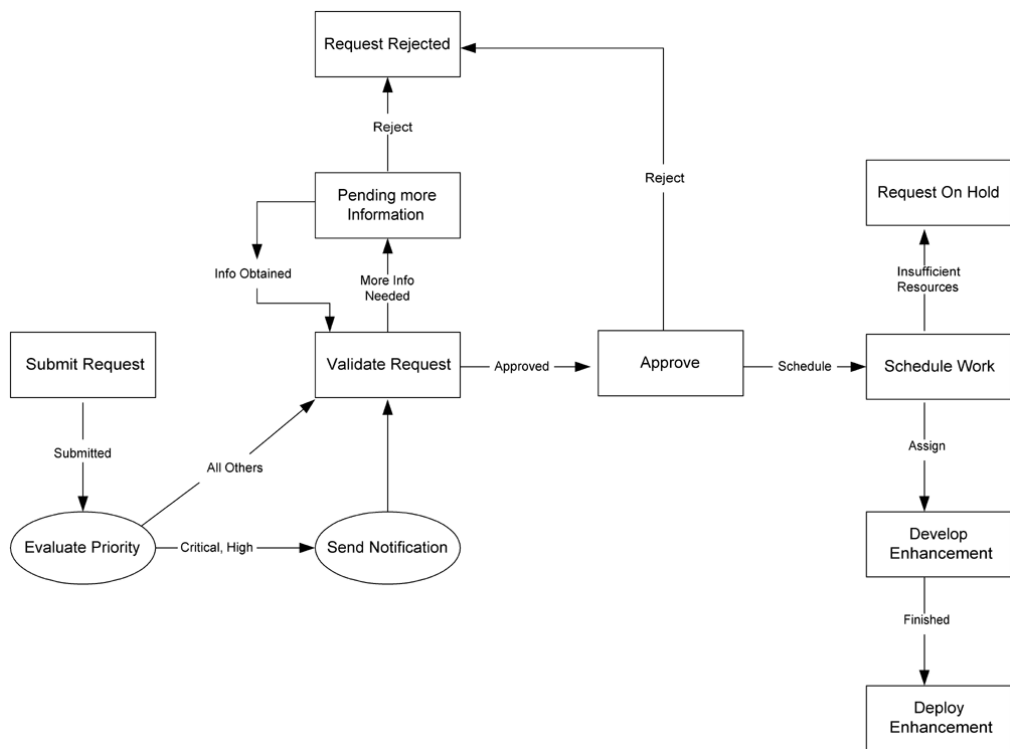


Figure 5-1 Revised Business Process

## Gather Information on Each Step in the Process

After designing the business flow of your request resolution process, you need to gather detailed information on each step and transition in your process. This

section discusses the information that you should collect. “*Configuration Worksheets*” on page 347 includes a worksheet that will help you collect the required information.

Kintana recommends a phased approach to collecting this information.

1. Model your business process graphically, possibly creating a diagram similar to *Figure 5-1*.
2. Capture step names, descriptions, transition values and step goals.
3. Fill out details for each step. The worksheets in “*Configuration Worksheets*” on page 347 can be a helpful guide.

### Consider Using Subworkflows

A Subworkflow is any Workflow that is referenced from within another Workflow. Subworkflows allow you to model complex business processes into logical, more manageable and reusable sub-processes.

Workflows can be used as Subworkflows within a parent Workflow. An entire Subworkflow is represented by a single icon 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 a Subworkflow*

ACME decides to use a Subworkflow for the development portion of their process. This Subworkflow can be referenced in one part of the process. *Figure 5-2* illustrates where ACME could implement a Subworkflow.

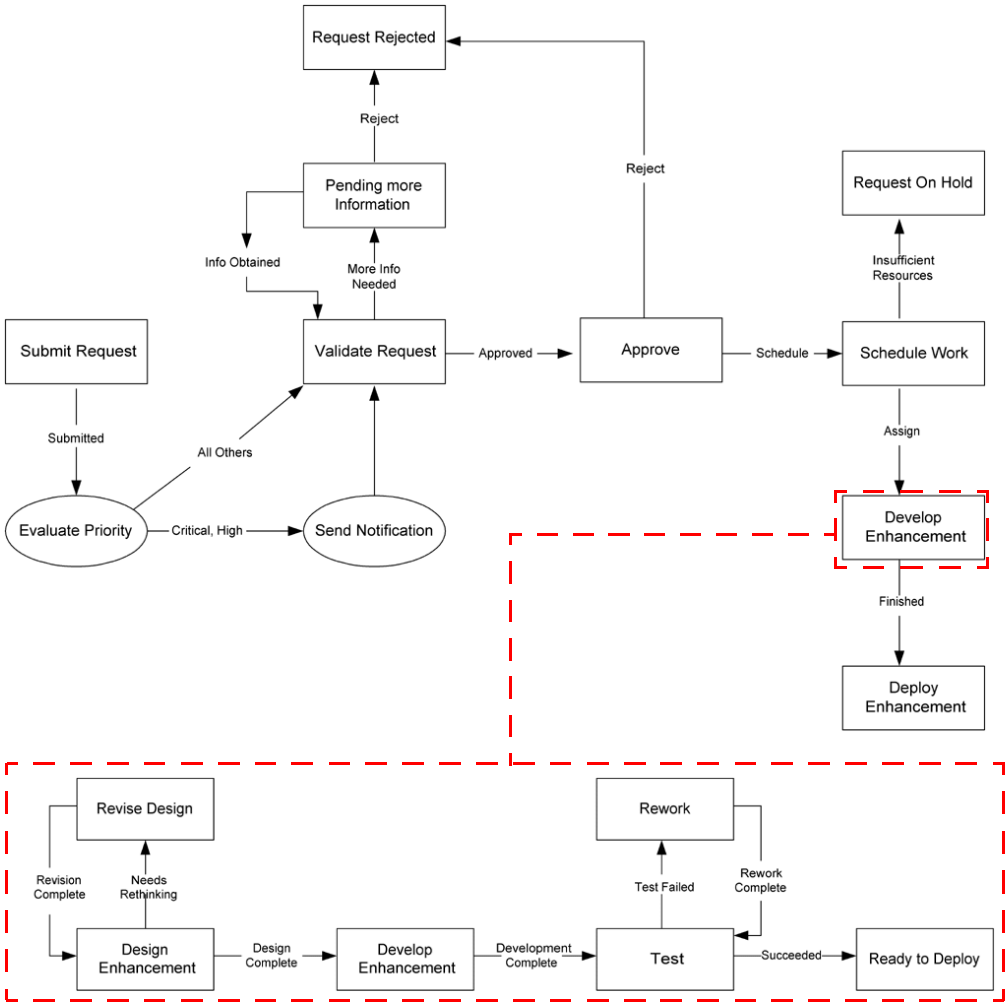


Figure 5-2 ACME Business Process with Subworkflow

### Consider 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. You will have to build the Request Type before you can associate its Statuses with Workflow Steps. See *“Creating Your Request Statuses”* on page 147 for more detailed information on Request Statuses.

## Gather Requirements for Request Type

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.

The following sections discuss the specific information that you need to gather to help you later with your Kintana Request Type configuration:

- [Request Type Fields](#)
- [Request-Workflow Interaction](#)
- [Request Header Type](#)
- [Request Type Commands](#)

### Request Type Fields

Each Request requires different information to process it. For example, to resolve a software bug, you need to know the unit, version, problem and priority. The information is captured and manipulated using Request Type fields.

For each field in the Request you will be running through your process, 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 do you want to collect? Should it be a text field? Will users pick from a pre-determined 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. For more detailed information on Validations, see "[Request Type Field Validations](#)" on page 117.
- Field behavior. There are many aspects of a field that can be controlled:
  - o The field can be configured to become uneditable or required depending on the value of other fields, or at a certain Workflow step.
  - o The field can also be configured to automatically populate itself based on values in other fields.
  - o The field can also be configured to be uneditable or invisible based on which user is looking at the Request.

More detailed information on field behavior can be found in “[Configuring Field Behavior - Overview](#)” on page 122.

The worksheets in “[Configuration Worksheets](#)” on page 347 can be a good way to systematize the collection of Request Type information and field specifications.

### *Example: ACME collects information for the software change Request*

ACME 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 (Billing, Inventory, Payroll, etc.)
- The priority of the request
- A description of the new or enhanced functionality
- Any supporting documents (detailed proposals, Web sites, etc.)

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

- Whether ACME 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, ACME 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 “*Constructing the Request Type*” on page 107 for additional examples on Request Type fields.

## Request-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 Step, it acquires a Status that determines its fields’ attributes. For more detailed information on Request Status Dependencies, see “*Configuring Field Behavior Using Status Dependencies*” on page 146.

In most cases, a single Request Type is processed through 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 1:1 mapping.

It is also possible to restrict which Workflows can be used by a Request Type, and vice versa. You should determine what, if any, restrictions are to be put in place at this level.



Requesting a software change requires different information and processing than requesting a scope change to a project. Therefore, it is likely that the Workflows built around each business process will be different, and at least one field in each Request Type will be different.

## Request Header Type

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. When creating or configuring a new Request Type, you associate a Request Header Type with that Request Type. Though Request Header Types already contain a standard set of fields that can be manipulated, you can also create new Request Header Type fields similar to Request Type fields.

Request Header Types can also contain Field Groups, which are used to enable Kintana functionality such as Resource Management and Kintana Solutions.





Note

When Field Groups are associated with existing Request Types (through the Request Header Type definition), tables in the Kintana database are updated to handle this new configuration. Because of the scope of database changes, you should re-run the Database Statistics on your Kintana Database. Instructions for this are included in the Kintana System Administration Guide. Contact your System Administrator for help with this procedure.

## 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.

You should collect the following information for each Request Type command that you design:

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

Use the worksheet in *"Configuration Worksheets"* on page 347 for assistance in collecting the correct data.

See *"Using Commands and Tokens"* for additional information on building commands in Kintana.

## Identify Participants and Security

Kintana allows you to exercise a great deal of control over your Request resolution process. Users can be arranged into Security Groups, and each Security Group given a different area or level of access.

For your Request resolution process, you should collect information that will help you to identify users, group them into security groups, and restrict access to certain functions in Kintana.

See the worksheet in *"Configuration Worksheets"* on page 347 for details.

See also *"Kintana Security Model"* for a comprehensive discussion of Kintana screen, entity and user security.



Kintana recommends using Security Groups or dynamic access (Tokens) whenever possible. You should 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 an organizational reorganization), you would have to update that list in many places on the workflow. By using a Security Group instead of a list of users, you can update the Security Group once, and the changes are propagated throughout the Workflow steps.

Security Groups can be assigned to the following entities:

- [Security for Workflows](#)
- [Security for Request Types](#)
- [Security Around Request Fields](#)
- [Configuration Security](#)

## Security for Workflows

You can determine who can approve / process each step in a Workflow. For this restriction, you can enable access by specifying users or Security Groups for each Workflow Step. You can also provide access dynamically by having a Kintana Token resolve to provide access. See [“Mapping your Process into a Kintana Workflow”](#) on page 55 for more information.

You can also restrict the users who can use a particular Workflow to a certain list of Security Groups.

## Security for Request Types

You can restrict users’ actions around:

- Who can create Requests.
- Who can use specific Request Types.
- Whether you only want “Participants” to process the Request. Participants are defined as the Assigned User, the creator of the Request, members of the Assigned Group, or any users who have access to the Workflow step(s). This restriction is performed at the Request Type level.

## Security Around Request Fields

You can determine whether there are any fields in the Request that should be hidden from or made non-editable to certain users, and if so, who can and can't see or edit them.

## Configuration Security

You can set up security around your Kintana configurations themselves by specifying the Security Groups who can perform the following actions:

- Who can change the Workflow.
- Who can change each Request Type.

## Example: ACME Determines Participants and Security

The process of approving changes to ACME's Financial system application involves many groups and individuals within the company.

- ACME Financial Group: Users of ACME's Financial system application
- Lucy Barnstorm: Director of ACME Financial Group
- Ralph Guderjahn: Business analyst
- Tyrone Chambers: Director of ACME IT group
- Hiroki Nanahara: Personnel Manager in ACME IT group
- ACME Development: Engineers for ACME IT
- Carlos Quintana: Lead Engineer
- Nora van Epstein: Manager of Release team
- Harold Lomax: Configuration Manager for ACME IT

Within this group of users, there are some logical divisions of labor. Using this division, ACME constructs the following Security Groups.

Table 5-1. ACME's Security Groups

Security Group	Members	Responsibilities
Financial Apps - Create and View Requests	ACME Financial Group Lucy Barnstorm	Responsible for creating Requests. These people can create Requests at any time and view the status of Requests they are involved in.
Financial Apps - Manage Resolution System	Harold Lomax	Responsible for Request resolution system. This person has the ability to modify the Request resolution process (Workflow, Request Types, and Security Groups). He can also act on any step in the process.
Financial Apps - Validate and Approve Requests	Lucy Barnstorm Ralph Guderjahn Tyrone Chambers	Responsible for evaluating and approving incoming Requests. Can reject or approve Requests for development.
Financial Apps - Schedule Requests	Tyrone Chambers Hiroki Nanahara	Responsible for approving Requests for development, scheduling and assigning work, or putting Requests on hold until sufficient resources are available.
Financial Apps - Develop Requests	ACME Development Carlos Quintana Nora van Epstein	Responsible for developing enhancements specified in Requests, including functional design, implementation, and QA.
Financial Apps - Deploy Changes	Nora van Epstein	Responsible for overseeing deployments to the ACME Financial group.

Using these Security Groups and user definitions, ACME collects specific information related to their Request resolution process. This information will be considered later when defining your Security Groups and Workflows. The information is presented in the following tables:

- Table 5-2, “ACME Request Creation Security,” on page 49
- Table 5-3, “ACME Request Processing Security - Financial System Change Workflow,” on page 50
- Table 5-4, “ACME - Security around managing the Financial System Change Process,” on page 51

Table 5-2. ACME Request Creation Security

Action	Users allowed to perform action	Controlled by: (Users, Security Group, Token)
Create a Request	ACME Financial Group; Lucy Barnstorm	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 Harold Lomax 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.

ACME decides not to use Kintana's Participant Restriction functionality in their Request resolution process. See [“Restricting Request Processing to Participants”](#) on page 176 for additional details on this configuration option. The following table documents which users can act on a specific step in the Workflow. ACME 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 you can specify multiple criteria 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.

Only a sub-set of the workflow steps are included in the below table. See [Figure 5-3](#) to see the process referenced in this table.

Table 5-3. ACME Request Processing Security - Financial System Change Workflow

Workflow Step Name	Users allowed to act on	Controlled by: (Users, Security Group, Token)
Validate Request	Lucy Barnstorm; Ralph Guderjahn; Tyrone Chambers	Financial Apps - Validate and Approve Requests  Financial Apps - Manage Resolution System
Pending More Information	Financial Group member who created the Request; Lucy Barnstorm	TOKEN (REQ. CREATED_BY);  Financial Apps - Create and View Requests (Security Group)  Financial Apps - Manage Resolution System
Approve	Lucy Barnstorm; Ralph Guderjahn; Tyrone Chambers	Financial Apps - Validate and Approve Requests  Financial Apps - Manage Resolution System
Schedule Work	Tyrone Chambers Hiroki Nanahara	Financial Apps - Schedule Requests  Financial Apps - Manage Resolution System
Develop Enhancement	ACME Development Carlos Quintana Nora van Epstein	Financial Apps - Develop Requests  Financial Apps - Manage Resolution System
Deploy Enhancement	Nora van Epstein	Financial Apps - Deploy Changes  Financial Apps - Manage Resolution System

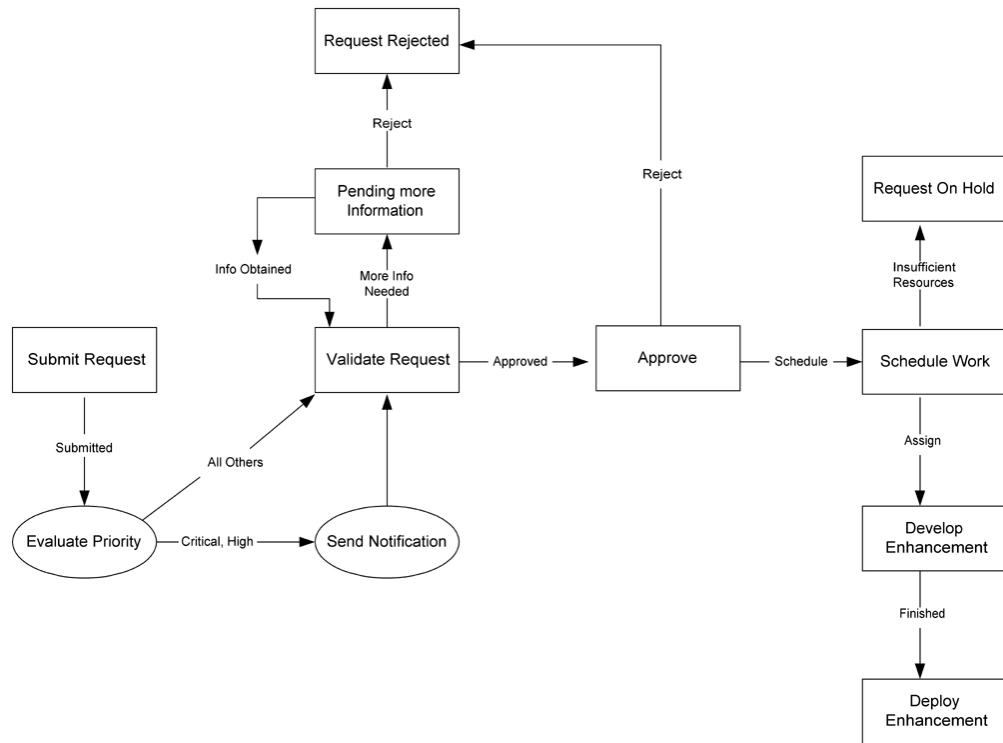


Figure 5-3 ACME Business Process

ACME must also specify who can modify the existing process. This level of security is configured using Kintana’s Ownership settings and Security Group access grants. See *“Setting Configuration Security”* on page 177 for more information on these topics.

Table 5-4. ACME - Security around managing the Financial System Change Process

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

## Establish Communication Points and Visibility

You must determine the communication points and methods for providing visibility into your process and Request statuses. Kintana provides the following helpful features to help you increase visibility:

- [Notifications on Workflow Steps](#)
- [Notifications on Field Changes](#)

This section lists the information that you should gather to define your Notifications. For more information on setting up Notifications, see [“Setting Up Communication Paths”](#) on page 183.

Portlets on the Kintana Dashboard and Kintana Reports can also be used to maintain visibility. For more information on defining and using Portlets and Reports, refer to the following documents:

- ["Configuring the Kintana Dashboard"](#)
- ["Kintana Reports"](#)

### Notifications on Workflow Steps

You can send a notification when a workflow step becomes eligible, has a specific outcome, or has a specific error. For each Workflow step in your process, collect the following information:

*Table 5-5. Information to gather for Workflow Step Notifications*

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 5-6. Information to gather for Workflow Step Notifications*

Parameter	Description
Workflow Step Name	The name of the step that requires a workflow.



Table 5-6. Information to gather for Workflow Step Notifications

Parameter	Description
Notification Event (All, Eligible, Specific Result, Specific Error)	Specifies the event that triggers the notification. the possible values are <b>ALL</b> , <b>ELIGIBLE</b> , <b>SPECIFIC RESULT</b> , or <b>SPECIFIC ERROR</b> .
Value (for Specific Result)	Specifies that a notification is sent for the selected result.
Error (for Specific Error)	Specifies that a notification is sent for the selected error.
Recipient	Determine who should receive the message. you can choose to send the notification to users based on: <b>USERNAME</b> , <b>EMAIL ADDRESS</b> , <b>SECURITY GROUP</b> , <b>STANDARD TOKEN</b> , or <b>USER DEFINED TOKEN</b> .
Message	Determine what the message will say. Also determine if it will contain a link to the Package.

See the worksheets in “[Configuration Worksheets](#)” on page 347 for assistance in collecting the correct data.

*Example: ACME configures notifications*

ACME determines that they would like to add a notification to the following steps:

Table 5-7. ACME - 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 also be sent when a field in a Request changes value. For more detailed information on setting up these Notifications, see [“Setting Notifications on Request Field Changes”](#) on page 202.

# Chapter 6

## Mapping your Process into a Kintana Workflow

This chapter provides an overview for how to set up a Kintana Workflow: all workflow steps, transitions and validations included in your process. It illustrates how the information gathered in *“Gathering Process Requirements and Specifications”* on page 35 and *“Configuration Worksheets”* on page 347 can be used to quickly build Workflow steps, transitions and validations.

This chapter discusses the following topics:

- *Building the Workflow Skeleton - Overview*
- *Create the Required Step Source*
- *Configure the Step’s Transition Values (Validation)*
- *Add Steps and Transitions to the Workflow Layout*

### Building the Workflow Skeleton - Overview

For each Workflow that you create, follow this general process:

1. Enter the general Workflow information in a new WORKFLOW window. Enter the NAME and WORKFLOW SCOPE in the **WORKFLOW** tab.
2. Create any new step sources using the WORKFLOW STEP SOURCES window. This includes:
  - a. Creating decision steps.
  - b. Creating execution steps.

- c. Creating subworkflow steps.
- d. Creating any new validations used by the above steps.
3. Add the Workflow steps to the **LAYOUT** tab.
4. Add transitions between the Workflow steps.
5. Add Security to the Workflow.
6. Add Notifications to select Workflow steps.
7. Synchronize Workflow steps with Request Type statuses.
8. Enable the Workflow.

### Required Workflow Settings for Request Resolution Process

- The **WORKFLOW SCOPE** must be set to **REQUESTS**
- The Workflow must be Enabled
- You must add steps and transitions to the **LAYOUT** of your Workflow
- If this Workflow is to integrate with a Package Workflow, you must specify the Package workflow in the **PACKAGE WORKFLOWS** tab
- You must enable Security for each Workflow step. This allows users to act on the step.



Note

- Workflows are created and configured using the Kintana Workbench.
- Users must have a Power License and have the proper Access Grants in order to create and edit a Workflow. See "[Kintana Security Model](#)" for details.

### Create the Required Step Source

Kintana provides a number of standard Workflow step sources that you can add to your Workflow. These sources are preconfigured with standard validations (transition values), workflow events, and workflow scope. These

available steps specify the following common attributes, which are expected to remain consistent across all Workflows which use that step source:

- 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 the WORKFLOW STEP SOURCES window to view the available steps at your site. When Kintana does not have a step source that meets your process requirements, you need to create one.



Tip

If Kintana has a Workflow step source that meets your process requirements, you can copy and rename it. This can save configuration effort and avoid user processing errors. For example, if you need a step to route a Request based on whether it needs more analysis, you could copy and use the REQUEST ANALYSIS Workflow step source that is delivered with Kintana.

Kintana recommends copying the step source so that you can use it uniquely for your processes. This allows you to control who can edit the step source, ensuring that your process isn't inadvertently altered by another Kintana user.

Create a new step source when the step requires any of the following:

- Unique Validation leaving the step
- Unique execution on the step: PL/SQL function, Token, SQL function, or Workflow Step Commands
- Different processing type: immediate vs. manual
- Specific Workflow Scope
- Unique combination of the above settings

The following sections discuss when and how to use specific settings in the Workflow Step Source:

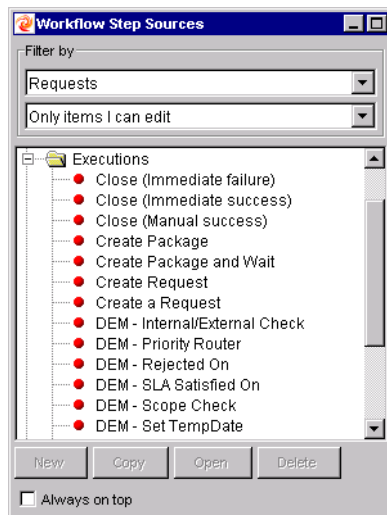
- [Creating a Workflow Step Source - Overview](#)
- [Creating a Decision Type Step](#)
- [Create an Execution Type Step](#)

## Creating a Workflow Step Source - Overview

You can create new Workflow step sources from the WORKFLOW STEP SOURCES window on the Kintana Workbench.

To create a new Workflow Step Source:

1. Click the **CONFIGURATION** screen group and click the **WORKFLOWS** icon. The WORKFLOW WORKBENCH and WORKFLOW STEP SOURCES windows open.
2. Select the WORKFLOW STEP SOURCES window.



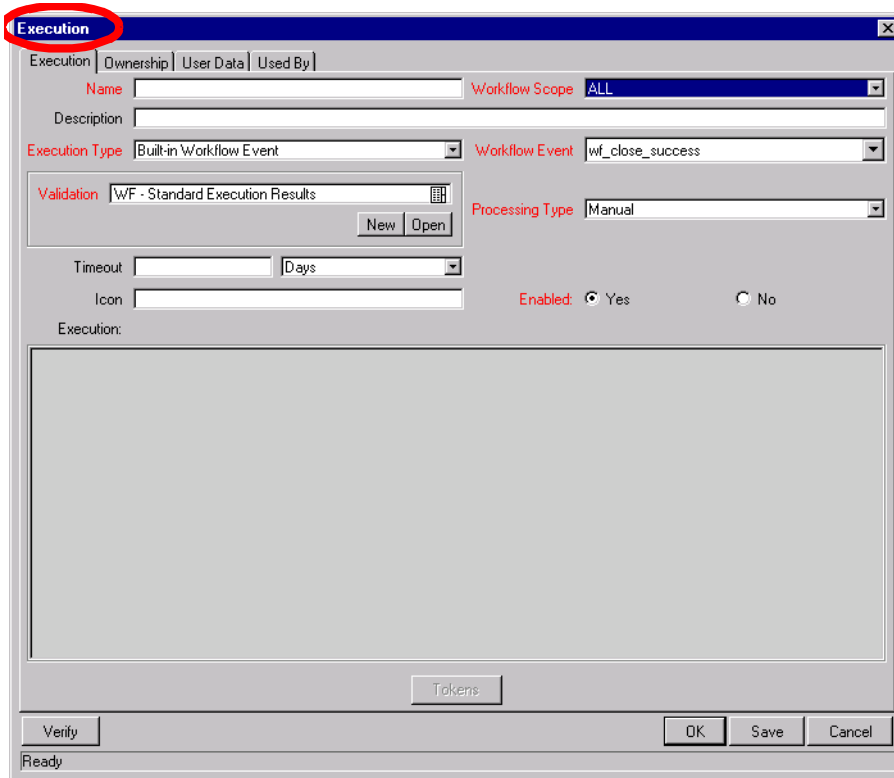
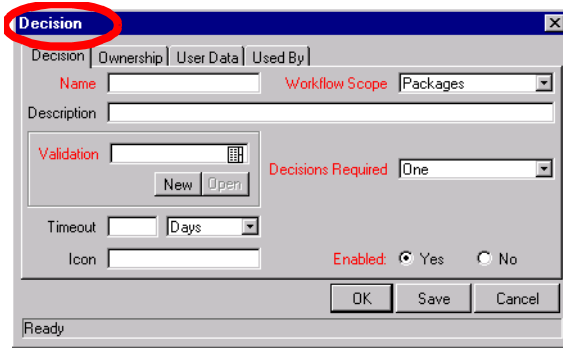
3. Select to FILTER BY **REQUESTS**.
4. Select the folder that corresponds to the type of Workflow step source that you would like to create. For example, to create an execution step, select the **EXECUTIONS** folder.
5. Click **NEW**. This opens a window that corresponds to the selected Workflow step source type. The **DECISION** and **EXECUTION** windows are shown below. For information on configuring Workflow Step Sources, see [“Creating a Decision Type Step”](#) on page 612 and [“Create an Execution Type Step”](#) on page 65.



Note

Condition steps cannot be added, deleted or modified. Kintana supports a set number of process Condition steps. These can be added to the Workflow layout just as any other Workflow step source.

If you select a Condition step in the WORKFLOW STEP SOURCES window, the **NEW** button will not be enabled.



6. Enter the required information and any optional information needed to define the step. See *“Creating a Decision Type Step”* on page 612 and

[“Create an Execution Type Step”](#) on page 65 for detailed information about setting up the steps.

7. Select **YES** in the **ENABLED** radio button to be able to use this step in a Workflow.
8. Click the **OWNERSHIP** tab. Select which Ownership Groups will have the ability to edit this Execution or Decision.
9. Click **OK** to save the changes and close the **EXECUTION** or **DECISION** window.

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**.

### Related Topics:

- [“Creating a Decision Type Step”](#) on page 61
- [“Create an Execution Type Step”](#) on page 65

## Workflow Step Source Configuration and Usage Restrictions

The following restrictions apply to Workflow step sources:

- You cannot delete a step source that is being used in a Workflow.
- You cannot change a validation for a step source that is being used. If you need to change the validation, copy the step source and configure a new validation.
- The Workflow step source must be **ENABLED** to use them on a Workflow.
- You can only add step sources to a Workflow when the Workflow has a matching **WORKFLOW SCOPE**, or the step source has a scope of **ALL**.
- You cannot delete a step from a Workflow that has been used to process a Request. This would compromise data integrity. Instead of deleting the step, remove all transitions to and from the step and disable it.



## Creating a Decision Type Step

To create a decision step source:

1. *Enter the general information on the Decision step source* (name, scope, description)
2. *Select a Validation*
3. *Specify the voting requirements on the step*
4. *Specify the default timeout value*

Table E-3. Workflow Step [Decision] -- Step Number \_\_\_\_.

	Value
Step Name	
Goal / Result of Step	
<b>Validation*</b>	
Decisions Required (Vote on Step's outcome?)	<ul style="list-style-type: none"> <li>• One</li> <li>• At Least One</li> <li>• All</li> </ul>
Timeout (Days)	
Security (who can act on step):	
<ul style="list-style-type: none"> <li>• Security Group</li> <li>• User Name</li> <li>• Standard Token</li> <li>• User Defined Token</li> </ul>	
Include Notification (Yes/No)	
Notification Event	
Notification Recipient:	
<ul style="list-style-type: none"> <li>• Username</li> <li>• Email Address</li> <li>• Security Group</li> <li>• Standard Token</li> <li>• User Defined Token</li> </ul>	
Notification Message	
Request Status at Step	
Request % Complete at Step	
Authentication Required (Y/N)	
Authentication Type (if Y)	

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

← Information used when adding the step source to the Workflow layout.

Figure 6-1 Information used to create the decision step source.

### Enter the general information on the Decision step source

Enter the following information in the DECISION window.

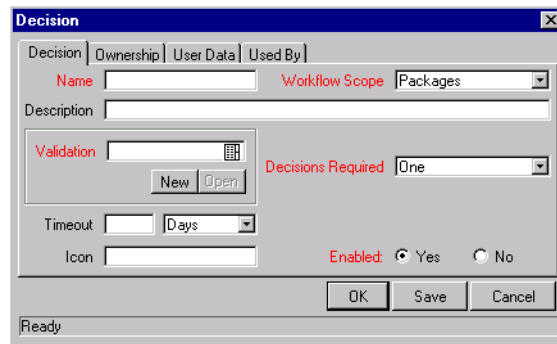


Table 6-1. Decision step source worksheet to window.

Field in Decision Window	Description
NAME	This is the name that describes the 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. This should be set to <b>ALL</b> or <b>REQUESTS</b> for Request resolution processes.
DESCRIPTION	Description of the step source.
VALIDATION	Specifies the possible values that can exit the Workflow step. See <i>“Configure the Step’s Transition Values (Validation)”</i> on page 81.
DECISIONS REQUIRED	This specifies the number of people who need to approve a specific step. See <i>“Specify the voting requirements on the step”</i> on page 63.
TIMEOUT	If this Workflow Step remains eligible for the value entered in the Timeout value, the Request can be configured to send an appropriate Notification, as well as escalate to other steps in the Workflow.
ICON	You can specify a different graphic to represent steps of this source for use on the Workflow layout tab.  This graphic needs to exist in the icons subdirectory of the directory specified by the server parameter BASE_PATH. If it is left blank, a default icon is used.
ENABLED	The step source must be enabled in order to add it to the Workflow layout.

## Select a Validation

Select a Validation that has the transition values required for leaving the step. If Kintana doesn't provide a Validation that meets your requirements, you can create a new one from the WORKFLOW STEP SOURCE window. See “[Validations](#)” on page 243 for a list of Kintana's seeded Validations.

See “[Configure the Step's Transition Values \(Validation\)](#)” on page 81 for additional details.

## Specify the voting requirements on the step

When a Decision step is defined, the number of decisions required for that Workflow Step can be defined. [Figure 6-2](#) displays the available options for the DECISIONS REQUIRED field.

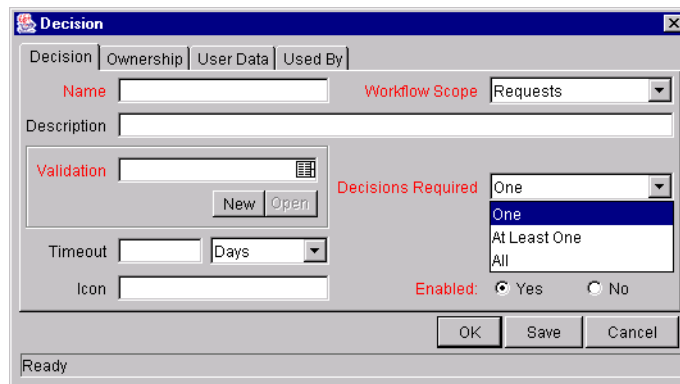


Figure 6-2 Decision Window - Decisions Required Drop Down List

The following choices are available for DECISIONS REQUIRED:

- **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**

A Timeout period must be defined to use this choice. When **AT LEAST ONE** is selected for the Workflow Step, the 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.



Note

You can define **SPECIFIC ERRORS** in Workflow Steps such as ‘**TIMEOUT**’ and ‘**NO CONSENSUS**’ as either Success or Failure in the **DEFINE TRANSITION** window. For more information, see “*Adding Transitions Between Steps*” on page 94.

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**

The **ALL** step is also commonly 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**.

### *Specify the default timeout value*

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** or **ONE**) or completing with a result (if **DECISIONS REQUIRED** is **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 can be configured to send an appropriate Notification and escalate to other steps in the Workflow. 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 step source acts as the default timeout value for the step. When you add a step to the Workflow using this step source, you can specify a different timeout value for the step.

## Create an Execution Type Step

To create an execution step source:

1. *Enter the general information on the Execution step source*
2. *Define the Executions*
3. *Select a Validation*
4. *Specify the default timeout value*

Table E-2. Workflow Step [Execution] -- Step Number \_\_\_\_.

	Value
Step Name	
Goal / Result of Step	
<b>Validation*</b>	
<b>Execution Type**</b>	
Processing Type	
Timeout (Days)	
Source Environment (Group)	
Dest Environment (Group)	
Security (who can act on step): <ul style="list-style-type: none"> <li>• User Name</li> <li>• Standard Token</li> <li>• User Defined Token</li> </ul>	
Include Notification (Yes/No)	
Notification Event	
Notification Recipient: <ul style="list-style-type: none"> <li>• Username</li> <li>• Email Address</li> <li>• Security Group</li> <li>• Standard Token</li> <li>• User Defined Token</li> </ul>	
Notification Message	
Request Status at Step	
Request % Complete at Step	
Authentication Required (Y/N)	
Authentication Type (if Y)	

Validation Information <sup>3</sup>	Value
Existing Validation?	
New Validation?	
Validation Type: (text field, auto-complete, drop down list, etc.)	
Validation Definition (list of values or SQL)	

Execution Type <sup>3*</sup>	Value
Built-in Workflow Event: <ul style="list-style-type: none"> <li>• Execute Commands</li> <li>• Close</li> <li>• Jump / Receive</li> <li>• Ready for Release</li> <li>• Return from Subworkflow</li> </ul>	
PL/SQL Function	
Token	
SQL Statement	
Workflow step commands	

Information used when adding the step source to the Workflow layout.

Figure 6-3 Information used to create the execution step source.

*Enter the general information on the Execution step source*

Enter the following information in the EXECUTION window.

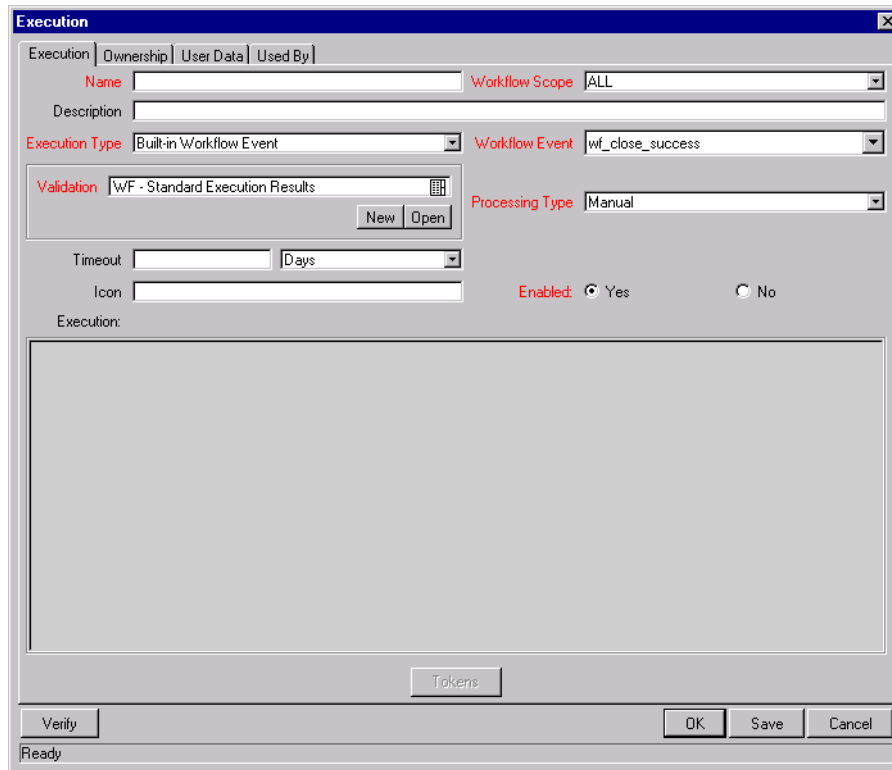


Table 6-2. Execution step source worksheet to window.

Field in Execution Window	Description
NAME	This is the name that describes the 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. This should be set to <b>ALL</b> or <b>REQUESTS</b> for Request resolution processes.
DESCRIPTION	Description of the step source. This should specify the execution that will occur.

Table 6-2. Execution step source worksheet to window.

Field in Execution Window	Description
EXECUTION TYPE	<p>Used to select the type of execution to be performed. The choices are:</p> <p><b>BUILT-IN WORKFLOW EVENT:</b> Executes a predefined command and returns its result as the result of the Step.</p> <p><b>SQL STATEMENT:</b> Executes a SQL statement and returns its result as the result for the Step.</p> <p><b>PL/SQL FUNCTION:</b> Runs a PL/SQL function and returns its result as the result for the Step.</p> <p><b>TOKEN:</b> Calculates the value of a token and returns its value as the result for the Step.</p> <p><b>WORKFLOW STEP COMMANDS:</b> Executes a set of commands, independent of an Object, at a Workflow Step.</p>
WORKFLOW EVENT	<p>For Executions of type 'Built-in Workflow Event', the specific event to perform must be selected. The available choices in the drop down list depend on which Workflow Scope has been selected. The choices are:</p> <p><b>EXECUTE_REQUEST_COMMANDS:</b> Executes the Request Type commands for a Request.</p> <p><b>CREATE_PACKAGE:</b> Generates a Kintana Deliver Package.</p> <p><b>CREATE_PACKAGE_AND_WAIT:</b> Generates a Kintana Deliver Package. The Create step that generates the Package holds it until the Package is closed.</p> <p><b>CREATE_REQUEST:</b> Generates another Kintana Create Request.</p> <p><b>WF_CLOSE_SUCCESS:</b> Sets the Request as closed with an end status of 'Success.'</p> <p><b>WF_CLOSE_FAILURE:</b> Sets the Request as closed with an end status of 'Failed.'</p> <p><b>WF_JUMP:</b> (Kintana Deliver and Kintana Create only) Instructs the Workflow to proceed to a corresponding Receive Workflow Step in another Kintana Workflow.</p> <p><b>WF_RECEIVE:</b> (Kintana Deliver and Kintana Create only) Instructs the Workflow to receive a Jump Workflow Step and continue processing a Request or Package Line initiated in another Workflow.</p> <p><b>WF_RETURN:</b> (Kintana Deliver and Kintana Create only) Used to route a Subworkflow process back to its parent Workflow.</p>
PL/SQL FUNCTION	<p>For Executions of type <b>PL/SQL FUNCTION</b>, the actual function to run. The results of the function will determine the outcome of the step.</p> <p>Note: The results of the function must be a subset of the Validation values for that step.</p>

Table 6-2. Execution step source worksheet to window.

Field in Execution Window	Description
TOKEN	For Executions of type <b>TOKEN</b> , the Token that will be resolved. The results of the Token resolution will determine the outcome of the step.
SQL STATEMENT	For Executions of type <b>SQL STATEMENT</b> , the actual query to run. The results of the query will determine the outcome of the step.  Note: The results of the query must be a subset of the Validation values for that step.
WORKFLOW STEP COMMANDS	For Executions of type <b>WORKFLOW STEP COMMANDS</b> , the actual commands to run. The commands will result with a <b>SUCCEEDED</b> or <b>FAILED</b> value. Use a validation with those values to enable transitioning out of the step based on the execution results.
PROCESSING TYPE	Indicates whether the Execution is performed immediately ( <b>IMMEDIATE</b> ) when the Step becomes eligible or whether the Execution needs to be manually activated by a user ( <b>MANUAL</b> ).
VALIDATION	Specifies the possible values that can exit the Workflow step. See <i>“Configure the Step’s Transition Values (Validation)”</i> on page 81.
TIMEOUT	If this Workflow Step remains eligible for the value entered in the Timeout value, the Request can be configured to send an appropriate Notification, as well as escalate to other steps in the Workflow. See <i>“Specify the default timeout value”</i> on page 81.
ICON	You can select a different graphic to represent this steps of this step source.
ENABLED	The step source must be enabled in order to add it to the Workflow layout.

## Define the Executions

Execution steps are used to perform specific actions. Kintana provides a number of number of built in Workflow events for processing some common execution events (running Request Type commands, closing a Request, etc.). Kintana also provides the flexibility to create your own executions based on SQL, PL/SQL, Token resolution, and Kintana commands.



This section discusses when to use specific types of executions and provides references for configuring these executions.

Execution steps can be created to perform the following actions:

- Execute the Request Type Commands and transition based on the success or failure of those commands.
- Close the Request and mark it as a `SUCCESS`
- Close the Request and mark it as a `FAILURE`
- Transition (jump) to another Workflow that is processing a Package
- Receive control from another Workflow that is processing a Package
- Return from a subworkflow to the parent Workflow
- Execute a PL/SQL function and then transition based on the result
- Execute a SQL statement and then transition based on the result
- Evaluate a Token and then transition based on the result
- Execute a number of system level commands and then transition based on the success or failure of those commands.
  - o Example: Start a server
  - o Example: Stop a server

### Execute the Request Type Commands

Certain process steps may require commands to be performed on them at different points in the resolution process. Kintana allows you to program these commands on a per-Request Type basis. You can then configure your Workflow to execute Request Type commands at a specific step in the process. Each step will run its own commands, ensuring the correct execution for that Request Type.

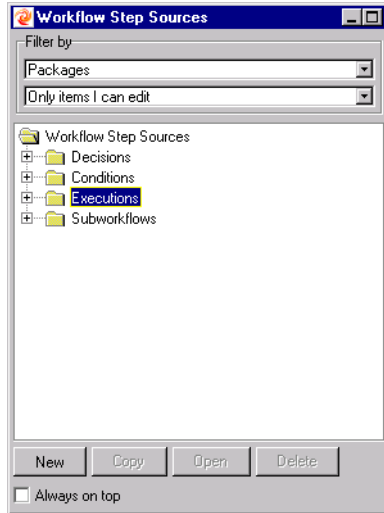
Note

Kintana provides a system step source that executes the Request Type commands. Use this step source unless it doesn't meet your exact specifications (Validation, Processing Type, etc.).

**STEP SOURCE = EXECUTE REQUEST COMMANDS**

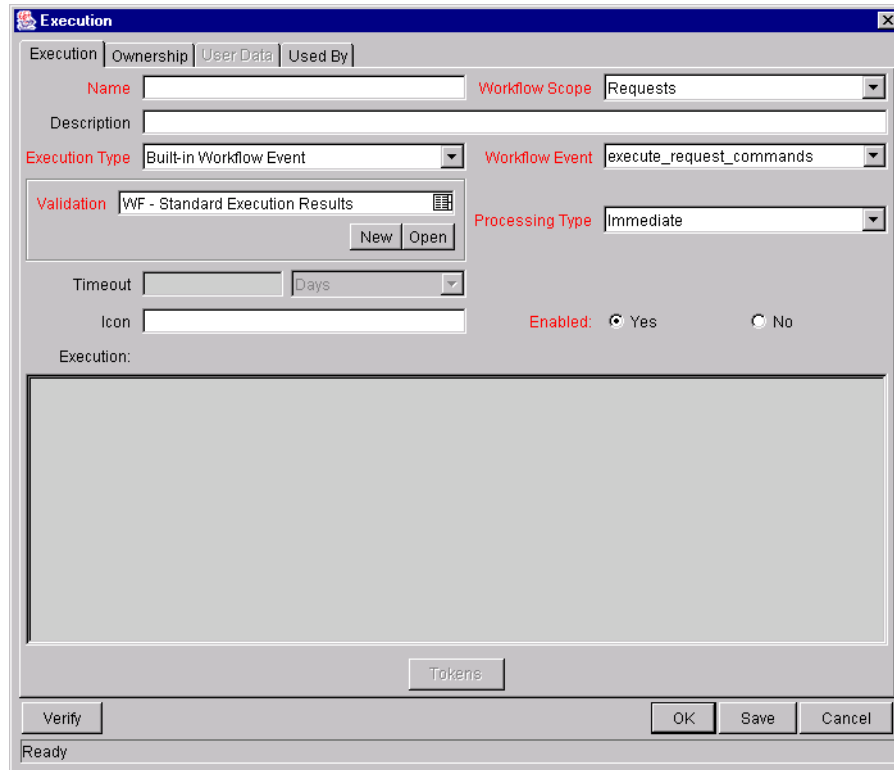
To create an execution step source that will execute the Request Type commands:

1. Open the WORKFLOW WORKBENCH.
2. Select the WORKFLOW STEP SOURCES window.
3. Select the EXECUTION directory.



4. Click **NEW**. The EXECUTION window opens.
5. Enter the following information:

Field in Execution Window	Value
NAME	Enter a descriptive name for the step source.
WORKFLOW SCOPE	<b>REQUESTS</b>
EXECUTION TYPE	<b>BUILT-IN WORKFLOW EVENT</b>
WORKFLOW EVENT	<b>EXECUTE_REQUEST_COMMANDS</b>
PROCESSING TYPE	<b>MANUAL</b> or <b>IMMEDIATE</b>
VALIDATION	<b>WF - STANDARD EXECUTION RESULTS</b> (This is the default selection. You can select another existing or create a new validation.)
ENABLED	<b>YES</b>



### Close the Request and mark it as a Success

You can create an execution step that closes a Request. Each Request Workflow should resolve with a closed Request. You can then report on all Requests that were closed successfully.

To configure an execution step source to close a Request and mark it as a Success, create an execution step source with the following settings:

Field in Execution Window	Value
NAME	Enter a descriptive name for the step source.
WORKFLOW SCOPE	<b>REQUESTS</b>
EXECUTION TYPE	<b>BUILT-IN WORKFLOW EVENT</b>
WORKFLOW EVENT	<b>WF_CLOSE_SUCCESS</b>
PROCESSING TYPE	<b>MANUAL</b> or <b>IMMEDIATE</b>

Field in Execution Window	Value
VALIDATION	<b>WF - STANDARD EXECUTION RESULTS</b> (This is the default selection. You can select another existing or create a new validation.)
ENABLED	<b>YES</b>



Kintana provides a system step source that performs this task. Use this step source unless it doesn't meet your exact specifications (Validation, Processing Type, etc.).

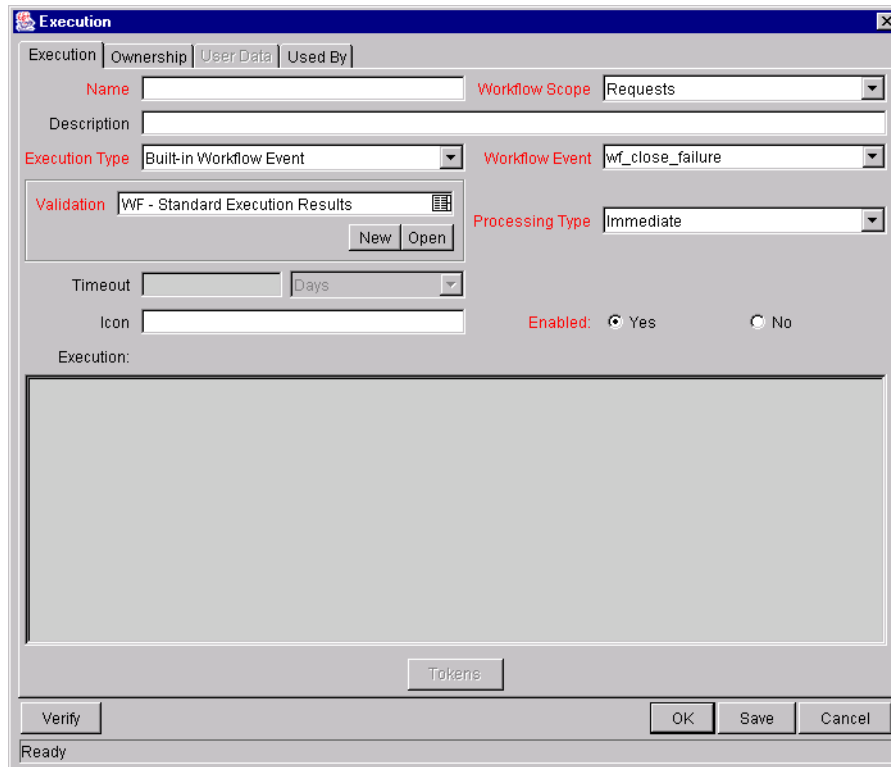
STEP SOURCE = **CLOSE (IMMEDIATE SUCCESS)** or **CLOSE (MANUAL SUCCESS)**

### Close the Request and mark it as Failed

You can create an execution step that closes a Request and marks it as Failed.

To configure an execution step source to close a Request and mark it as a Failed, set the following in the Execution window:

Field in Execution Window	Value
NAME	Enter a descriptive name for the step source.
WORKFLOW SCOPE	<b>REQUESTS</b>
EXECUTION TYPE	<b>BUILT-IN WORKFLOW EVENT</b>
WORKFLOW EVENT	<b>WF_CLOSE_FAILURE</b>
PROCESSING TYPE	<b>MANUAL</b> or <b>IMMEDIATE</b>
VALIDATION	<b>WF - STANDARD EXECUTION RESULTS</b> (This is the default selection. You can select another existing or create a new validation.)
ENABLED	<b>YES</b>





Kintana provides a system step source that performs this task. Use this step source unless it doesn't meet your exact specifications (Validation, Processing Type, etc.).

STEP SOURCE = **CLOSE (IMMEDIATE FAILURE)**

### Transition (jump) to a Workflow that is Processing a Package

Request Workflows can communicate with Package Workflows at specific jump and receive points. To effectively utilize this functionality, you need to properly configure both the jump and receive execution Workflow steps. See [“Advanced Workflow Topics”](#) on page 213 for additional details.

### Receive control from a Workflow that is Processing a Package

Request Workflows can communicate with Package Workflows at specific jump and receive points. To effectively utilize this functionality, you need to properly configure both the jump and receive execution Workflow steps. See [“Advanced Workflow Topics”](#) on page 213 for additional details.

### Return from a Subworkflow to the Parent Workflow

Execution steps can be configured to automatically return from a subworkflow to its parent workflow. Include an execution step with the following configuration:

Field in Execution Window	Value
NAME	Enter a descriptive name for the step source.
WORKFLOW SCOPE	<b>REQUESTS</b>
EXECUTION TYPE	<b>BUILT-IN WORKFLOW EVENT</b>
WORKFLOW EVENT	<b>WF_RETURN</b>
PROCESSING TYPE	<b>MANUAL</b> or <b>IMMEDIATE</b>

Field in Execution Window	Value
VALIDATION	<b>WF - STANDARD EXECUTION RESULTS</b> (This is the default selection. You can select another existing or create a new validation.)
ENABLED	<b>YES</b>



Note

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. 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. See [“Advanced Workflow Topics”](#) on page 213 for additional details.



Tip

Kintana provides a system step source that performs this task. Use this step source unless it doesn't meet your exact specifications (Validation, Processing Type, etc.).

Step Source = RETURN FROM SUBWORKFLOW

### Execute a PL/SQL function and then transition based on the result

A PL/SQL function execution step runs a PL/SQL function and returns its results as the result of that workflow step. Include an execution step with the following source configuration:

Field in Execution Window	Value
NAME	Enter a descriptive name for the step source.
WORKFLOW SCOPE	<b>REQUESTS</b>
EXECUTION TYPE	<b>PL/SQL FUNCTION</b>
PROCESSING TYPE	<b>MANUAL</b> or <b>IMMEDIATE</b>

Field in Execution Window	Value
VALIDATION	Select or create 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 from the execution minus the WHERE clause.
EXECUTION	Enter the PL/SQL function.
ENABLED	<b>YES</b>

### Execute a SQL statement and then transition based on the result

SQL statement Execution steps are used when a Workflow needs to be routed based on the result of a query. A SQL statement execution step runs a SQL query and returns its results as the result of that workflow step.

Include an execution step with the following source configuration:

Field in Execution Window	Value
NAME	Enter a descriptive name for the step source.
WORKFLOW SCOPE	<b>REQUESTS</b>
EXECUTION TYPE	<b>SQL STATEMENT</b>
PROCESSING TYPE	<b>MANUAL</b> or <b>IMMEDIATE</b>
VALIDATION	Select or create 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	<b>YES</b>

#### Configuration notes:

- Only use select statements
- You can use Kintana Tokens within the WHERE clause



- Query must return only 1 value

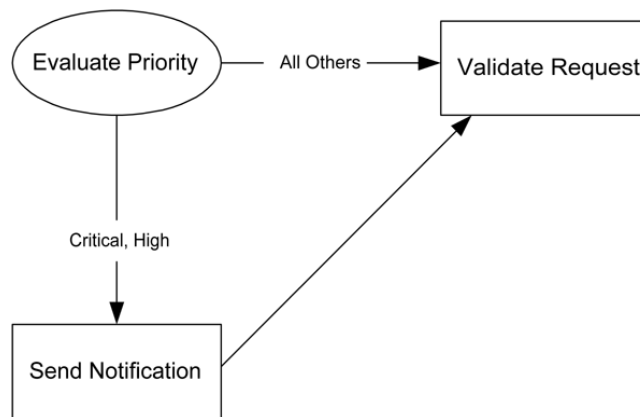
### Evaluate a Token and then transition based on the result

Kintana has workflow Execution steps that may be used to set up data-dependent rules for the routing of Workflow processes. Token Execution steps enable a workflow to be routed based on the value of any field within a particular Kintana entity. A Token Execution step references the value of a given Token and uses that value as the result of the workflow step.

You can transition based on the value stored in Kintana by using Tokens in your Execution step. Include an execution step with the following source configuration:

Field in Execution Window	Value
NAME	Enter a descriptive name for the step source.
WORKFLOW SCOPE	<b>REQUESTS</b>
EXECUTION TYPE	<b>TOKEN</b>
PROCESSING TYPE	<b>MANUAL</b> or <b>IMMEDIATE</b>
VALIDATION	Select or create 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 you would like to transition based on.
ENABLED	<b>YES</b>

For example, ACME needs to send an email Notification to the Validate and Approve Requests group if the Request's PRIORITY is **HIGH** or **CRITICAL**.



They decide to use an Execution 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 step that sends a Notification to the Validate and Approve Requests group before continuing along the Workflow as normal. They create an execution step source (Evaluate Priority), configured with the following parameters.

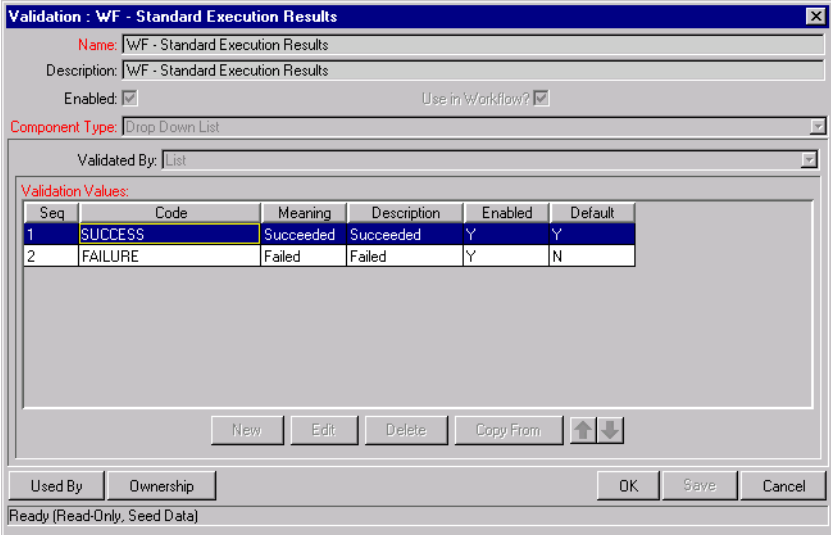
Field in Execution Window	Value
NAME	Evaluate Priority
WORKFLOW SCOPE	<b>REQUESTS</b>
EXECUTION TYPE	<b>TOKEN</b>
PROCESSING TYPE	<b>IMMEDIATE</b>
VALIDATION	<b>CRT - PRIORITY - ENABLED</b>
EXECUTION	<b>[REQ.PRIORITY_CODE]</b>
ENABLED	<b>YES</b>

**Execute a number of system level commands and then transition based on the success or failure of those commands.**

System level commands can be run for execution steps of the following EXECUTION TYPE: **BUILT-IN WORKFLOW EVENT (EXECUTE\_REQUEST\_COMMANDS)** and **WORKFLOW STEP COMMANDS**. When either the Workflow or the Request Type commands execute at this step, the commands will either Succeed or Fail. To

transition based on these results, the code for the validation values must have the following values:

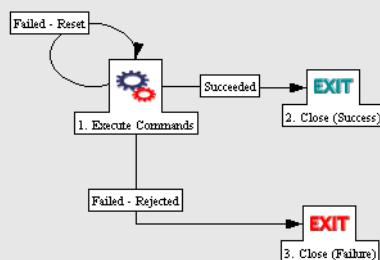
- SUCCESS
- FAILURE





You may want 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. To configure this:

1. Create an execution step source to execute the Workflow or Request Type commands.
2. Create a Validation with the following Validation Values.
  - a. SUCCEEDED
  - b. FAILED
  - c. FAILED - RESET
  - d. FAILED - REJECTED
3. Add the step to the Workflow **LAYOUT** tab.
4. Add transitions based on the following Specific Results:
  - a. SUCCEEDED
  - b. FAILED - RESET -- set the transition to return back into the same step.
  - c. FAILED - REJECTED



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 Execution step and select Failed - Retry. The execution will re-run.

## Select a Validation

Select a Validation that has the transition values required for leaving the step. If Kintana doesn't provide a Validation that meets your requirements, you can create a new one from the WORKFLOW STEP SOURCE window. See “[Validations](#)” on page 243 for a list of Kintana's seeded Validations

See “[Configure the Step's Transition Values \(Validation\)](#)” on page 81 for additional details.

## Specify the default timeout value

Timeouts in the execution steps can be set at two levels:

- Step level: the amount of time that a step is eligible before completing with an error. This is set in the EXECUTION window.
- Command level: 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. It is set in the COMMAND window.

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 can be configured to send an appropriate Notification and escalate to other steps in the Workflow.

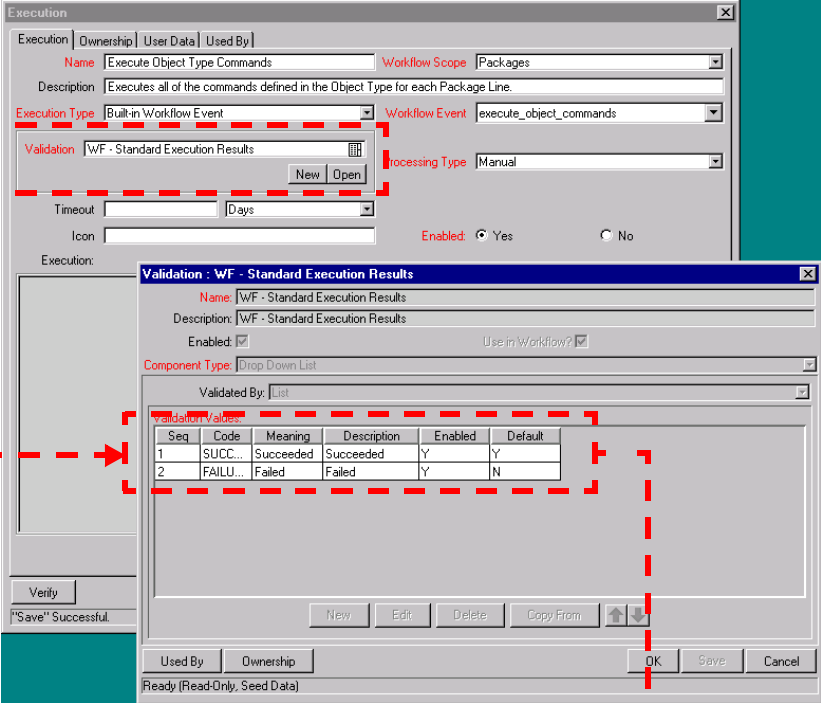
Timeouts can be uniquely configured for each Workflow Step in the LAYOUT tab. The timeout value specified in the step source acts as the default timeout value for the step. When you add a step to the Workflow using this step source, you can specify a different timeout value for the step.

## Configure the Step's Transition Values (Validation)

Kintana Workflows can be configured to transition based on values automatically returned from an execution or values selected by the user. For each Workflow step, you must define all of the possible values for the step's transition. This is set in the Validation field on the EXECUTION window or the

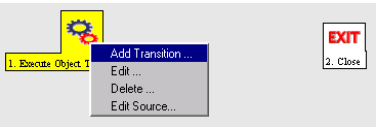
DECISION window. The Validation dictates the values in the SPECIFIC RESULT section on the ADD TRANSITION window.

1. Validation specifies all possible results for the step.

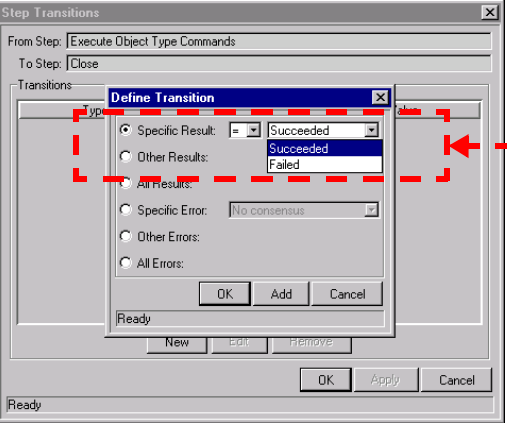


Seq	Code	Meaning	Description	Enabled	Default
1	SUCC...	Succeeded	Succeeded	Y	Y
2	FAILU...	Failed	Failed	Y	N

2. Add a transition between two steps in the Workflow Layout tab.



3. Optionally base the transition on the values defined in the Validation.



When you specify the Validation for the execution step source, you specify all possible transition values in the Validation. When you use that step source on

the Workflow (add it to the Layout tab), you can decide to transition on one of the specific results, or a number of other transition options:

- OTHER RESULTS
- ALL RESULTS
- SPECIFIC ERROR
- OTHER ERRORS
- ALL ERRORS

## 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 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.

The following table summarizes this relationship between Validations and Execution types.

*Table 6-3. Relationship between Validation and Execution Type*

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. Tip: you can use the same SQL in the Validation (drop down or auto-complete list) minus the WHERE clause.



You can use the information captured in the “*Configuration Worksheets*” on page 347 to construct your validation.

Consider copying existing Validations to save time and ensure that the SQL or other Validation technique is configured properly.

## Add Steps and Transitions to the Workflow Layout

Build your process graphically by dragging and dropping Workflow step sources onto the WORKFLOW window’s LAYOUT tab. When a Workflow step source is included in a Workflow, it is then referred to as a “Workflow step.” If Kintana doesn’t provide a step source that meets your requirements (decision, execution, correct transition Validation values, processing type, etc.) you can create one.

When you add the step source to the LAYOUT tab, you will be required to provide supplemental information. The following sections discuss the configuration required when:

- *Adding Decision Steps*
- *Adding Execution Steps*
- *Adding a Subworkflow*
- *Adding Transitions Between Steps*

### Adding Decision Steps

To add a Decision step to your Workflow:

1. Drag the Step Source onto the LAYOUT tab.
2. *Enter the general information on the Decision step*
3. *Specify the Security*
4. *Configure Notifications for the Workflow Step*



Table E-3. Workflow Step (Decision) -- Step Number \_\_\_\_.

	Value
Step Name	
Goal / Result of Step	
<b>Validation*</b>	
Decisions Required (Vote on Step's outcome?)	<ul style="list-style-type: none"> <li>• One</li> <li>• At Least One</li> <li>• All</li> </ul>
Timeout (Days)	
Security (who can act on step):	
<ul style="list-style-type: none"> <li>• Security Group</li> <li>• User Name</li> <li>• Standard Token</li> <li>• User Defined Token</li> </ul>	
Include Notification (Yes/No)	
Notification Event	
Notification Recipient:	
<ul style="list-style-type: none"> <li>• Username</li> <li>• Email Address</li> <li>• Security Group</li> <li>• Standard Token</li> <li>• User Defined Token</li> </ul>	
Notification Message	
Request Status at Step	
Request % Complete at Step	
Authentication Required (Y/N)	
Authentication Type (if Y)	

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

Information used when adding the step source to the Workflow layout.

Figure 6-4 Information used to create the decision step.

### Enter the general information on the Decision step

Enter the following information in the WORKFLOW STEP window.

Table 6-4. Decision Workflow Step window fields

Field/Tab on Workflow Step Window	Description
STEP NAME	This is the name of the step that appears on the WORKFLOW window <b>LAYOUT</b> tab.
ACTION SUMMARY	The text that appears on the action button in the Request status panel.
DESCRIPTION	A description of the step; could describe the goal of the step.
ENABLED	Whether or not the step is enabled. Read-only.
DISPLAY	Whether or not to show the step on the Request status panel.
WORKFLOW PARAMETER	Used to save the results of a workflow step for later use in the workflow processing.
AVG LEAD TIME	A user-specified metric for comparing actual performance to estimated goals. It does not affect any transactional logic.
REQUEST STATUS	The Status acquired by the Request when it reaches this step.

Table 6-4. Decision Workflow Step window fields

Field/Tab on Workflow Step Window	Description
CURRENT % COMPLETE	The percentage of the resolution process that is complete at this step. This value rolls up into a Kintana Drive Project when a Task has been created from a Request.
PARENT ASSIGNED TO USER	If this field is not empty when the step becomes Eligible, the Assigned to User of the Package automatically changes to the user specified in the field.
PARENT ASSIGNED TO GROUP	If this field is not empty when the step becomes Eligible, the Assigned to Group of the Request automatically changes to the Security Group specified in the field.
WORKFLOW STEP INFORMATION	A text entry field in which any URL can be entered. This is where users can find documents, instructions or comments to aid them in working the Workflow Step.
AUTHENTICATION REQUIRED	Determines whether the user acting on the step will need to provide authentication before acting, and if so, what kind.
SECURITY TAB	Determines who can act on the step.
NOTIFICATIONS TAB	Specify who will receive an email notification when this step becomes eligible or has a specific result or error.
TIMEOUT TAB	You can specify the Timeout value for this step. In the Timeout tab, select to use the Workflow step source timeout value or specify your own in the SPECIFIC VALUE section.

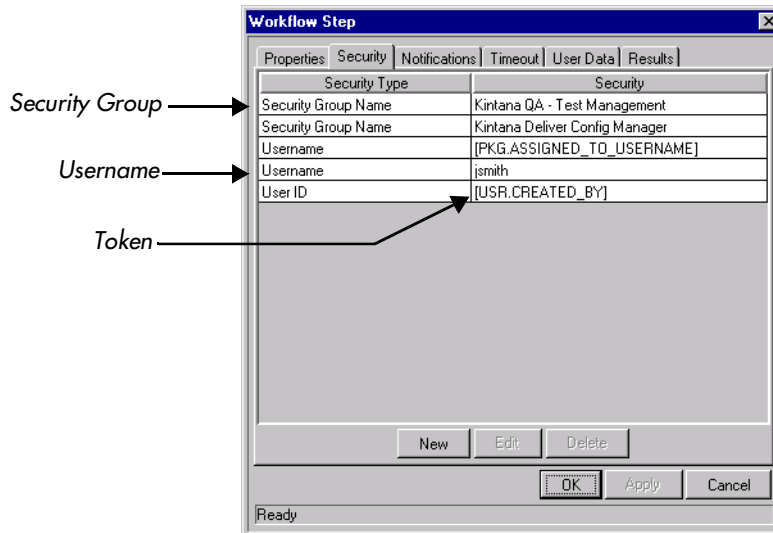
## Specify the Security

*“Integrating Participants into Your Request Resolution System”* on page 159 provides information on setting up security for your Request resolution process. This includes such things as controlling who can create Requests and who can act on specific steps in the process. Security related directly to processing a Workflow step is configured in the WORKFLOW STEP window.

You can define who can act on the step by:

- Security Group

- Username
- Token (standard or user-defined)



Kintana recommends using Security Groups or dynamic access (Tokens) when defining the Workflow step security. You should 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 an organizational reorganization), you would have to update that list in many places on the workflow. By using a Security Group instead of a list of users, you can update the Security Group once, and the changes are propagated throughout the Workflow steps.

### Configure Notifications for the Workflow Step

*“Setting Up Communication Paths”* on page 183 provides information on setting up notifications for steps in your deployment process. This includes such things as configuring the notification’s recipients and message. You can configure notifications for specific steps.

See the following sections for more details:

- *“Establish Communication Points and Visibility”* on page 52
- *“Setting Up Communication Paths”* on page 183

## Adding Execution Steps

To add an Execution step to your Workflow:

1. Drag the Step Source onto the **LAYOUT** tab.
2. *Enter the general information on the Execution step*
3. *Specify the Security*
4. *Configure Notifications for the Workflow Step*

Table E-2. Workflow Step [Execution] -- Step Number \_\_\_\_.

	Value
Step Name	
Goal / Result of Step	
<b>Validation*</b>	
<b>Execution Type**</b>	
Processing Type	
Timeout (Days)	
Source Environment (Group)	
Dest Environment (Group)	
Security (who can act on step):	
<ul style="list-style-type: none"> <li>• User Name</li> <li>• Standard Token</li> <li>• User Defined Token</li> </ul>	
Include Notification (Yes/No)	
Notification Event	
Notification Recipient:	
<ul style="list-style-type: none"> <li>• Username</li> <li>• Email Address</li> <li>• Security Group</li> <li>• Standard Token</li> <li>• User Defined Token</li> </ul>	
Notification Message	
Request Status at Step	
Request % Complete at Step	
Authentication Required (Y/N)	
Authentication Type (if Y)	

Validation Information <sup>3</sup>	Value
Existing Validation?	
New Validation?	
Validation Type: (text field, auto-complete, drop down list, etc.)	
Validation Definition (list of values or SQL)	

Execution Type**	Value
Built-in Workflow Event:	
<ul style="list-style-type: none"> <li>• Execute Commands</li> <li>• Close</li> <li>• Jump / Receive</li> <li>• Ready for Release</li> <li>• Return from Subworkflow</li> </ul>	
PL/SQL Function	
Token	
SQL Statement	
Workflow step commands	

← Information used when adding the step source to the Workflow layout.

Figure 6-5 Information used to create the execution step.

### Enter the general information on the Execution step

Enter the following information in the **PROPERTIES** tab in the **WORKFLOW STEP** window.

Table 6-5. Execution Workflow Step window fields

Field on Workflow Step Window	Description
STEP NAME	This is the name of the step that appears on the Layout tab.
ACTION SUMMARY	The text that appears on the action button in the Request status panel.
DESCRIPTION	A description of the step; could describe the goal of the step.
ENABLED	Whether or not the step is enabled. Read-only.
DISPLAY	Whether or not to show the step on the Request status panel.
WORKFLOW PARAMETER	Used to save the results of a workflow step for later use in the workflow processing.

Table 6-5. Execution Workflow Step window fields

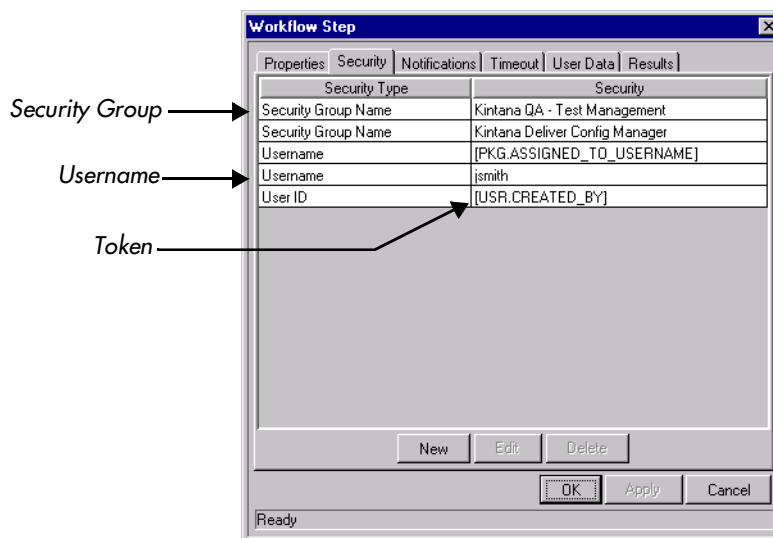
Field on Workflow Step Window	Description
SOURCE ENVIRONMENT	Specifies the Source Environment where the software that is to be changed is located.
SOURCE ENVIRONMENT GROUP	<p>Specifies the Source Environment Group which contains the Environment from which the software change is obtained.</p> <p>The Source Environment Group can also be used in conjunction with the Environment Application Codes to provide a dynamic Source Environment selection.</p>
DEST ENVIRONMENT	Specifies the Destination Environment to which the software change is deployed.
DEST ENVIRONMENT GROUP	Specifies the destination Environment Group. The destination consists of multiple Kintana Environments to which the software change is deployed.
AVG LEAD TIME	A user-specified metric for comparing actual performance to estimated goals. It does not affect any transactional logic.
REQUEST STATUS	The Status acquired by the Request when it reaches this step.
CURRENT % COMPLETE	The percentage of the resolution process that is complete at this step. This value rolls up into a Kintana Drive Project when a Task has been created from a Request.
PARENT ASSIGNED TO USER	If this field is not empty when the step becomes Eligible, the Assigned to User of the Request automatically changes to the user specified in the field.
PARENT ASSIGNED TO GROUP	If this field is not empty when the step becomes Eligible, the Assigned to Group of the Request automatically changes to the Security Group specified in the field.
WORKFLOW STEP INFORMATION	A text entry field in which any URL can be entered. This is where users can find documents, instructions or comments to aid them in working the Workflow Step.
AUTHENTICATION REQUIRED	Determines whether the user acting on the step will need to provide authentication before acting, and if so, what kind.

## Specify the Security

“*Integrating Participants into Your Request Resolution System*” on page 159 provides information on setting up security for your Request resolution process. This includes such things as controlling who can create Requests and who can act on specific steps in the process. Security related directly to processing a Workflow step is configured in the WORKFLOW STEP window.

You can define who can act on the step by:

- Security Group
- Username
- Token (standard or user-defined)



Kintana recommends using Security Groups or dynamic access (Tokens) when defining the Workflow step security. You should 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 an organizational reorganization), you would have to update that list in many places on the workflow. By using a Security Group instead of a list of users, you can update the Security Group once, and the changes are propagated throughout the Workflow steps.



## Configure Notifications for the Workflow Step

*“Setting Up Communication Paths”* on page 183 provides information on setting up notifications for steps in your deployment process. This includes such things as configuring the notification’s recipients and message. You can configure notifications for specific steps.

See the following sections for more details:

- *“Establish Communication Points and Visibility”* on page 52
- *“Setting Up Communication Paths”* on page 183

## Adding a Subworkflow

A Subworkflow can be selected from the **WORKFLOW STEP SOURCES** window and dragged onto the **LAYOUT** tab. When the Request 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.

To add an enabled Subworkflow to another Workflow:

1. Select the desired Subworkflow and drag it to the **LAYOUT** tab. The **WORKFLOW STEP** window opens.

The screenshot shows the 'Workflow Step' configuration window. It features a tabbed interface with 'Properties' selected. The 'Step Name' is 'Review and Test Changes'. The 'Enabled' radio button is set to 'Yes'. The 'Display' dropdown is set to 'Always'. The 'Source Type' is 'Workflow' and the 'Source Name' is 'Review and Test Changes'. There are also fields for 'Source Environment', 'Dest Environment', 'Avg Lead Time', 'Project Status', 'Parent Assigned To User', and 'Parent Assigned To Group'. The window is signed by Kintana, Inc.

This window contains preconfigured information which is specific to the selected Workflow Step.

2. Configure this step as you would configure an execution or decision step.
3. Click **OK**.

See *“Advanced Workflow Topics”* on page 213 for a detailed discussion of using Subworkflows in Kintana.

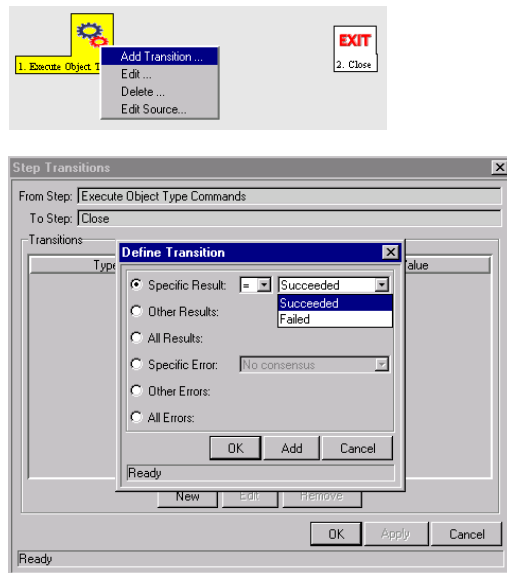
## Adding Transitions Between Steps

After adding the steps to the Workflow **LAYOUT** tab, you need to configure the transitions between them. You can choose to transition between steps based on the following step results:

- **SPECIFIC RESULT** (based on the Validation configured in the step source)
- **OTHER RESULTS**
- **ALL RESULTS**

- **SPECIFIC ERROR**
- **OTHER ERRORS**
- **ALL ERRORS**

The following sections provides some example scenarios and transition configuration options:

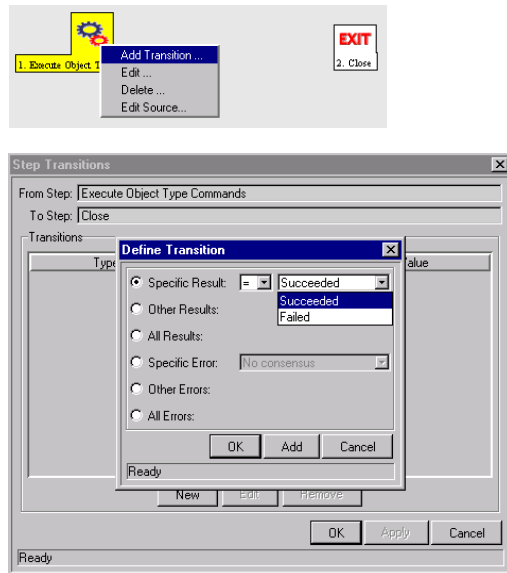


### *Transition based on a specific result*

Transitioning based on the result of a specific decision or execution is the most basic transition method in Kintana. It allows you to branch your business process based on anticipated results of a step in the Workflow.

To transition based on a specific Workflow step result:

1. Add a transition between two steps by right-clicking on a step, selecting **ADD TRANSITION**, and connecting the arrow to the appropriate target step. The **DEFINE TRANSITION** window opens.
2. Select the **SPECIFIC RESULT** radio button.
3. Select the desired result from the drop down list. The values in this list will vary depending on the Validation set in the Workflow step source for the transitioning step.



### *Transition based on a value in a field*

You can transition a Request based on the value in a particular field. This can be a general field in the Request Header (Priority, Assigned To, Request Group, etc.) or a custom field specified in the Request (defined on the Request Type). For example, if the Request's PRIORITY field is set to **CRITICAL**, then you may want the Request to follow a different, more robust process. This is done by resolving a Kintana field Token in a Workflow execution step. The Workflow engine evaluates the field's value at a specific step and then can route the Request accordingly.

To transition based on the value in a field

1. Add an immediate execution step source to the Workflow. You may have to create a custom Workflow step source for this operation. The step source should be configured as follows:

Field in Execution Window	Value
WORKFLOW SCOPE	REQUESTS
EXECUTION TYPE	TOKEN
PROCESSING TYPE	IMMEDIATE

Field in Execution Window	Value
VALIDATION	Select or create 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 <b>[REQ.PRIORITY_CODE]</b> token and the <b>CRT - PRIORITY - ENABLED</b> 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.
ENABLED	<b>YES</b>

2. Add transition between two steps. The DEFINE TRANSITION window opens.
3. Select the SPECIFIC RESULT radio button.
4. Select the desired result from the drop down list. The values in this list will vary depending on the Validation set in the Workflow step source for the transitioning step. For the above Priority example, the possible values will be the values allowed in the Request's PRIORITY field.

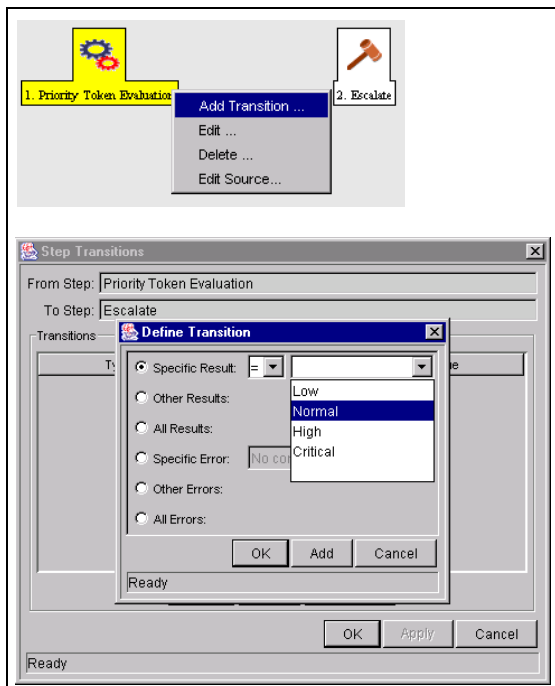


Figure 6-6 Example: Transitioning based on value in a field (Token)

### *Transition based on data in a table*

You can transition based on information stored in a table. To transition using this method, you must use a Workflow execution step with an execution type of SQL. See [“Execute a SQL statement and then transition based on the result”](#) on page 76 for more information.

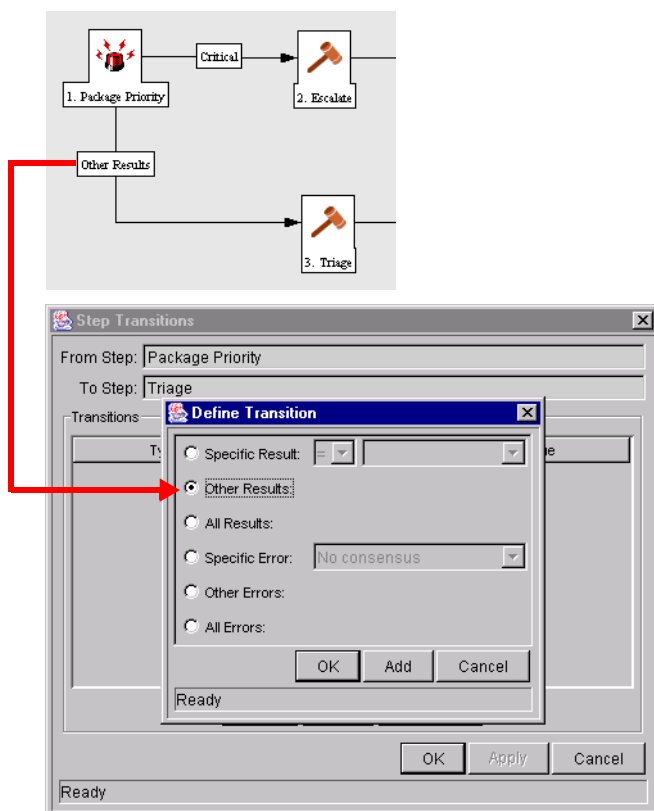
When transitioning from a properly configured execution step (Execution Type = SQL Statement), you will transition based on a Specific Result. The possible results are defined in the Workflow step source’s Validation. The values in this list are determined by a SQL query of a database table.

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

### *Transition based on all but one specific value*

You can transition based on all but one specified value. For example, you want to transitional all “Critical” Requests one way and all other results another. To configure this:

1. Create a transition from a step based on a specific result.
2. Create another transition from the same step and specify **OTHER RESULTS**.



In the above example, only Requests with a “Critical” priority will follow the ESCALATE path. All other results are sent to TRIAGE.

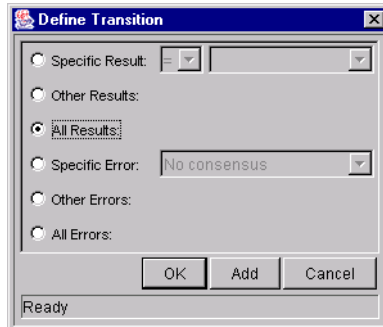


You can use OTHER RESULTS when multiple transitions are exiting a single step. OTHER RESULTS will act as the transition if none of the other explicit transition conditions are satisfied.

### Transition based on all results

You can define a Request to transition regardless of the step’s actual results. For example, you want to run a subworkflow to perform server maintenance after the on-call server contact is identified. To do this, add a transition from the SPECIFY CONTACT step to the subworkflow. Because the next step in the process doesn’t 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**. The DEFINE TRANSITIONS window is shown below.



Tip

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.

### *Transition based on error*

You can transition based on a specific error that occurs during an execution step. This allows you to branch your business process based on likely execution errors such as **TIMEOUT**, **COMMAND EXECUTION** or **INVALID TOKEN**.

To transition based on a specific Workflow step error:

1. Add transition between two steps. The **DEFINE TRANSITION** window opens.
2. Select the **SPECIFIC ERROR** radio button.
3. Select the error from the drop down list. All values in this list are defined in [Table 6-6](#).



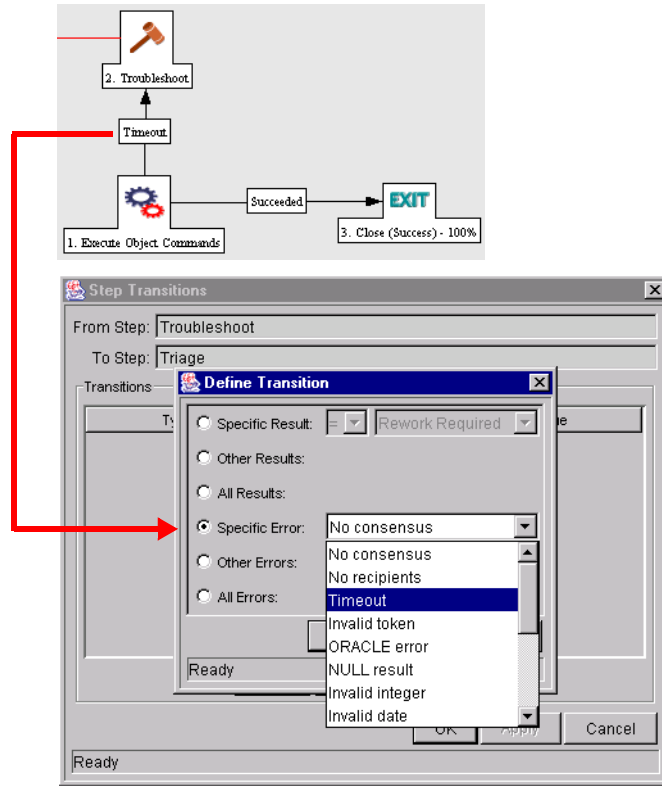


Table 6-6. Workflow Transition Errors

Transition Option	Meaning
<b>MULTIPLE RETURN RESULTS</b>	When the Package Level subworkflow receives multiple results from Package Lines that traversed through it.
<b>NO CONSENSUS</b>	When all users of all Security Groups, or users linked to the Workflow Step need to vote, and there is no consensus.
<b>NO RECIPIENTS</b>	When none of the Security Groups linked to the Workflow Step has users linked to it, no user can act on the Workflow Step.
<b>TIMEOUT</b>	When the Workflow Step times out. Used for Executions and Decisions.
<b>INVALID TOKEN</b>	Invalid Token used in the execution.
<b>ORACLE ERROR</b>	Failed PL/SQL Execution.
<b>NULL RESULT</b>	No result is returned from the execution.
<b>INVALID INTEGER</b>	Validation includes an invalid value in the Integer field.
<b>INVALID DATE</b>	Validation includes an invalid value in the Date field.

Table 6-6. Workflow Transition Errors

Transition Option	Meaning
<b>COMMAND EXECUTION ERROR</b>	Execution engine has failed or has a problem.
<b>INVALID RESULT</b>	Execution or Subworkflow has returned a result not included in the Validation.
<b>PARENT CLOSED</b>	For wf_receive or wf_jump steps, a Package Line is expecting a message from a Request that is cancelled or closed.
<b>CHILD CLOSED</b>	For wf_receive or wf_jump steps, a Request is expecting a message from a Package Line that is cancelled or closed.
<b>NO PARENT</b>	For wf_receive or wf_jump steps, a Package Line is expecting a message from a Request that has been deleted.
<b>NO CHILD</b>	For wf_receive or wf_jump steps, a Request is expecting a message from a Package Line that has been deleted.
<b>MULTIPLE JUMP RESULTS</b>	For wf_jump steps in a Package Line, different result values were used to transition to the step.

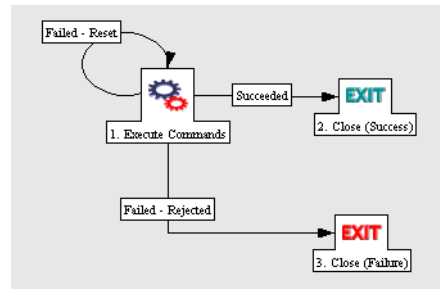
### *Transition back to the same step*

You may want 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. To configure this:

1. Create an execution step source to execute the Workflow or Request Type commands.
2. Create a Validation with the following Validation Values.
  - A. **SUCCEEDED**
  - B. **FAILED**
  - C. **FAILED - RESET**
  - D. **FAILED - REJECTED**
3. Add the step to the Workflow **LAYOUT** tab.
4. Add transitions based on the following Specific Results:
  - A. **SUCCEEDED**

b. **FAILED - RESET** -- set the transition to return back into the same step.

c. **FAILED - REJECTED**



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.

Note

Be careful when using an immediate execution step that the **FAILED** result isn't feeding directly back into the execution step. This would result in a continual execution-failure loop.

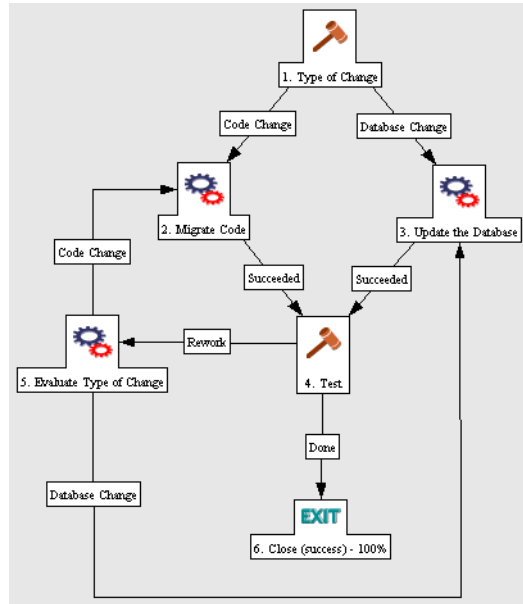
### *Transition based on a previous workflow step result (parameters)*

You can use Workflow parameters to store the result of a workflow step. This value can then be used later to define a transition. To configure this, you need to:

1. Create a WORKFLOW PARAMETER in the WORKFLOW window.
2. Specify that WORKFLOW PARAMETER in a Workflow step on the **LAYOUT** tab.
3. Create a token execution step that will resolve the value in the Workflow parameter.

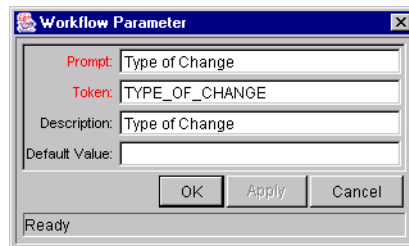
#### **Example: Using a Workflow Parameter to Transition**

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.

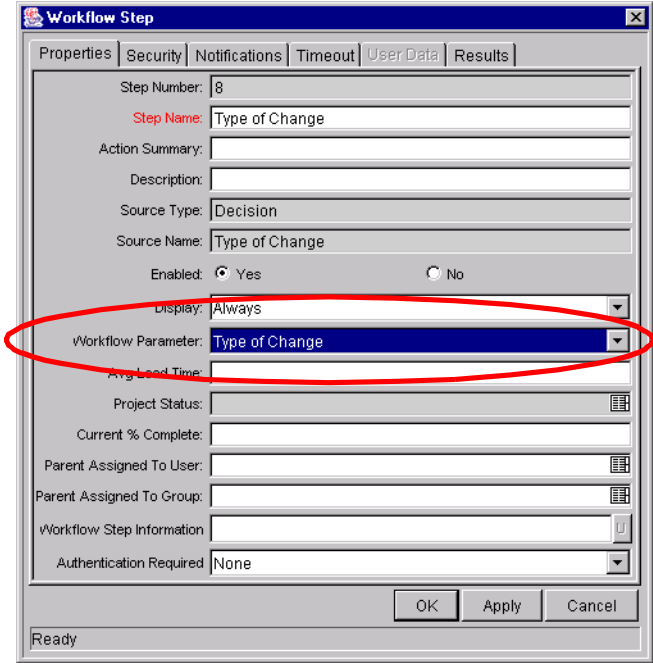


To enable this process, set the following in the Workflow:

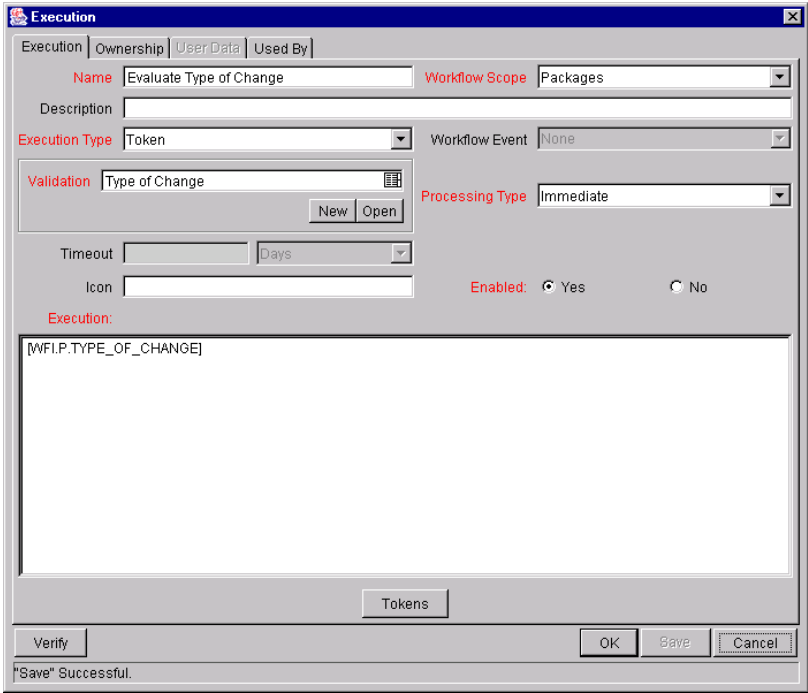
1. Create a **WORKFLOW PARAMETER** in the **WORKFLOW** window. This is done by clicking **ADD** in the **WORKFLOW** tab on the **WORKFLOW** window.



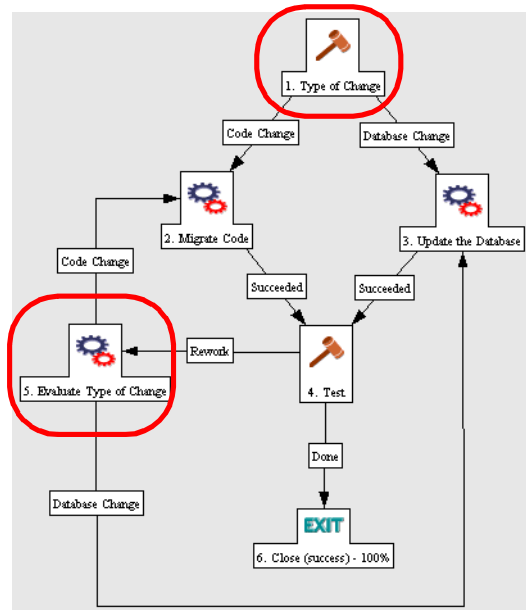
2. Select **TYPE OF CHANGE** for the **WORKFLOW PARAMETER** in **TYPE OF CHANGE** Workflow step on the **LAYOUT** tab.



- 3. Create a token execution step that will resolve 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.



4. Add the steps and transitions as shown below.



### *Transition to and from Subworkflows*

There are special configuration requirements when transitioning to and from Subworkflows. See detailed instructions in [“Advanced Workflow Topics”](#) on page 213.

### *Transition to and from a Package Workflow*

There are special configuration requirements when transitioning to and from a Package Workflow using Jump and Receive steps in the Workflows. See detailed instructions in [“Advanced Workflow Topics”](#) on page 213.

# Chapter 7

## Constructing the Request Type

This chapter provides an overview for how to configure the Kintana Request Type that will be used to process Requests through your Request resolution Workflow. This includes configuring Request Type and Request Header Type fields, as well as Request Statuses, Status Dependencies, Rules, and Commands.

This chapter discusses the following topics:

- [Creating a Request Type - Overview](#)
- [Choosing a Request Header Type](#)
- [Request Type Field Validations](#)
- [Configuring Field Behavior - Overview](#)
- [Creating a Field](#)
- [Configuring Request Type Defaulting Behavior \(Rules\)](#)
- [Configuring Field Behavior Using Status Dependencies](#)
- [Modifying the Request Type Layout](#)



Use the Worksheets in “[Configuration Worksheets](#)” on page 347 to help you gather and manage information required to build a Request Type.

### Creating a Request Type - Overview

Request Types are created and configured using the Kintana Workbench. To create a new Request Type:

1. Click the **CREATE** screen group on the Workbench and click the **REQUEST TYPES** icon. The REQUEST TYPE WORKBENCH window opens.
2. Click **NEW REQUEST TYPE**. The REQUEST TYPE window opens.
3. Enter the Request Type general information. This includes the Request Type's NAME, DESCRIPTION, CREATION ACTION NAME, META LAYER VIEW and CATEGORY.
4. Select a Request Header Type to be used with this Request Type. You can either select an existing Request Header Type from the auto-complete list or create a new Request Header Type by clicking New. See [“Choosing a Request Header Type”](#) on page 109 for more detailed information.
5. Create fields that describe your Request. See [“Request Type Field Validations”](#) on page 117. This includes configuring the following:
  - o Field names
  - o Validations and component types (dictated by the validation)
  - o Field Behaviors: whether fields are displayed or have any defaulting behavior, etc. Other aspects of field behavior should be taken into account later, and are discussed below.
6. Configure the Fields' layout. This will determine how the fields are positioned on the Request. See [“Modifying the Request Type Layout”](#) on page 154.
7. Determine what, if any, fields have more intricate defaulting behaviors and configure them in the **RULES** tab of the REQUEST TYPE window. For example, you can configure a set of fields to automatically populate based on a change to another field in the Request. See [“Configuring Request Type Defaulting Behavior \(Rules\)”](#) on page 135 for more detailed information.
8. Link or create the Statuses you will need the Request to take on as it moves through its Workflow. Request Statuses are linked to Workflow Steps. Set Status Dependencies on fields to define additional field behavior. See [“Creating Your Request Statuses”](#) on page 147 for more detailed information.
9. Open the Workflow to be used by the Request Type and link the proper Request Statuses to the appropriate Workflow Steps. See [“Assigning Request Statuses to Workflow Steps”](#) on page 153 for more detailed information.



10. Create the Request Type's commands, if any have been determined necessary. Request Type commands can be useful for carrying out complex operations on fields. See "[Defaulting](#)" on page 124 for an example.
11. Set Ownership for the Request Type. This controls who can modify or delete the Request Type. See "[Kintana Security Model](#)" for details.
12. Set up any desired Notifications for Request field changes. See "[Setting Notifications on Request Field Changes](#)" on page 202 for more details.



Tip

It is often advantageous to use the **COPY** functionality to copy an existing Request Type and then edit the new copy. To reduce the amount of editing required choose an existing Request Type similar to the Request Type to be generated.



Note

Only Kintana users with the appropriate security can create or edit Request Types. To edit Request Types, you must belong to a Security Group that has the access grant CREATE: EDIT REQUEST TYPES. See the "[Integrating Participants into Your Request Resolution System](#)" on page 159 for more information.

## Choosing a Request Header Type

Request Header Types are used to define common Request header field configurations to be used in multiple Request Types. Kintana uses Request Header Types to quickly apply a preconfigured set of header fields to a Request Type.



### Request Header Type

A Request Header Type can be thought of as a basic template for the header area that appears in a Request. Request Header Types perform the following functions:

- Provide a framework for the storage and manipulation of Request header data
  - Header data represents attributes common to multiple types of Requests. Header data is useful for locating and reporting certain types of Requests. Examples of Header Data are CREATOR, ASSIGNED USER, DESCRIPTION, SUMMARY, and DEPARTMENT
- Label and arrange header fields in a manner most familiar to specific Business Units
- Remove header fields from Requests where that data field is irrelevant or distracting. For example, in a Request for novice users, the Workflow, assignment, and contact fields can be hidden

Each Request Type is associated with a Request Header Type. This association determines how the header fields are presented and validated when Requests of that type are viewed. Therefore, it is possible for every type of Request to have a unique view of the header fields.

The base installation of Kintana Create is delivered with the following default Request Header Types. These system Request Header Types are provided in [Table 7-1](#).

*Table 7-1. System Request Header Types*

System Header Type	Description
(REFERENCE) Default	The default Request Header Type. Includes a % Complete field.
(REFERENCE) Comprehensive	Displays all information. Consistent with previous versions of Kintana Create.
(REFERENCE) Simple	Displays only the most essential information.
(REFERENCE) Departmental	An example Header Type for simple cross-departmental Requests.
(REFERENCE) Application	An example Header Type for simple cross-application Requests.

Table 7-1. System Request Header Types

System Header Type	Description
(REFERENCE) Help Desk	An example Header Type for help desk Requests, including contact and assignment information.

Request Header Types can map to more than one Request Type. If you have fields in mind that might come in handy in more than one Request Type, consider putting them in the Request Header Type to save configuration time.

If none of the system Request Header Types are adequate for your intended Request Type, you can copy and modify any of them as you see fit. New Request Header Types can also be created from scratch. See [“Copying a Request Header Type”](#) on page 115 for more details on copying an existing Request Header Type.

Every system Request Header Type comes with the same set of fields; they differ in the particular fields that are disabled, set to display only, activated for transaction or Notes history, etc. New fields can also be created specifically for a single Request Header Type. See [“Creating New Request Header Type Fields”](#) on page 115 for more detailed information.

If you have found an existing Request Header Type that meets your needs, the next step is to make a new Request Type and begin creating fields. See [“Request Type Field Validations”](#) on page 117 for more details.

Note

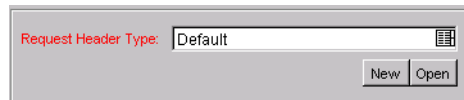
When Field Groups are associated with existing Request Types (through the Request Header Type definition), tables in the Kintana database are updated to handle this new configuration. Because of the scope of database changes, you should re-run the Database Statistics on your Kintana Database. Instructions for this are included in the Kintana System Administration Guide. Contact your System Administrator for help with this procedure.

## Creating a New Request Header Type

To create a new Request Header Type:

1. Click the **CREATE** screen group on the Workbench and click the **REQUEST HEADER TYPES** icon. The REQUEST HEADER TYPE WORKBENCH window opens.
2. Click **NEW REQUEST HEADER TYPE**. The REQUEST HEADER TYPE window opens.
3. Fill in any general information for the Request Header Type.

4. Modify the existing fields in the Request Header Type as you see fit. This includes field security. For more information, see [“Modifying Existing Request Header Type Fields”](#) on page 112.
5. (optional) Create any new Request Header Type fields you deem necessary. Field security can be configured for new Request Header Type fields. For more information, see [“Creating New Request Header Type Fields”](#) on page 115.
6. Save the Request Header Type.
7. Open the Request Type you wish to associate with the Request Header Type.
8. Specify the Request Header Type in the REQUEST HEADER TYPE field.



9. Save the Request Type.

### *Modifying Existing Request Header Type Fields*

The single most important difference between creating a new Request Header Type and a new Request Type is that a new Request Header Type already has fields. These fields can be modified.

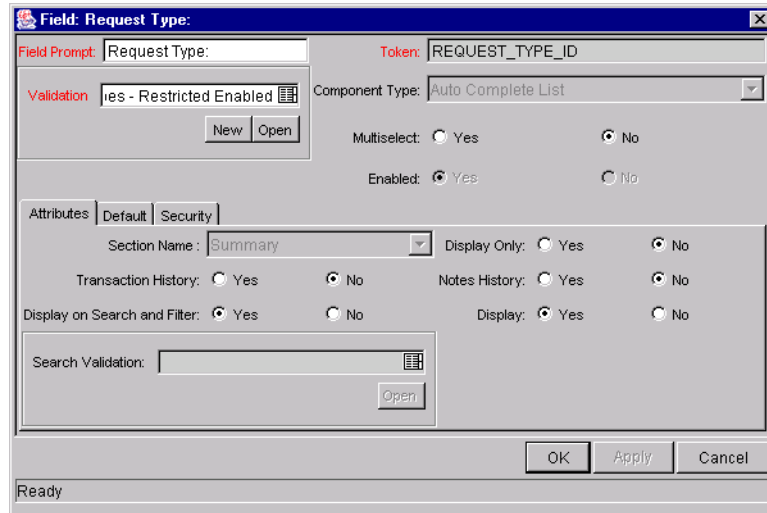


Note

There are some important differences between Request Header Type fields and Request Type fields:

- The Validation for an existing system-provided Request Header Type field cannot be changed. A different Validation can be specified when creating a new Request Header Type field or copying a Request Header Type field.
- Request Header Type fields can be used with many Request Types simultaneously. This may necessitate specifying a different Validation to be used in Search/Filter functionality. The SEARCH VALIDATION can be specified from the Request Header Type FIELD window's **ATTRIBUTES** tab.

Selecting a field and clicking **EDIT** in the **FIELDS** tab of the REQUEST HEADER TYPE window opens the FIELD window.



The FIELD window controls the behavior of the Request Header Type field. For discussion of each field, as well as what can and cannot be altered on existing system-provided Request Header Type fields, see [Table 7-2](#).

*Table 7-2. Request Header Type Field Window - General Information Region*

Field	Behavior
FIELD PROMPT	The prompt visible for the Request Header Type field on the Request.
TOKEN	An uppercase text string used to identify this field. The token name must be unique for the specific Request Header Type. An example of a token name is ASSIGNED_TO_USER_ID. On existing Request Header Type fields, this field cannot be edited.
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, etc. See <a href="#">“Determining the Field Type (Selecting a Validation)”</a> on page 118 for more details.
COMPONENT TYPE	Defines the visual characteristics of the field (drop down list, free form text field, etc.). This is derived from the Validation chosen. This field cannot be edited.
MULTISELECT	Determines whether or not the field allows users to select more than one entry. Only valid for fields with an auto-complete component for the Validation.
ENABLED	Determines whether or not the field is turned on for this Request Header Type.

*Table 7-3. Request Header Type Field Window - Attributes Tab*

<b>Field</b>	<b>Behavior</b>
SECTION NAME	The area of the Request on which the field is displayed.
DISPLAY ONLY	Determines 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 the field is set to <b>YES</b> , whenever it 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 the field is set to <b>YES</b> , whenever it changes in a Request, the change is logged in Notes for the Request.
DISPLAY ON SEARCH AND FILTER	Determines whether or not the field will be displayed in Search and Filter pages in the Kintana interface.
DISPLAY	Determines whether or not the field is seen by Requests that use the given Request Header Type. If Disabled, the Request Header Type field will no longer be displayed.
SEARCH VALIDATION	Useful for fields that are used in multiple Request Header Types. Allows you to specify a Validation to be used as the Search/Filter Validation across all Request Header Types.

*Table 7-4. Request Header Type Field Window - Default Tab*

<b>Field</b>	<b>Behavior</b>
DEFAULT TYPE	Defines whether or not the field will have a default value. Either default the field with a constant value ( <b>DEFAULT FROM CONSTANT</b> ) or default it from the value in another field ( <b>DEFAULT FROM PARAMETER</b> ).
VISIBLE VALUE	Defines the constant value that defaults in the field. Enabled only when the DEFAULT TYPE of <b>DEFAULT FROM CONSTANT</b> is selected.

*Table 7-5. Request Header Type Field Window - Storage Tab (for new fields)*

<b>Field</b>	<b>Behavior</b>
MAX LENGTH	Determines the maximum field character length. The two possible values are 200 and 1800.

Table 7-5. Request Header Type Field Window - Storage Tab (for new fields)

Field	Behavior
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	Determines 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 Header Type. Information can be stored in up to 50 columns using Request Header Types, allowing up to 50 fields/batch. No two fields in a Request Header Type can use the same column number within the same batch.

Table 7-6. Request Header Type Field Window - Security Tab

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 <i>“Creating a Field”</i> on page 126 for more detailed information.

## Creating New Request Header Type Fields

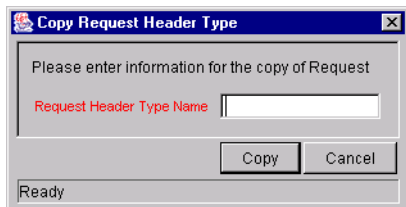
If none of the existing Request Header Type fields match your needs, you can create brand-new fields. New Request Header Type fields can be created and modified in much the same way as Request Type fields. See *“Request Type Field Validations”* on page 117 for more detailed information.

## Copying a Request Header Type

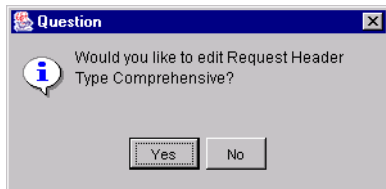
To copy a Request Header Type:

1. Click the **CREATE** screen group on the Workbench and click the **REQUEST HEADER TYPES** icon. The REQUEST HEADER TYPE WORKBENCH window opens.

2. Enter any necessary search criteria and click **LIST**. The **RESULTS** tab displays the results of your search.
3. Select the desired Request Header Type and click **COPY**. The COPY REQUEST HEADER TYPE window opens.



4. Enter a new REQUEST HEADER TYPE NAME and click **COPY**.
5. A question dialog opens, asking if you would like to edit the new Request Header Type.



6. Click **YES**. The REQUEST HEADER TYPE window opens.
  7. Fill in any general information for the Request Header Type.
  8. Modify the existing fields in the Request Header Type as you see fit. For more information, see [“Modifying Existing Request Header Type Fields”](#) on page 112.
  9. (optional) Create any new Request Header Type fields you deem necessary. For more information, see [“Creating New Request Header Type Fields”](#) on page 115.
- Save the Request Header Type.



## Request Type Field Validations

Request Type fields define the information collected from the end users when a Request is created, and during the Request resolution process. You can configure prompts, tokens, and validations for each field in a Request Type.

In addition, field properties can change based on the Status of the Request, which in turn can be set by the Workflow. As the Request moves through its lifecycle, these changes in field properties can reflect your business process.



To resolve a software bug, Kintana Create must know the severity of the bug's impact. A field named `IMPACT` can be created to capture that information.

A screenshot of a form field. The label "Impact" is on the left. To its right is a rectangular input box with a small downward-pointing arrow on the right side, indicating it is a dropdown menu.

Impact

The general steps for configuring a Request Type field are:

1. Open the `REQUEST TYPE` window.
2. Click **NEW**. The `FIELD` window opens.
3. Enter the general field information: `FIELD PROMPT`, `TOKEN`, and `DESCRIPTION`.
4. Select a `Validation` for the field. If a `Validation` doesn't exist that meets your needs, you can create one. The `Validation` dictates the possible values that can be entered in the field. They also dictate the field type (text field, drop down list, date field, etc.).
5. Configure the field's behavior. This includes the following aspects of the field:
  - Visibility
  - Editability
  - Basic defaulting
  - Field properties

Field behavior configuration consists of setting options in the FIELD window's **ATTRIBUTES**, **DEFAULT**, **STORAGE**, and **SECURITY** tabs, as well as in the Request Type's **RULES** and **STATUS DEPENDENCIES** tabs. See [“Configuring Field Behavior - Overview”](#) on page 122. Note that some field behavior is dependent on other Request Type fields. You may have to revisit this step after creating the other fields in your Request Type.

6. Enable the field.



You can copy fields from other Request Types and modify them to suit your needs. See [“Copying a Request Type Field”](#) on page 132 for more detailed information.

## Determining the Field Type (Selecting a Validation)

When configuring your Request Type or Request Header Type, you can specify a different Validation for each field (except existing system-provided Request Header Type fields). The Validation dictates the possible values that can be entered in the field. They also dictate the field type (text field, drop down list, date field, etc.). The following sections provide some general information related to Validations.

- [“Available Field Types”](#) on page 118
- [“Selecting the Validation”](#) on page 120
- [“Building a Validation”](#) on page 121

See [“Validations”](#) on page 243 for more detailed implementation instructions.

### *Available Field Types*

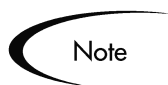
Fields located in the Request Type can be of the following types.

Table 7-7. Field types.

Field Type	Description
Text Field, Text Area	Text fields and text areas are generic entry fields. Text fields are displayed on a single line, while text areas are displayed on multiple lines using a scroll bar if necessary. The values that are entered can be constrained. If an attempt is made to type non-conforming values into a text field or text area, the entries are ignored. For example, if the letter "A" is typed into a numeric field, the character does not appear.
Drop down list	Field that allows the user to select from a predefined set of values. The values in a drop down list can be specified in two ways: <ul style="list-style-type: none"> <li>• In the Validated By field, by selecting List to enter specific values.</li> <li>• By selecting SQL to use a SQL statement to build the contents of the list.</li> </ul>
Auto-complete list	Field that allows the user to select from a predefined set of values. The values in an auto-complete list can be specified in the following ways. In the VALIDATE BY field, select one of the following: <ul style="list-style-type: none"> <li>• List: used to enter specific values.</li> <li>• SQL: uses a SQL statement to build the contents of the list.</li> <li>• Command With Delimited Output: uses a system command to produce a character-delimited text string and uses the results to define the list.</li> <li>• Command With Fixed Width Output: uses a system command to produce a text file and parses the result on the basis of the width of columns, as well as the headers.</li> </ul>
Radio Button	Radio buttons are used for fields where there is a possible Yes/No choice. Selecting on option disables the other. For example, clicking Yes in a Yes/No radio button pair disables the No option. To select a choice, click the button to the left of the appropriate choice.
Date Field	Date fields can accept a variety of formats. The current date field Validations are separated into two categories: all systems, and systems using only the English language.

Table 7-7. Field types.

Field Type	Description
Web Address (URL)	The Web Address field is a generic text entry field in which any URL can be entered. When this field is used, a U button appears next to the field. If U is clicked, a Web page is opened using the field value as the Web address.
Password	The Password Field component type creates a text field with an associated <b>C</b> button. Data is entered through a dialog that asks for the new password and a verification of the password. The text is then displayed in the field as *****.
Table Component	Used to enter multiple records into a single Kintana component. The table component can be configured to include multiple columns of varied data types. Additionally, this component supports rules for populating elements within the table and provides functionality for capturing column totals. See <i>“Configuring the Table Component”</i> on page 272 for details.
Budget	Field that can be added to the Request Type to enable access to view, edit or create Budgets associated with a Request or Project.
Staffing Profile	Field that can be added to the Request Type to enable access to view, edit or create Staffing Profiles associated with a Request or Project.
Resource Pool	Field that can be added to the Request Type to enable access to view, edit or create Resource Pools associated with a Request or Project.



Fields of type Directory Chooser and File Chooser cannot be used in Request Types.

### Selecting the Validation

Use the information gathered in *“Gathering Process Requirements and Specifications”* on page 35 to determine the appropriate Validation for the Request Type field. If a Validation does not exist that meets your requirements (has the appropriate values) you can create one. See *“Validations”* on page 243 for a complete list of Validations that are delivered during a Kintana installation.

You can also select a Validation that has been configured for use at your site.



Be careful when using a Validation that has been configured for use in another process. If the owner of the other process changes the Validation, it will also be changed for the items in your process. Consider creating a new Validation by copying the existing one. You can then control who can alter the Validation values by setting Ownership on that Validation.

## Building a Validation

If a Validation does not exist that meets your requirements (has the appropriate values) you can create a new one. [Table 7-8](#) provides a few examples of when to use specific types of validations. See “[Validations](#)” on page 243 for instructions on creating the Validation.

*Table 7-8. Field/Validation Examples*

Field/Validation	Possible Uses
Lists (Drop down or auto-complete)	<ul style="list-style-type: none"> <li>List of all users</li> <li>List of all users in a specific security group</li> <li>Desired actions</li> <li>List of information located in another (non-Kintana) system. (example: list of managers stored in PeopleSoft)</li> </ul>
Descriptive parameters (Radio buttons or check boxes)	<ul style="list-style-type: none"> <li>Question specifying an action. (Example: File needs to be compiled?)</li> <li>Results of an execution step (radio button). The result can be used later in a processing decision.</li> </ul>
Information used for reporting	<ul style="list-style-type: none"> <li>Description of the change (text field).</li> <li>Release version number</li> </ul>

## Tips for Configuring Validations

Consider the following tips when creating a Validation for your Request Type:

- Be careful when creating Validations (drop down lists and auto-complete lists) that are validated by lists. Each time the set of values changes, you

will be forced to update the Validation. Consider, instead, validating using a SQL query or PL/SQL function. For example, to create an auto-complete field that lists all Kintana users in a specific department, validate the list by SQL.

```
SELECT U.USER_ID, U.USERNAME, U.FIRST_NAME, U.LAST_NAME
FROM KNTA_USERS U, KNTA_SECURITY_GROUPS SG,
     KNTA_USER_SECURITY US
WHERE SG.SECURITY_GROUP_ID = US.SECURITY_GROUP_ID AND
      US.USER_ID = U.USER_ID
AND SG.SECURITY_GROUP_NAME = 'Support Team'
and UPPER(u.username) like UPPER('?%')
and (u.username like upper(substr('?',1,1)) || '%')
    or u.username like lower(substr('?',1,1)) || '%')
order by 2
```

In the above example, when a new user is added to Kintana and included in the “Support Team” Security Group, that user will automatically be included in the auto-complete list.

- Reuse SQL and PL/SQL from existing Validations. Review Kintana’s seeded Validations (see “[Validations](#)” on page 243) to see if there are other similar Validations in the system. If there are, copy the validation and modify the VALIDATED BY specifications to meet your requirements.

## Configuring Field Behavior - Overview

Field behavior and properties can be configured in a number of different ways:

- [Visibility](#)
- [Editability](#)
- [Defaulting](#)
- [Required/Reconfirm](#)
- [Notifications](#)

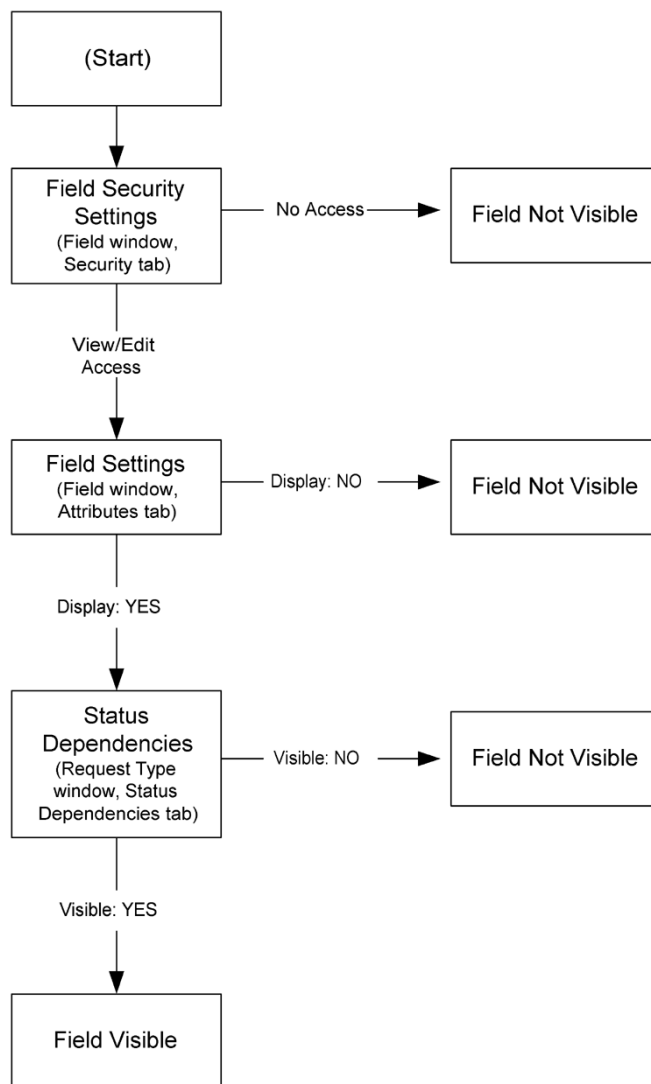
### Visibility

Fields can be set to be visible or hidden to the user based on the following criteria:

- 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. For more detailed information, see “[Creating a Field](#)” on page 126.

- 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. For more information, see *“Configuring Field Behavior Using Status Dependencies”* on page 146.
- 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. For more detailed information, see *“Creating a Field”* on page 126.

*Figure 7-1* illustrates the hierarchy that determines whether a field is visible to a particular user.



*Figure 7-1 Field Visibility Interactions*

For details on configuring each of these features, see [Table 7-9](#).

*Table 7-9. Quick Reference Guide - Field Visibility Parameters*

<b>Field Parameter</b>	<b>Location</b>	<b>Section &amp; Page</b>
Field Security Settings	REQUEST TYPE window -> <b>FIELDS</b> tab -> FIELD window -> <b>SECURITY</b> tab	<a href="#">“Creating a Field”</a> on page 126
Field Settings	REQUEST TYPE window -> <b>FIELDS</b> tab -> FIELD window -> <b>ATTRIBUTES</b> tab	<a href="#">“Request Type Field window - Attributes Tab”</a> on page 128
Status Dependencies	REQUEST TYPE window -> <b>STATUS DEPENDENCIES</b> tab	<a href="#">“Status Dependencies - Visible”</a> on page 151

## Editability

Fields can be set to become display-only, so that their contents are frozen and they become non-editable, based on the following criteria:

- Request Status — Based on the Status of the Request itself, the field can be set to become non-editable. For more information, see [“Configuring Field Behavior Using Status Dependencies”](#) on page 146.
- Field security — Fields can also be configured to be visible but non-editable to particular users or Security Groups. This is controlled from the FIELD window’s **SECURITY** tab. For more detailed information, see [“Creating a Field”](#) on page 126.

## Defaulting

Fields can be configured to populate themselves automatically based on the following criteria:

- 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. For more detailed information, see [“Creating a Field”](#) on page 126.



- Request Type Rules — A Request Type can be configured to automatically populate several fields at once 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** fields can be automatically filled. This is controlled from the **REQUEST TYPE** window's **RULES** tab. For more detailed information, see [“Configuring Request Type Defaulting Behavior \(Rules\)”](#) on page 135.
- Request Type Commands — Kintana Commands can also be used to control certain behavior of Request Type fields. At specific points (Workflow execution steps) in your resolution process, you can select to run the commands stored in the Request Type. These commands can then manipulate the data inside a Request Type field. For example, you can construct a Command to 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 on the same Request Type.

Controlling field values using Commands can be useful in the following situations (examples):

- o Store a value from an execution (Note: this can also be done using Workflow parameters. See [“Transition based on a previous workflow step result \(parameters\)”](#) on page 103 for details.)
- o Clearing a field after evaluating a number of parameters.

See the [“Using Commands and Tokens”](#) for more information on setting up commands to control field defaulting.

## Required/Reconfirm

Fields can be configured to be required or need to be reconfirmed, or even cleared entirely, based on the Status of the Request. For more information, see [“Configuring Field Behavior Using Status Dependencies”](#) on page 146.

## Notifications

Fields can be configured to send an email Notification based on a change in value. For more information, see [“Setting Notifications on Request Field Changes”](#) on page 202.

# Creating a Field

New fields are created and configured using the FIELD window, accessed from the REQUEST TYPE window's **FIELDS** tab.

From the FIELD window you can 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



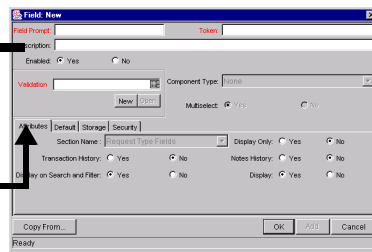
Note

Because field behavior is often dependent on other fields in the Request Type, you often have to create the other Request Type fields before configuring a field's behavior.

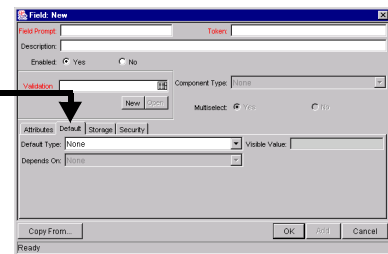
### General Information

**Region:** used to enter basic field parameters

**Attributes tab:** used to set basic display, edit and requirement field properties.

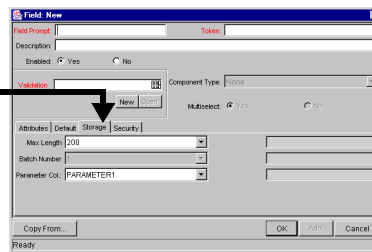


**Default tab:** used to set the value in the field.



### Storage tab:

used to set properties of the field relating to its storage in the database.



### Security tab:

used to set users who can view and edit this field.

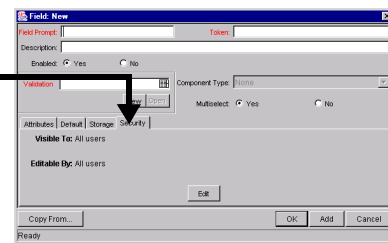
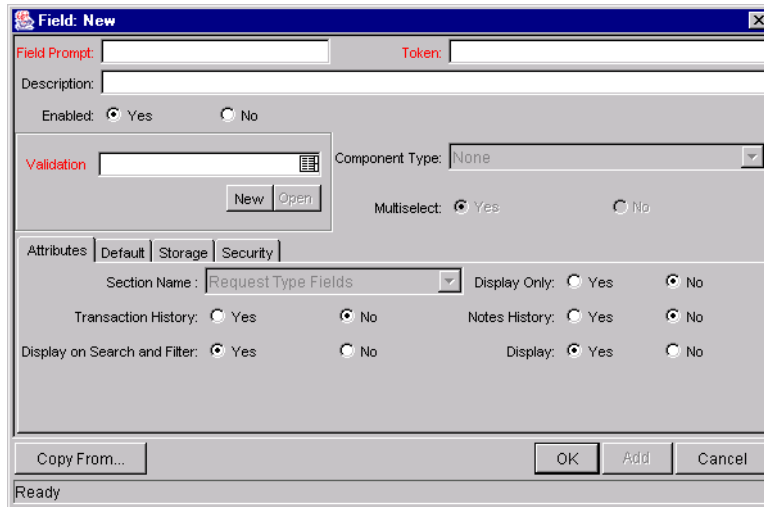


Figure 7-2 Request Type Field Window

To generate a new field:

1. Click **NEW** in the **FIELDS** tab.



2. Enter the general information for the Request field. See [Table 7-10](#) for a definition of the fields in this area.

Table 7-10. Request Type Field window - General Information Region

Field	Description
FIELD PROMPT	The prompt visible for the Request Type field on 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	Determines 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, etc. See <a href="#">“Determining the Field Type (Selecting a Validation)”</a> on page 118 for more details.
COMPONENT TYPE	Defines the visual characteristics of the field (drop down list, free form text field, etc.). This is derived from the Validation chosen. This field cannot be edited.
MULTISELECT	Determines whether or not the field allows users to select more than one entry. Only valid for fields with an auto-complete component for the Validation.

- Click the **ATTRIBUTES** tab to define the field's basic properties (DISPLAY ONLY, TRANSACTION HISTORY, NOTES HISTORY, etc.). See [Table 7-11](#) for a definition of the fields in this area.

*Table 7-11. Request Type Field window - Attributes Tab*

Field	Description
SECTION NAME	The area of the Request on which the field is displayed.
DISPLAY ONLY	Determines 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	Determines whether or not the field will be displayed in Search and Filter pages in the Kintana interface.
DISPLAY	Determines whether or not the field is seen by Requests that use the given Request Type. If set to <b>No</b> , the Request Type field will no longer be displayed.

- Click the **DEFAULTS** tab to define the default value for that field. See [Table 7-12](#) for a definition of the fields in this area.

*Table 7-12. Request Type Field window - Defaults Tab*

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 <b>CONSTANT</b> 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.

5. Click the **STORAGE** tab to view the field's location in the database. See [Table 7-13](#) for a definition of the fields in this area.

Table 7-13. Request Type Field window - Storage Tab

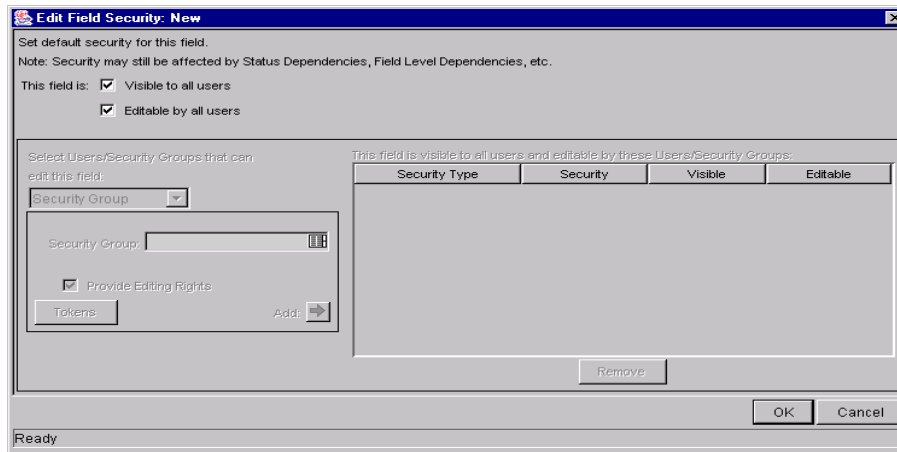
Field	Description
MAX LENGTH	Determines the maximum field character length. The two possible values are 200 and 1800.
BATCH NUMBER	Based on the number of maximum fields. For every 50 fields, one batch is created. 10 of these 50 fields can be more than 200 characters in length. Enabled only when there are more than 50 fields (creating more than one batch).
PARAMETER COL	Determines 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 Types, allowing up to 50 fields/batch. No two fields in a Request Type can use the same column number within the same batch.

6. Click the **SECURITY** tab to view the field's security configuration. See [Table 7-14](#) for a definition of the fields in this area.

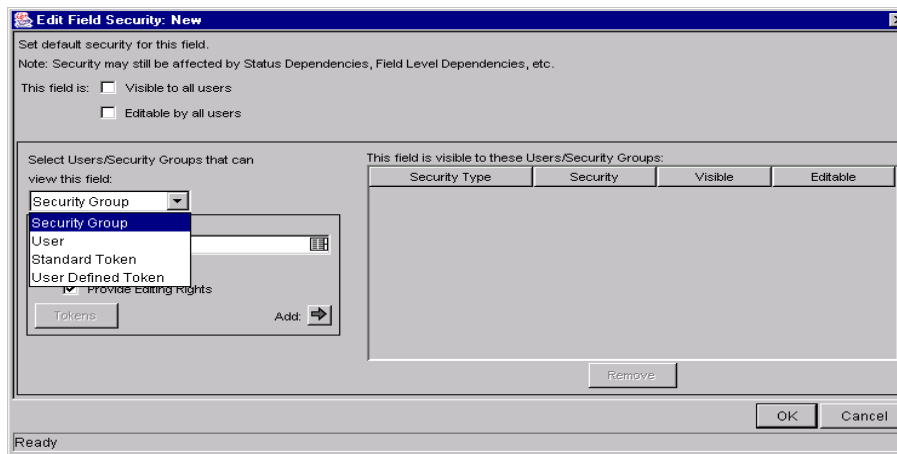
Table 7-14. Request Type Field window - Security Tab

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.
<b>EDIT</b>	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 <a href="#">“Creating a Field”</a> on page 126 for more detailed information.

7. Click **EDIT**. The EDIT FIELD SECURITY window opens.



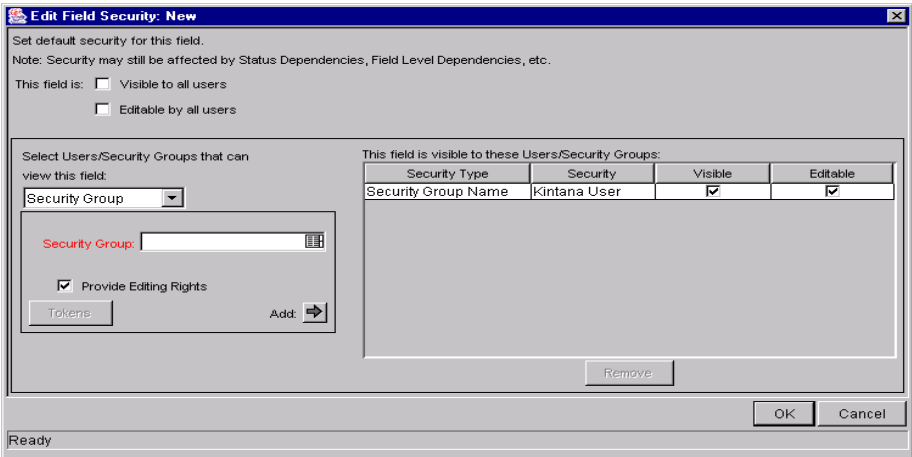
8. Uncheck the VISIBLE TO ALL users box to begin fine-tuning field properties.
9. Make a choice from the SELECT USERS/SECURITY GROUPS THAT CAN VIEW THIS FIELD drop down list. You can select a **USER**, **SECURITY GROUP**, **STANDARD TOKEN**, or **USER DEFINED TOKEN**.



10. Once you have made your choice from the drop down list, select the User, Security Group, or Token from the auto-complete list.

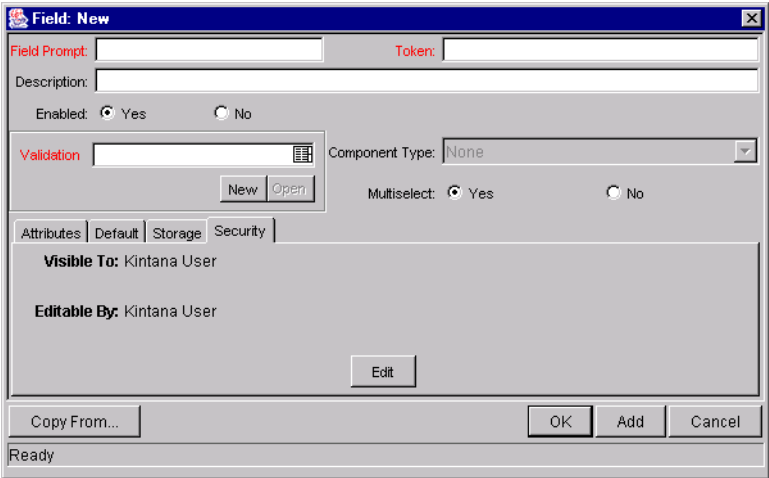
If you wish to assign the selected User, Security Group, or Token editing rights as well as viewing rights to the field, check the PROVIDE EDITING RIGHTS box.

11. 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.



You can change the VISIBLE and EDITABLE settings for each entry directly in the EDIT FIELD SECURITY window. Uncheck 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.

- 12. When you are finished adding Users, Security Groups, or Tokens to the THIS FIELD IS VISIBLE TO THESE USERS/SECURITY GROUPS area, click **OK** to save changes and return to the **SECURITY** tab. The **SECURITY** tab is updated with the list of Users, Security Groups, or Tokens with viewing or editing rights to the field.





Note

When you add field-level security to existing fields on a Request Type that has been used to create Requests, tables in the Kintana database are updated to handle this new configuration. Because of the scope of database changes, you should re-run the Database Statistics on your Kintana Database. Instructions for this are included in the Kintana System Administration Guide. Contact your System Administrator for help with this procedure.



Note

There can only be 500 rows per column, 3 columns per tab, and a maximum of 20 tabs for each Request Type.

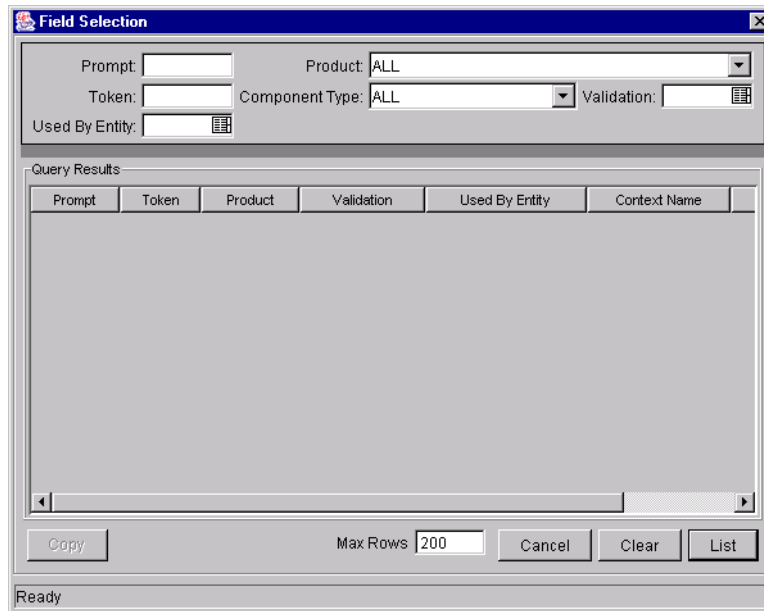
When taking advantage of the Reporting Meta Layer functionality, those fields contained within the first four batches (200 fields) will be available for reporting.

## Copying a Request Type Field

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 field

1. Open the Request Type.
2. Click **NEW** in the **FIELD** Tab. The **FIELD** window opens.
3. Click **COPY FROM**. The **FIELD SELECTION** window opens.





4. Fields can then be queried by a number of criteria, 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 Kintana fields, you should limit the list of fields by one or more of the query criteria.
5. Once a list of fields matching the selection criteria is obtained, highlight the desired field, and click **COPY**. This closes the window and copies the definition of the selected field into the `NEW FIELD` window.
6. Make any necessary modifications.
7. Click **OK**.

This saves the changes and closes the `FIELD` window.

## Removing Request Type Fields

To remove a field permanently from a Request Type:

1. Open the Request Type.
2. Select the field in the **FIELDS** tab.
3. Click **REMOVE**.
4. Click **OK**.

This removes the field and closes the window.

## Setting the Number of Maximum Fields for a Request Type

Initially, Request Types have a set group of selectable, maximum field values. The values are in increments of 50 and stop at 300. To change the number of maximum allowable fields for all Request Types:

1. Click the **CONFIGURATION** screen group and the **VALIDATIONS** screen. The **VALIDATION WORKBENCH** opens.
2. Open the **CRT- MAX CUSTOM FIELDS** Validation.

Seq	Code	Meaning	Description	Enabled	Default
1	50	50		Y	Y
2	100	100		Y	N
3	150	150		Y	N
4	200	200		Y	N
5	250	250		Y	N
6	300	300		Y	N

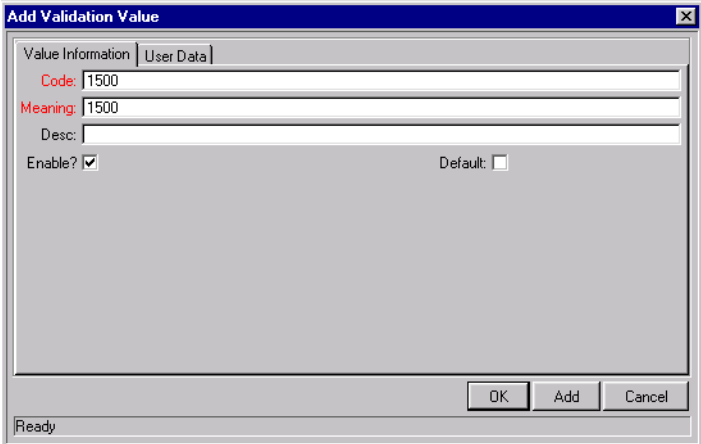
3. Click **NEW** to open the **ADD VALIDATION VALUE** window. Enter the new maximum field value in both the **CODE** and **MEANING** fields (the values entered must be the same).



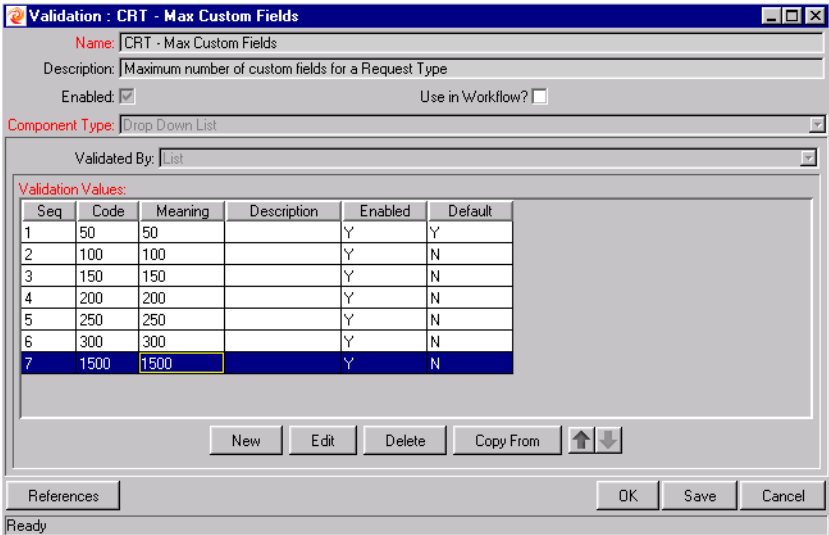
Note

To change an existing Validation Value, select the desired line and click **EDIT**. Enter the new values and then click **OK**.

All values entered must be multiples of 50 (i.e. 350, 1500, 1550, etc.)



4. Click **OK**.



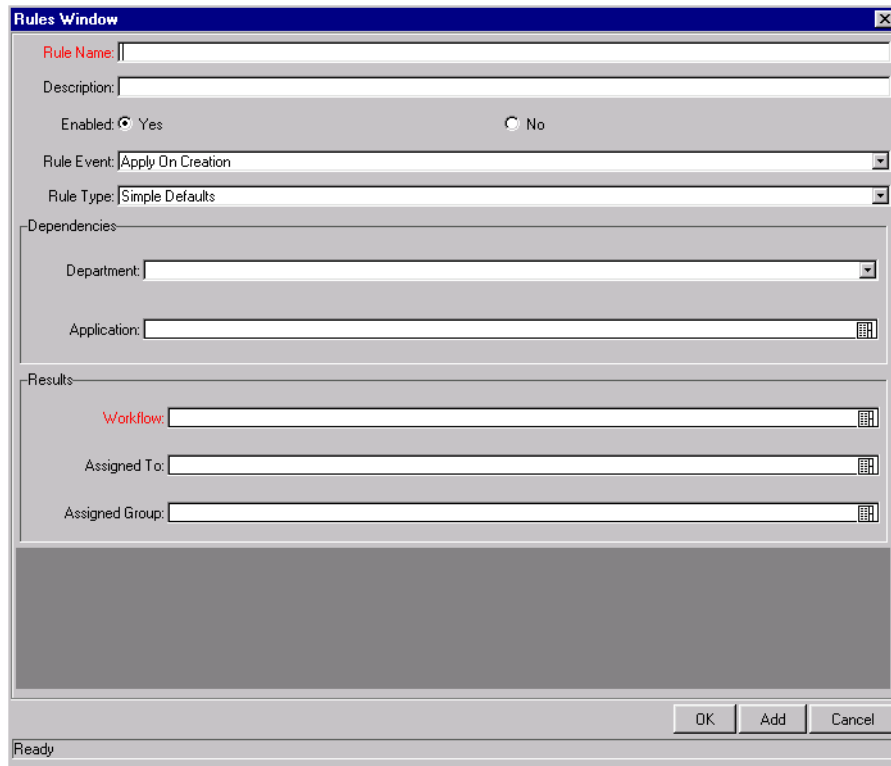
The maximum custom fields value is now one of the selectable value options from the MAX FIELDS drop down list on the REQUEST TYPE window.

## Configuring Request Type Defaulting Behavior (Rules)

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

- A default WORKFLOW, ASSIGNED TO user or ASSIGNED GROUP should be specified when a Request of this Type is initially created.
- Multiple Request fields should be populated depending on the value of a single field.

The RULES window, opened from the REQUEST TYPE window's **RULES** tab, configures Request Rules.



The screenshot shows the 'Rules Window' dialog box. It has a title bar with 'Rules Window' and a close button. The main area contains several fields and controls:

- Rule Name:** A text input field.
- Description:** A text input field.
- Enabled:** Radio buttons for 'Yes' (selected) and 'No'.
- Rule Event:** A dropdown menu with 'Apply On Creation' selected.
- Rule Type:** A dropdown menu with 'Simple Defaults' selected.
- Dependencies:**
  - Department:** A dropdown menu.
  - Application:** A text input field with a list icon.
- Results:**
  - Workflow:** A text input field with a list icon.
  - Assigned To:** A text input field with a list icon.
  - Assigned Group:** A text input field with a list icon.

At the bottom right, there are three buttons: 'OK', 'Add', and 'Cancel'. The status bar at the bottom left shows 'Ready'.

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**.



Note

When switching between Rule Types, whatever work has been done in the first Type will be lost when the switch is made.

The following sections discuss setting up Request Rules in more detail:

- [Configuring Simple Default Rules](#)
- [Configuring Advanced Default Rules](#)

## Configuring Simple Default Rules

Simple Default Rules are used to auto-fill the WORKFLOW, ASSIGNED TO and ASSIGNED GROUP fields.

The screenshot shows the 'Rules Window' dialog box. It includes the following fields and controls:

- Rule Name:** Text input field.
- Description:** Text input field.
- Enabled:** Radio buttons for 'Yes' (selected) and 'No'.
- Rule Event:** Dropdown menu set to 'Apply On Creation'.
- Rule Type:** Dropdown menu set to 'Simple Defaults'.
- Dependencies:**
  - Department:** Dropdown menu.
  - Application:** Text input field with a grid icon.
- Results:**
  - Workflow:** Text input field with a red label and a grid icon.
  - Assigned To:** Text input field with a grid icon.
  - Assigned Group:** Text input field with a grid icon.
- Buttons:** 'OK', 'Add', and 'Cancel' at the bottom right.
- Status:** 'Ready' at the bottom left.

These fields can be filled based on the RULE EVENT and DEPENDENCIES fields, discussed in [Table 7-15](#).

Table 7-15. Simple Default Rule Control Fields

Field	Description
RULE EVENT	<p>Specifies the event that triggers the Rule.</p> <p><b>APPLY ON CREATION</b> - The Rule will fire when the Request is created, filling in whichever of the RESULTS fields have been specified.</p> <p><b>APPLY ON FIELD CHANGE</b> - The Rule will fire when one of the DEPENDENCIES fields is changed to the specified value.</p> <p><b>APPLY ON FIELD CHANGE AND STOP PROCESSING RULES</b> - 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.</p>
Department	Specifies the department that triggers the Rule.
Application	Specifies the application that triggers the Rule.

Using any appropriate combination of these control fields, the WORKFLOW, ASSIGNED TO, or ASSIGNED GROUP fields can be specified.



Note

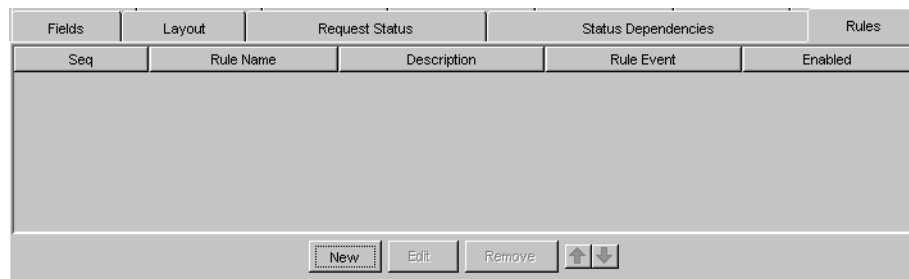
The WORKFLOW field is the only required field for Simple Default Rules.

By setting the desired WORKFLOW and the RULE EVENT to **APPLY ON CREATION**, you can default the Workflow you want every time a Request of this Type is created.

### Creating a Simple Default Rule

To add a new Simple Default Rule to a Request Type:

1. Open the REQUEST TYPE window for the desired Request Type and click the **RULES** tab.



2. Click **NEW**. The REQUEST TYPE RULES window opens in **SIMPLE DEFAULTS** mode.

The screenshot shows a 'Rules Window' dialog box with the following fields and controls:

- Rule Name:** Text input field.
- Description:** Text input field.
- Enabled:** Radio buttons for 'Yes' (selected) and 'No'.
- Rule Event:** Dropdown menu with 'Apply On Creation' selected.
- Rule Type:** Dropdown menu with 'Simple Defaults' selected.
- Dependencies:**
  - Department:** Dropdown menu.
  - Application:** Text input field with a calendar icon.
- Results:**
  - Workflow:** Text input field with a calendar icon.
  - Assigned To:** Text input field with a calendar icon.
  - Assigned Group:** Text input field with a calendar icon.

Buttons at the bottom right: OK, Add, Cancel. Status bar at the bottom left: Ready.

3. Enter the required **RULE NAME** and **WORKFLOW**.
4. Enter any desired **DEPENDENCIES**. These **DEPENDENCIES** must be met in order for the rule to be executed.
5. Enter the **RESULTS** for this rule. A **WORKFLOW** must be entered, but all other fields are optional.
6. Click **OK** to save this rule and close the window, or click **ADD** to add this default rule and clear the window to add another rule.

Note

The fields displayed in the **RULES** window shown are from the Request Header Type.

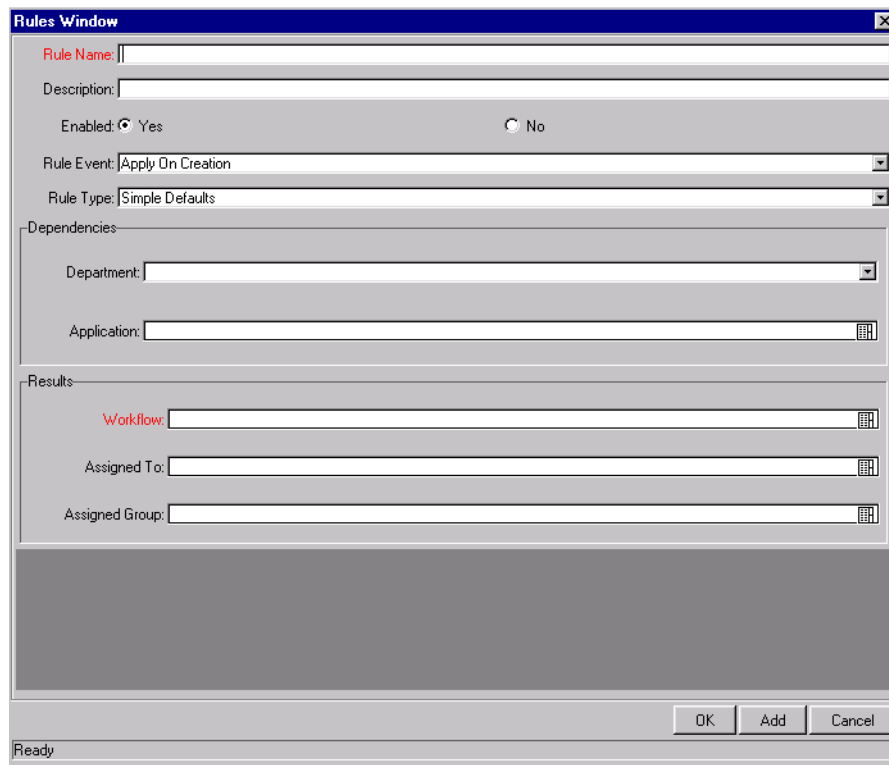
Once this rule is saved, any new Request matching the combination of **REQUEST TYPE**, **DEPARTMENT**, and **APPLICATION** for the rule automatically updates the **WORKFLOW**, **ASSIGNED TO**, and **ASSIGNED GROUP** fields to the default values specified in the rule.

If more than one rule applies for a given Request, then Kintana uses the most specific rule. See “[Configuring Advanced Default Rules](#)” on page 141 for more detailed information.

### *Example: ACME Defaults Software Change Workflow*

ACME is creating their Financial Software Change Request Type, and have decided that it would be convenient for the WORKFLOW field to be populated automatically with the value **FINANCIAL SOFTWARE CHANGE WORKFLOW** whenever a Request of this type is created.

1. Harold Lomax, ACME’s IT Configuration Manager, opens the Financial Software Change Request Type and clicks the **RULES** tab.
2. Lomax clicks **NEW**. The RULES window opens in **SIMPLE DEFAULTS** mode.



The screenshot shows the 'Rules Window' dialog box. It has a title bar with 'Rules Window' and a close button. The main area contains several fields and controls:

- Rule Name:** A text input field.
- Description:** A text input field.
- Enabled:** Radio buttons for 'Yes' (selected) and 'No'.
- Rule Event:** A dropdown menu with 'Apply On Creation' selected.
- Rule Type:** A dropdown menu with 'Simple Defaults' selected.
- Dependencies:** A section with two fields: 'Department:' (dropdown) and 'Application:' (text input with a grid icon).
- Results:** A section with three fields: 'Workflow:' (text input with a grid icon), 'Assigned To:' (text input with a grid icon), and 'Assigned Group:' (text input with a grid icon).

At the bottom right, there are three buttons: 'OK', 'Add', and 'Cancel'. The status bar at the bottom left shows 'Ready'.

3. Lomax enters a Rule Name and specifies Financial Software Change Workflow in the Workflow field.



4. Lomax clicks **OK**. The new Rule is added to the **RULES** tab.

Fields		Layout		Request Status		Status Dependencies		Rules	
Seq	Rule Name	Description	Rule Event	Enabled					
1	Financial System Workfl...		Apply On Creation	Y					

## Configuring Advanced Default Rules

Advanced Default Rules differ from Simple Default Rules in the following ways:

- Simple Default Rules can only trigger from Request creation or changes to the DEPARTMENT or APPLICATION fields. Advanced Default Rules can trigger from changes to any field in the Request.
- Simple Default Rules can only populate the WORKFLOW, ASSIGNED TO, or ASSIGNED GROUP fields. Advanced Default Rules can populate any field or

set of fields in the Request simultaneously, including fields in the Request or in the Request Header.



Note

Configuring Advanced Default Rules requires knowledge of SQL.

Field Name	Value
------------	-------

Field Name	Column	Token
------------	--------	-------

Advanced Default Rules are where the options besides **APPLY ON CREATION** in the **RULE EVENT** field become more useful.

- **APPLY ON FIELD CHANGE** — The rule applies when values in other fields change. This functions two ways:
  - a. **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.
  - b. **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 will be automatically populated according to rule order. This is useful in the event of multiple Dependency field matches.



A Kintana configuration expert specifies Rule One with Dependency Field A = **123** and Rule Two with Dependency Field B = **XYZ**.

A user fills in Field B with **XYZ**, then fills in Field A with **123**.

Kintana executes Rule One first, followed by Rule Two.

- **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.



A Kintana user specifies Rule One with Dependency Field A = **123** and Field B = **XYZ**, and Rule Two with Dependency Field B = **XYZ**.

A user fills in Field A with **123**, then fills in Field B with **XYZ**.

Kintana executes Rule One only and stops.

## *Creating an Advanced Default Rule*

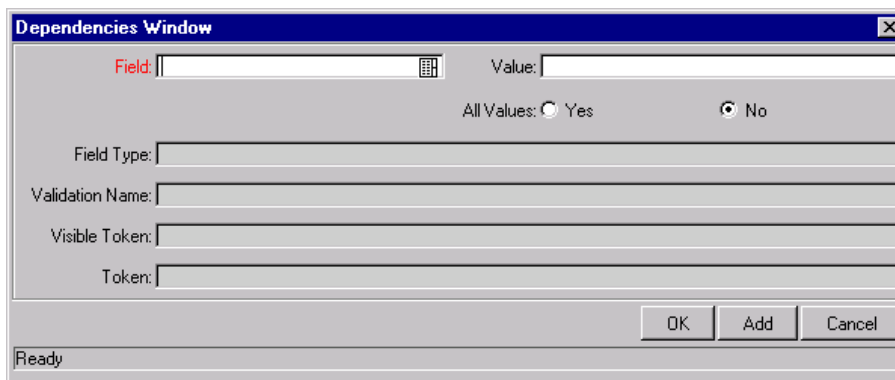
To create an Advanced Default Rule to set up automatic population of fields for a Request Type:

1. Open the **REQUEST TYPE** window for the desired Request Type and click the **RULES** tab.
2. Click **NEW**. The **RULES** window opens in **SIMPLE DEFAULTS** mode.
3. From the **RULE TYPE** drop down list, select **ADVANCED DEFAULTS**.

The screenshot shows a 'Rules Window' dialog box with the following fields and controls:

- Rule Name:** A text input field.
- Description:** A text input field.
- Enabled:** Radio buttons for 'Yes' (selected) and 'No'.
- Rule Event:** A dropdown menu currently showing 'Apply On Creation'.
- Rule Type:** A dropdown menu currently showing 'Advanced Defaults'.
- Dependencies:** A table with columns 'Field Name' and 'Value'. Below the table are 'New', 'Edit', and 'Remove' buttons.
- Results:** A table with columns 'Field Name', 'Column', and 'Token'. Below the table are 'New' and 'Remove' buttons.
- SQL:** A large empty text area.
- Buttons:** 'OK', 'Add', and 'Cancel' buttons at the bottom right.
- Status:** 'Ready' at the bottom left.

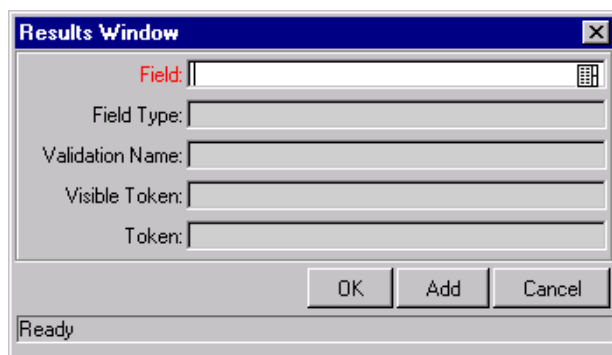
4. Enter a name for the new rule in the **RULE NAME** field. A description is optional.
5. Decide whether or not the rule is enabled by selecting **YES** or **NO** next to **ENABLED**.
6. Select an event that will trigger the auto-population from the **RULE EVENT** drop down list (**APPLY ON CREATION, APPLY ON FIELD CHANGE, APPLY ON FIELD CHANGE AND STOP PROCESSING OTHER RULES**)
7. Select a field or fields to trigger the rule by clicking **NEW** in the **DEPENDENCIES** area. The **DEPENDENCIES** window opens.



Note

Request Default Rules cannot be configured to trigger from a multi-select auto-complete field. Do not choose a multi-select auto-complete for the FIELD.

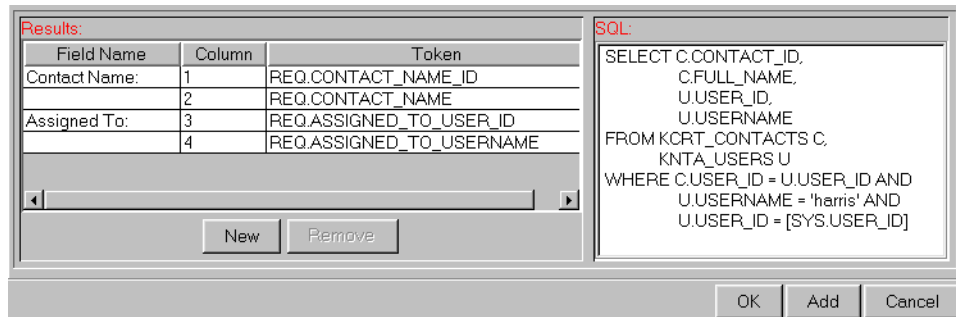
8. Select the field and value that you want to trigger the auto-population. To make all values for the field trigger the auto-population, select **YES** next to ALL VALUES.
9. Click **ADD** to add the field to the DEPENDENCIES area and continue to add more without closing the DEPENDENCIES window. Click **OK** to add the field and close the DEPENDENCIES window.
10. Select a field or fields for the rule to auto-populate by clicking **NEW** in the RESULTS area. The RESULTS window opens.



11. Specify the field you want to be populated in the FIELD auto-complete list.
12. Click **ADD** to add the field to the RESULTS area and continue to add more without closing the RESULTS window. Click **OK** to add the field and close the RESULTS window.

The RESULTS area's table displays the FIELD name, its column number, and token.

- 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 SQL statement shown above will load C.CONTACT\_ID into column 1, C.FULL\_NAME into column 2, U.USER\_ID into column 3, and U.USERNAME into column 4.

- Click **APPLY** to apply the rule and continue to create more, or click **OK** to apply the rule and close the RULES window.

Kintana will validate 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. See *"Tokens"* on page 305 for more detailed information.

- Click **SAVE** in the **RULES** tab to save any changes.

## Configuring Field Behavior Using Status Dependencies

During a Request's resolution process, it 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.

The following sections discuss setting up field behavior using Status dependencies in more detail:

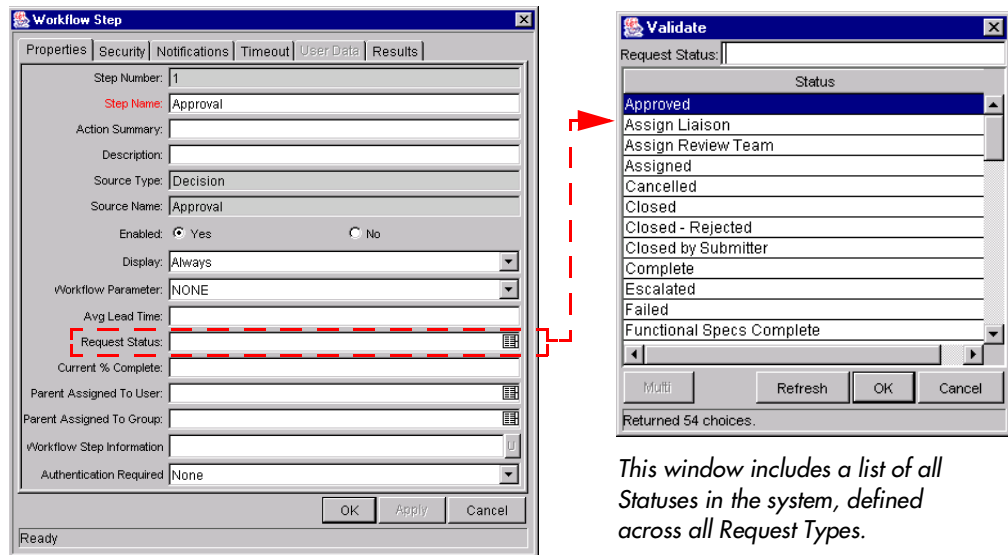
- [Creating Your Request Statuses](#)
- [Configuring Field Status Dependency Behavior](#)
- [Assigning Request Statuses to Workflow Steps](#)

## Creating Your 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 can be linked to Workflow Steps.



*This window includes a list of all Statuses in the system, defined across all Request Types.*

As a Request is processed 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 Field Status Dependency Behavior”](#) on page 150.

Before linking Request Statuses to Workflow Steps, the Request Type must first possess all desired Statuses. The list of possible Statuses the Request can take on is created in the REQUEST TYPE window's **REQUEST STATUS** tab.

Fields    Layout    Request Status    Status Dependencies    Rules

Available Request Statuses:

- Approved
- Assign Review Team
- Escalated
- Failed
- In Review
- On Hold
- Open

Linked Request Statuses:

- Assign Liaison
- Assigned
- Cancelled
- Closed
- Closed - Rejected
- Closed by Submitter
- Complete

Request Status: ...    Initial Request Status: Not Submitted

Status Name	Enabled	Auto Link
Approved	Y	N
Assign Liaison	Y	Y
Assign Review Team	Y	N
Assigned	Y	Y
Cancelled	Y	Y
Closed	Y	Y
Closed - Rejected	Y	Y
Closed by Submitter	Y	Y
Complete	Y	Y
Escalated	Y	N
Failed	Y	N
Functional Steps Complete	Y	Y

New    Edit    Delete    Refresh    Close

54 Request Status Records Loaded

If a desired Status does not appear in the AVAILABLE REQUEST STATUSES list, it can be created.

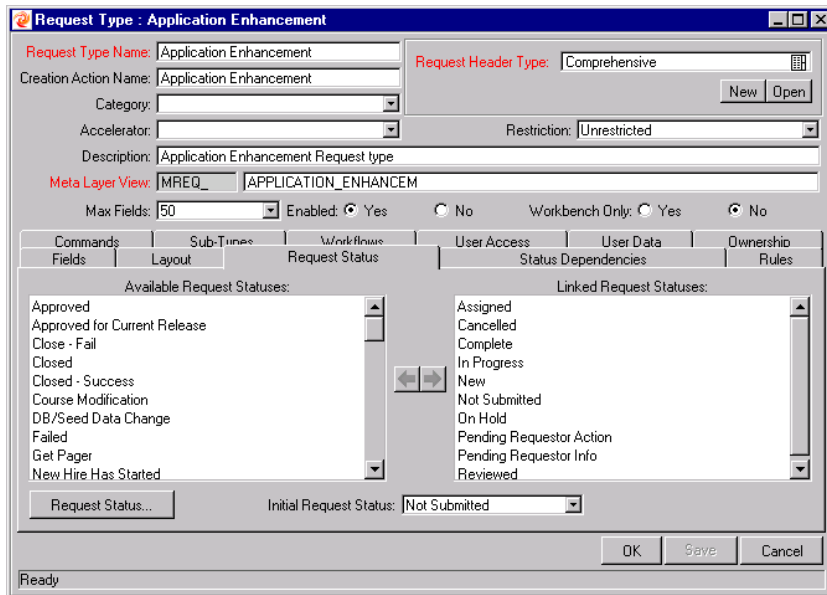
The Request's initial Status can be set using the INITIAL REQUEST STATUS drop down list.

### *Adding and Linking a Request Status*

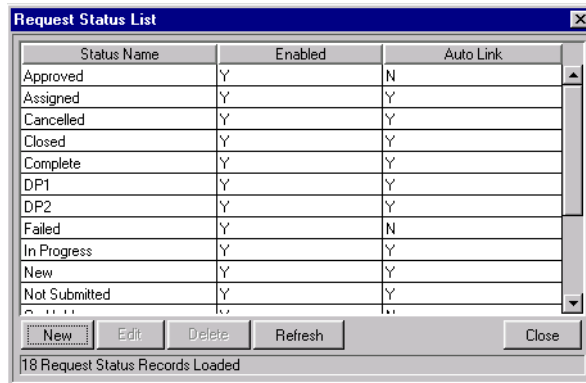
To add a new Request Status to the list of available Request Statuses:

1. Open the REQUEST TYPE window and click the **REQUEST STATUS** tab.

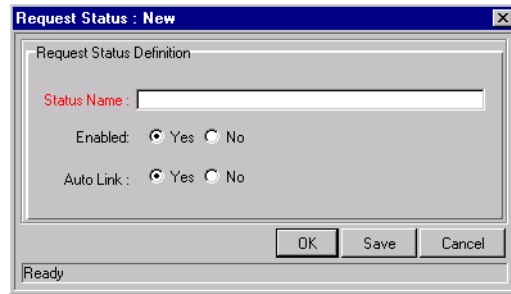




2. Click **REQUEST STATUS**. The REQUEST STATUS LIST opens.



3. Click **NEW**. The REQUEST STATUS: NEW window opens.

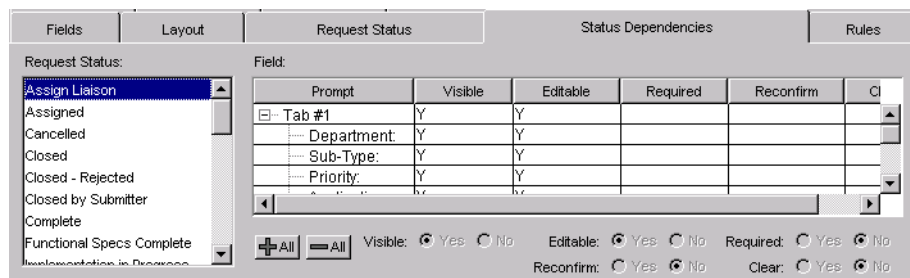


4. Enter a STATUS NAME.
5. Select ENABLED = **YES** for the status to appear in the AVAILABLE REQUEST STATUS column for all new Request Types.
6. Select AUTO LINK = **YES** for the status to automatically link to all new Request Types.
7. Click **OK** to save the information and close the REQUEST STATUS: NEW window, **SAVE** to save the information and leave the window open, or **CANCEL** to exit the window without saving the changes.

Request Statuses can also be edited and deleted from the REQUEST STATUS LIST. The NAME, ENABLED, and AUTO LINK fields can be changed when a Status from this list is edited.

## Configuring Field Status Dependency Behavior

Certain aspects of Request field behavior can be linked directly to the Statuses the Request can take on. This is done in the **STATUS DEPENDENCIES** tab of the REQUEST TYPE window.



To assign field properties based on Request Status:

1. Select a Request Status from the REQUEST STATUS list.
2. Select a field in the FIELD table.
3. Assign the field's attributes under the given Request Status. This is done by toggling the radio buttons at the bottom of the screen.

Prompt	Visible	Editable	Required	Reconfirm	Clear
Application:	<input checked="" type="radio"/> Y	<input checked="" type="radio"/> Y			
Contact Name:	<input checked="" type="radio"/> Y	<input checked="" type="radio"/> Y			
Assigned To:	<input checked="" type="radio"/> Y	<input checked="" type="radio"/> Y			
Assigned Group:	<input checked="" type="radio"/> Y	<input checked="" type="radio"/> Y			

+ All   = All   Visible:  Yes  No   Editable:  Yes  No   Required:  Yes  No  
 Reconfirm:  Yes  No   Clear:  Yes  No

Multiple fields can be configured simultaneously by using the Ctrl or Shift buttons to select the fields and then change the attribute values. Selecting a tab row, such as **HEADER FIELDS**, will allow you 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.

The following sections discuss each field attribute in more detail:

- [Status Dependencies - Visible](#)
- [Status Dependencies - Required Field](#)
- [Status Dependencies - Updateable Field](#)
- [Status Dependencies - Reconfirm Field](#)
- [Status Dependencies - Clear Field](#)
- [Status Dependencies Interactions](#)

### *Status Dependencies - Visible*

The **VISIBLE** radio button 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.

### *Status Dependencies - Required Field*

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.



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**

### *Status Dependencies - Updateable Field*

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 is non-updateable due to this logic, the field is grayed out in the Request. The value is visible but cannot be changed.

### *Status Dependencies - Reconfirm Field*

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.

### *Status Dependencies - Clear Field*

The CLEAR flag 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 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.

### Status Dependencies Interactions

*Table 7-16* illustrates the results of different combinations of the REQUIRED, RECONFIRM, and CLEAR functions.

*Table 7-16. Status Dependencies Interactions*

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



Note

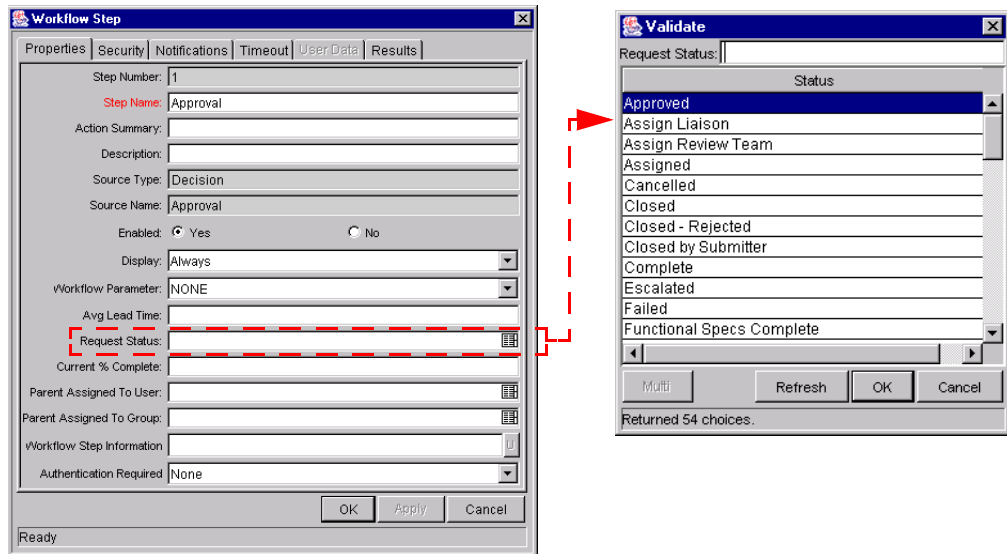
For each Request Status within a Request Type, there can be up to a maximum of 250 fields with a REQUIRED state and 250 fields with a RECONFIRM state.

## Assigning Request Statuses to Workflow Steps

The final step in linking Request field logic to a Workflow is to assign the appropriate Request Statuses to their respective Workflow Steps.

To assign a Request Status to a Workflow Step:

1. Open the WORKFLOW window.
2. Click the LAYOUT tab.
3. Double-click on the desired Workflow Step.
4. Select the desired Request Status from the REQUEST STATUS auto-complete list.



5. Repeat as needed with all necessary Workflow Steps.
6. Save the Workflow.



Note

Not all Workflow Steps need to have a Request Status assigned. A Request retains the last-encountered Status.

As the Request progresses through this Workflow, it will take on the Status assigned in each Workflow Step.

## Modifying the Request Type Layout

Modifying the layout of a Request Type can consist of the following activities:

- [Modifying the Request Type Field Width](#)
- [Moving Fields in a Request Type](#)
- [Adding Sections to the Request Type](#)

## Modifying the Request Type Field Width

To change the column width of a field:

1. Open the Request Type.
2. Click the **LAYOUT** tab.
3. Select the section of the Request that contains the field.
4. Select the field.
5. Select a width of **1**, **2**, or **3** in the FIELD WIDTH drop down list.



Note

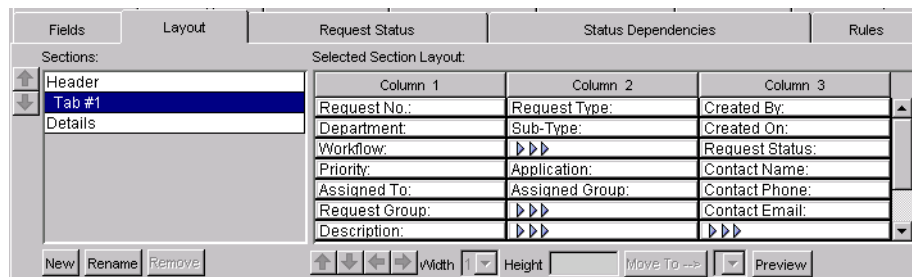
The field width has to correspond to the column location. For example, a field located in column two cannot have a width set to **3**.

Additionally, for fields of component type Text Area, it is possible to determine the number of lines the Text Area will display. Select the 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.

## Moving Fields in a Request Type

To move a field or a set of fields:

1. Open the Request Type.
2. Click the **LAYOUT** tab.



3. Select the section of the Request that contains the field(s) you want to move.
4. Select the field(s). To select more than one field, press the Shift key while selecting the last field in the set. Selection is either singular or a sequential group. Only groups of adjacent fields can be selected. Blank cells can be selected as part of the group.
5. Move the fields to the desired location in the layout builder, either by clicking the arrow buttons or using the corresponding keyboard arrow keys.
6. 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, click on the field and then click **MOVE TO -->**. If there is more than one section, select the desired section from the **MOVE TO -->** button drop down list first.

## Adding Sections to the Request Type

The layout for a new Request Type defaults to have only one section for the custom fields.

The screenshot shows the 'Layout' tab of the configuration interface. On the left, there is a 'Sections' list with 'Header', 'Tab #1', and 'Details'. 'Tab #1' is selected. The main area is titled 'Selected Section Layout:' and contains a table with three columns: 'Column 1', 'Column 2', and 'Column 3'. The table contains the following fields:

Column 1	Column 2	Column 3
Request No.:	Request Type:	Created By:
Department:	Sub-Type:	Created On:
Workflow:	Request Status:	Request Status:
Priority:	Application:	Contact Name:
Assigned To:	Assigned Group:	Contact Phone:
Request Group:	Contact Email:	Contact Email:
Description:		

At the bottom of the interface, there are buttons for 'New', 'Rename', and 'Remove', along with arrow keys, 'Width', 'Height', 'Move To -->', and 'Preview'.

To add a new section:

1. Open the Request Type.
2. Click the **LAYOUT** tab.
3. Click **NEW** in the SECTIONS area (on the left side of the window). The INPUT window opens.





4. 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.

Tip

You can change the section name:

- a. Select the section name to be changed.
- b. Click **RENAME** and enter a new value in the Input window that appears. The change will be reflected immediately.

5. 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.

Note

- 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.

The screenshot displays a web-based form for configuring a request resolution system, divided into two main sections: 'Request Header' and 'Bug'.

**Request Header Section:**

- Created By:** johnsmith
- Department:** [Dropdown menu]
- Sub-Type:** [Dropdown menu]
- \*Workflow:** [Text input]
- Priority:** [Dropdown menu]
- Application:** [Dropdown menu]
- Assigned To:** [Dropdown menu]
- Assigned Group:** [Dropdown menu]
- Request Group:** [Text input]
- Description:** [Text input]
- Request Status:** Not Submitted
- Contact Name:** [Text input]
- Contact Phone:** [Text input]
- Contact Email:** [Text input]

**Bug Section:**

- Module:** [Dropdown menu]
- Platform:** [Dropdown menu]
- Impact:** [Dropdown menu]
- Error Log:** [Text input] with a **View URL** button
- Reproducible:**  Yes  No
- Steps To Replicate:** [Text area]
- Difficulty:** [Dropdown menu]
- Estimated Time to Complete:** [Text input]
- Resolution:** [Dropdown menu]
- Duplicate ID:** [Text input]
- Resolution Summary:** [Text input]

Both sections include a 'Close Window' button in the top right corner.

Figure 7-3 Request Field Layout Preview

# Chapter 8

## Integrating Participants into Your Request Resolution System

This chapter provides an overview for how to integrate Kintana users into your Request resolution process. It includes information on using Security Groups and controlling users' access to actions in Kintana.

This chapter discusses the following topics:

- *User Security and Participation - Overview*
- *Establishing Security Groups*
- *Setting Request Creation Security*
- *Setting Request Processing Security*
- *Setting Configuration Security*



Note

This chapter presents a number of configuration options available to you. It does not, however, provide detailed instructions on implementing each configuration. See "*Kintana Security Model*" for a comprehensive resource for configuring user access to Kintana screens and features.

### User Security and Participation - Overview

Kintana allows you to exercise a great deal of control over your Request resolution process. You can restrict users' actions around:

- **Request creation:**

- o Who can create Requests.
- o Who can use a specific Workflow.
- o Who can use specific Request Types.
- **Request processing:**
  - o Who can approve / process each step in the Workflow. For this restriction, you can enable access by specifying users or Security Groups. You can also provide access dynamically by having a Kintana Token resolve to provide access. See [“Mapping your Process into a Kintana Workflow”](#) on page 55 for more information.
  - o Who can view or edit certain fields in a Request. For this restriction, you can enable view or edit access to Request fields by specifying users or Security Groups. You can also provide access dynamically by having a Kintana Token resolve to provide access.
  - o Whether you only want “Participants” to process the Request. Participants are defined as the Assigned User, the creator of the Request, members of the Assigned Group, or any users who have access to the Workflow step(s).
- **Managing your Request resolution process:**
  - o Who can change the Workflow.
  - o Who can change each Request Type.



Kintana recommends using Security Groups or dynamic access (Tokens) to provide access to Kintana functionality whenever possible. You should 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 an organizational reorganization), you would have to update that list in many places on the workflow. By using a Security Group instead of a list of users, you can update the Security Group once, and the changes are propagated throughout the Workflow steps.

## Establishing Security Groups

Security Groups are used in Kintana to control who can access certain screens and functionality in Kintana. The following sections provide instructions on defining Security Groups:

- [Creating a Security Group by Specifying a List of Users](#)
- [Using Kintana's Resource Management to Control User Security](#)

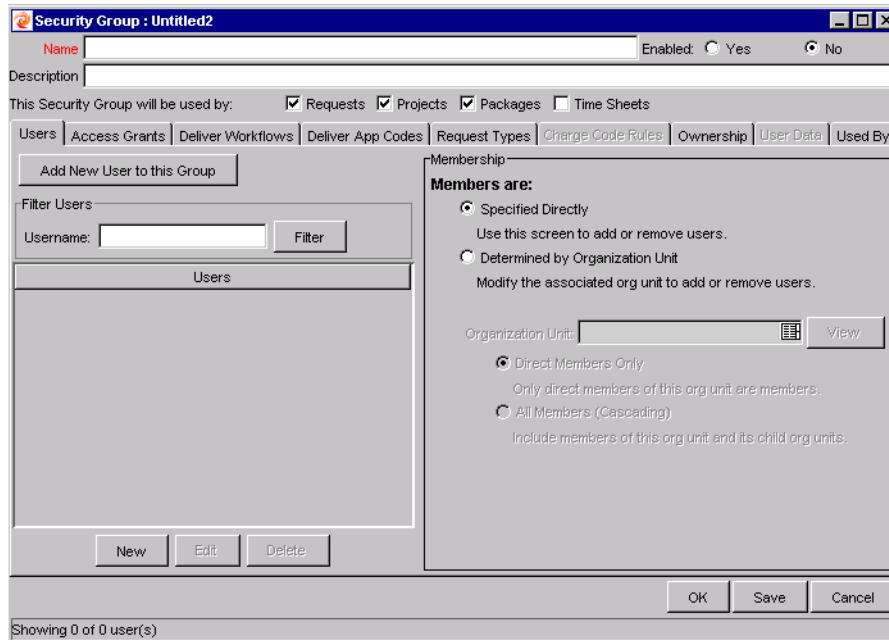
The general process for creating a Security Group is as follows:

1. Specify Security Group membership on the **USERS** tab. This can be accomplished by providing a list of users or by associating the group with an organization unit defined in Kintana.
2. Specify the screen and feature access by linking the appropriate Access Grants. See "[Access Grants](#)" on page 271 for details.
3. Specify which Request Types users in this Security Group can use when making Requests. This is set in the **REQUEST TYPES** tab.

### Creating a Security Group by Specifying a List of Users

To generate and define a new Security Group:

1. Click **NEW SECURITY GROUP** in the SECURITY GROUP WORKBENCH or select **FILE** -> **NEW** -> **SECURITY GROUP** from the menu. The SECURITY GROUP window opens.



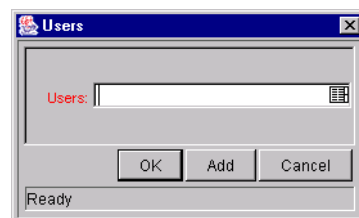
The screenshot shows a window titled "Security Group : Untitled2". It has a "Name" field and an "Enabled" radio button set to "No". Below is a "Description" field. A section labeled "This Security Group will be used by:" contains checkboxes for "Requests", "Projects", "Packages", and "Time Sheets", with "Requests", "Projects", and "Packages" checked. A tabbed interface at the top includes "Users", "Access Grants", "Deliver Workflows", "Deliver App Codes", "Request Types", "Charge Code Rules", "Ownership", "User Data", and "Used By". The "Users" tab is active, showing an "Add New User to this Group" button and a "Filter Users" section with a "Username:" field and a "Filter" button. Below this is a large empty area labeled "Users". At the bottom of the "Users" tab are "New", "Edit", and "Delete" buttons. To the right is a "Membership" section with radio buttons for "Specified Directly" (selected), "Determined by Organization Unit", "Direct Members Only", and "All Members (Cascading)". There is also an "Organization Unit:" field with a "View" button. At the bottom right of the window are "OK", "Save", and "Cancel" buttons. The status bar at the bottom left says "Showing 0 of 0 user(s)".

2. Enter the NAME and DESCRIPTION.
3. Select **YES** to enable this Security Group.

If the Security Group is not enabled, it does not appear as a choice when generating or updating users or Workflows.

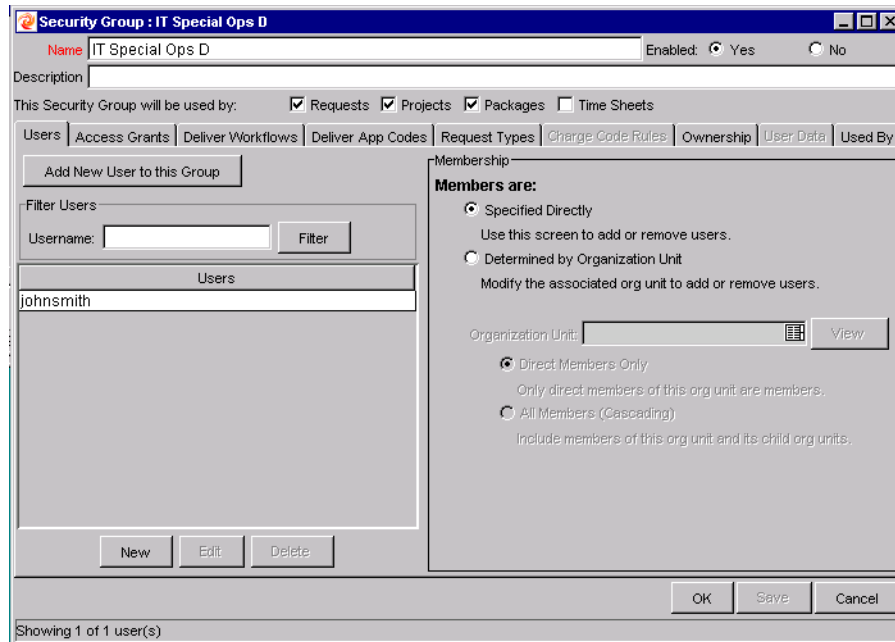
4. Select which Kintana entities (Requests, Projects or Packages) will use the Security Group by clicking their respective check boxes in the THIS SECURITY GROUP WILL BE USED BY field.
5. Link the desired Users to the Security Group.

- a. Click **ADD NEW USER TO THIS GROUP** in the **USERS** tab. The **USERS** window opens.



The screenshot shows a small dialog box titled "Users". It has a "Users:" label followed by a text input field and a list icon. Below the input field are "OK", "Add", and "Cancel" buttons. The status bar at the bottom says "Ready".

- b. Select the desired usernames from the **USERS** field and click **OK** to add your selection to the **USERS** tab.



6. Link the desired Access Grants. Each Access Grant enables certain functions performed on a Kintana screen. See "[Kintana Security Model](#)" for a description of each available access grant.
  - a. Select the desired Access Grants in the AVAILABLE ACCESS GRANTS list.
  - b. Click the right arrow button pointing to the LINKED ACCESS GRANTS list. The selected Access Grants are moved into the column.
7. Restrict the Security Group from using certain Request Types.
  - a. Click the **REQUEST TYPES** tab.
  - b. Select the Request Types in the ALLOWED REQUEST TYPES list.
  - c. Click the left arrow button pointing to the RESTRICTED REQUEST TYPES list. The selected Request Types are moved into the column.
8. Optional: If you plan on using the same Security Group for processing Packages in Kintana Deliver, specify which Workflows the Security Group can use in the **DELIVER WORKFLOWS** tab.

See "[Kintana Security Model](#)" for more information about allowing/restricting Deliver Workflows in Security Groups.

9. Click the **OWNERSHIP** tab and select the Ownership Groups that have the right to edit, copy or delete the current Security Group. See [“Setting Ownership for Security Groups”](#) on page 92 for more information about setting Ownership for a new or existing Security Group.
10. (Optional) Enter any necessary information in the **USER DATA** tab’s custom fields.
11. Click **OK** to register the current Security Group and close the SECURITY GROUP window. Click **SAVE** to save the information and leave the SECURITY GROUP window open.

## Using Kintana’s Resource Management to Control User Security

Users can also be associated to Security Groups through their inclusion in an organization model definition. Using Kintana’s resource management capabilities, a user can be placed into a model that includes security and access information. See [“Managing Resources in Kintana”](#) for details.

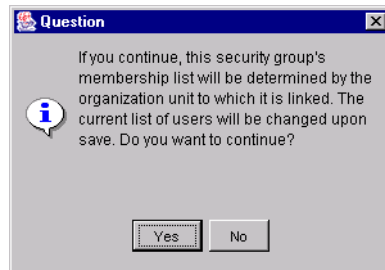
The screenshot shows the 'Security Group : Untitled1' window. At the top, there is a 'Name' field and an 'Enabled' toggle set to 'No'. Below is a 'Description' field. A section titled 'This Security Group will be used by:' contains three checked checkboxes: 'Requests', 'Projects', and 'Packages'. A tabbed interface below shows 'Users' selected, with other tabs including 'Access Grants', 'Deliver Workflows', 'Deliver App Codes', 'Request Types', 'Ownership', 'User Data', and 'Used By'. The 'Filter Users' section has a 'Username' input field and a 'Filter' button. The 'Users' list area is currently empty. At the bottom of this section are 'New', 'Edit', and 'Delete' buttons. The 'Membership' section on the right has a 'Members are:' heading with two radio buttons: 'Specified Directly' (selected) and 'Determined by Organization Unit'. Below 'Specified Directly' is the text 'Use this screen to add or remove users.' Below 'Determined by Organization Unit' is the text 'Modify the associated org unit to add or remove users.' There is an 'Organization Unit:' input field with a 'View' button. At the bottom of the membership section are two radio buttons: 'Direct Members Only' (selected) and 'All Members (Cascading)'. Below 'Direct Members Only' is the text 'Only direct members of this org unit are members.' Below 'All Members (Cascading)' is the text 'Include members of this org unit and its child org units.' At the bottom of the window are 'OK', 'Save', and 'Cancel' buttons. The status bar at the very bottom says 'Showing 0 of 0 user(s)'.

To define a Security Group to use the members of an organization unit:

1. Open the SECURITY GROUP window.



2. Select **DETERMINED BY ORGANIZATION UNIT** in the **MEMBERSHIP** section of the **USERS** tab. The following question dialog opens.



3. Click **YES**.



Note

When you select an Organization Unit to control user access to the Security Group, any users specified in the Users list will be replaced with the members of the organization unit.

4. Select the **ORGANIZATION UNIT**.
5. Select whether you want to include:
  - **Direct Members Only:**  
Only direct members of the specified organization unit.
  - **All Members (cascading)**  
Members of this organization unit and its child units.
6. Click **SAVE**.

#### Related Topics:

- ["Managing Resources in Kintana"](#)
- ["Kintana Security Model"](#)

## Setting Request Creation Security

You can control who can create certain Requests or use specific Request Types and Workflows. This provides a great deal of control over who can process changes of a certain type to specific environments. The following sections discuss how to control security related to Request creation:

- [Enabling Users to Create Requests](#)
- [Restricting Users from Selecting a Specific Workflow](#)
- [Restricting Users from Selecting a Specific Request Type](#)

### Enabling Users to Create Requests

You can control which Kintana users have the ability to create and submit Requests. To enable a user to create and submit a Request, ensure that the following are set.

Table 8-1. Settings required to enable a user to create Requests in Kintana

Setting	Value	Description
License	Kintana Create: Standard License	The Standard License provides a Kintana user with access to the Kintana interface, where the Request is created.  This is set in the USER window on the KINTANA WORKBENCH.

Table 8-1. Settings required to enable a user to create Requests in Kintana

Setting	Value	Description
Access Grants linked to the Security Group	Create: Edit Requests	<p>Standard License:</p> <ul style="list-style-type: none"> <li>• Allows the user to generate Requests.</li> <li>• User cannot change the Workflow when creating or editing a Request.</li> <li>• To edit the Request, user must be its creator, a member of the Workflow Steps security group or associated with the current contacts. Otherwise, user can only view the Request.</li> </ul> <p>Power License:</p> <ul style="list-style-type: none"> <li>• Has the same permissions as those listed for Standard License.</li> <li>• Allows user to delete the Request if the user is the creator and the Request has not been submitted.</li> </ul> <p>Access Grants are set in the SECURITY GROUP window.</p>

Table 8-1. Settings required to enable a user to create Requests in Kintana

Setting	Value	Description
	Create: Manage Requests	<p>Perform advanced Request processing actions: creating, editing, deleting, changing the Request's workflow, and overriding references. This access enables different functions depending on your license (standard versus power).</p> <p>Standard License:</p> <ul style="list-style-type: none"> <li>• User can change the Workflow when creating and editing a Request.</li> <li>• User always has permission to edit the Request.</li> <li>• Override and/or remove any References on any Request.</li> </ul> <p>Power License:</p> <ul style="list-style-type: none"> <li>• Has the same permissions as those listed for Standard License.</li> </ul> <p>User always has permission to delete or cancel a Request.</p> <p>Access Grants are set in the SECURITY GROUP window.</p>
Allowed Request Types in the SECURITY GROUP window	You must have at least one Request Type allowed.	<p>In order to process the intended Request correctly, you are required to select a Request Type when creating a Request. The Request Type you wish to use must be enabled in order for you to be able to create and submit a Request of that Type.</p> <p>This is set on the SECURITY GROUP window - <b>REQUEST TYPES</b> tab.</p>

Table 8-1. Settings required to enable a user to create Requests in Kintana

Setting	Value	Description
Allowed Request Types in the WORKFLOW window.	You must allow at least one Request Type in each Workflow used to resolve Requests.	<p>You can associate Request Types with Workflows such that only certain Request Types can be processed through the Workflow. The Request Type you wish to use must be enabled so that the user can create a Request when using that Workflow.</p> <p>You can also opt to restrict all new Request Types.</p> <p>The default Request Type to be used with this Workflow can also be specified.</p> <p>This is set on the WORKFLOW window - <b>REQUEST TYPES</b> tab.</p>
Allowed Security Groups in the REQUEST TYPE window.	You must allow at least one Security Group to create Requests of this Type.	<p>You can associate Security Groups with Request Types such that only certain Security Groups are allowed to create Requests of a particular Type.</p> <p>You can also opt to allow all Security Groups enabled for Requests to create Requests of this Type.</p> <p>New Security Groups can automatically be added to this window if you so choose.</p> <p>This is set on the REQUEST TYPE window - <b>USER ACCESS</b> tab.</p>



Note

Screen and function access provided through Access Grants are cumulative. If a user belongs to three different Security Groups, he will have all access provided to each of the groups. Therefore, to restrict certain screen and feature access, you need to remove the user from any Security Group that grants that access.

You can use the **ACCESS GRANTS** tabs in the **USER** window to see all Security Groups where specific access grants are included. You can then:

- Remove the user from the Security Group (using the **SECURITY GROUP** tab on the **USER** window)
- Remove the Access Grants from the Security Group (in the Security Group window). Note: you should only do this if no one in that Security Group needs the access provided in that Access Grant.

## Restricting Users from Selecting a Specific Workflow

You can restrict users from selecting specific Workflows when creating a new Request. To do this, ensure that the following conditions are met.

*Table 8-2. Settings required to restrict Workflow selection*

Setting	Value	Description
Allowed Workflows in the <b>REQUEST TYPE</b> window	<p>Include the Workflows that you would like to allow.</p> <p>You can opt to allow all Workflows to be used with the Request Type.</p>	<p>When creating a Request, you are required to select a Workflow for the Request to proceed through. Users (in the Security Group) will not be able to select any Workflows not included in the <b>WORKFLOWS</b> tab of the <b>REQUEST TYPE</b> window.</p> <p>This is set on the <b>REQUEST TYPE</b> window - <b>WORKFLOWS</b> tab.</p>

Table 8-2. Settings required to restrict Workflow selection

Setting	Value	Description
Allowed Request Types in the WORKFLOW window.	You must allow at least one Request Type in each Workflow used to resolve Requests.	<p>You can associate Request Types with Workflows such that only certain Request Types can be processed through the Workflow. The Request Type you wish to use must be enabled so that the user can create a Request when using that Workflow.</p> <p>You can also opt to restrict all new Request Types.</p> <p>The default Request Type to be used with this Workflow can also be specified.</p> <p>This is set on the WORKFLOW window - REQUEST TYPES tab.</p>

## Restricting Users from Selecting a Specific Request Type

You can restrict users from selecting specific Request Types when creating a new Request. To do this, ensure that the following conditions are met.

Table 8-3. Settings required to restrict Request Type selection

Setting	Value	Description
Allowed Security Groups in the REQUEST TYPE window.	You must allow at least one Security Group to create Requests of this Type.	<p>You can associate Security Groups with Request Types such that only certain Security Groups are allowed to create Requests of a particular Type.</p> <p>You can also opt to allow all Security Groups enabled for Requests to create Requests of this Type.</p> <p>New Security Groups can automatically be added to this window if you so choose.</p> <p>This is set on the REQUEST TYPE window - USER ACCESS tab.</p>

Table 8-3. Settings required to restrict Request Type selection

Setting	Value	Description
Allowed Request Types in the WORKFLOW window.	You must allow at least one Request Type in each Workflow used to resolve Requests.	<p>You can associate Request Types with Workflows such that only certain Request Types can be processed through the Workflow. The Request Type you wish to use must be enabled so that the user can create a Request when using that Workflow.</p> <p>You can also opt to restrict all new Request Types.</p> <p>The default Request Type to be used with this Workflow can also be specified.</p> <p>This is set on the WORKFLOW window - REQUEST TYPES tab.</p>

## Setting Request Processing Security

You can control who can process Requests following a Request submission. You can also control who can act on certain steps (decisions and executions) in your process. The following sections discuss how to control security related to Request processing:

- [\*Providing Users with General Access to Update Requests\*](#)
- [\*Enabling Users to Act on a Specific Workflow Step\*](#)
- [\*Restricting Request Processing to Participants\*](#)

### Providing Users with General Access to Update Requests

All users who will be processing Requests must meet the following conditions:



Table 8-4. Settings required to enable a user to process Requests in Kintana

Setting	Value	Description
License (at least one is required)	Kintana Create: Standard	<p>The Standard License provides a Kintana user with access to the Kintana interface. Users can act on all decision Workflow steps.</p> <p>This is set in the USER window on the KINTANA WORKBENCH.</p>
Access Grants linked to the Security Group	Create: Edit Requests	<p>Perform basic Request processing actions: create Requests, edit certain Requests, and delete your un-submitted Requests depending on your license (standard versus power).</p> <p>Standard License:</p> <ul style="list-style-type: none"> <li>• Allows the user to generate Requests.</li> <li>• User cannot change the Workflow when creating or editing a Request.</li> <li>• To edit the Request, user must be its creator, the 'assigned to' user, a member of the assigned group, a member of the Workflow Steps security group or associated with the current contacts. Otherwise, user can only view the Request.</li> </ul> <p>Power License:</p> <ul style="list-style-type: none"> <li>• Has the same permissions as those listed for Standard License.</li> </ul> <p>Allows user to delete the Request if the user is the creator and the Request has not been submitted.</p> <p>Access Grants are set in the SECURITY GROUP window.</p>

Table 8-4. Settings required to enable a user to process Requests in Kintana

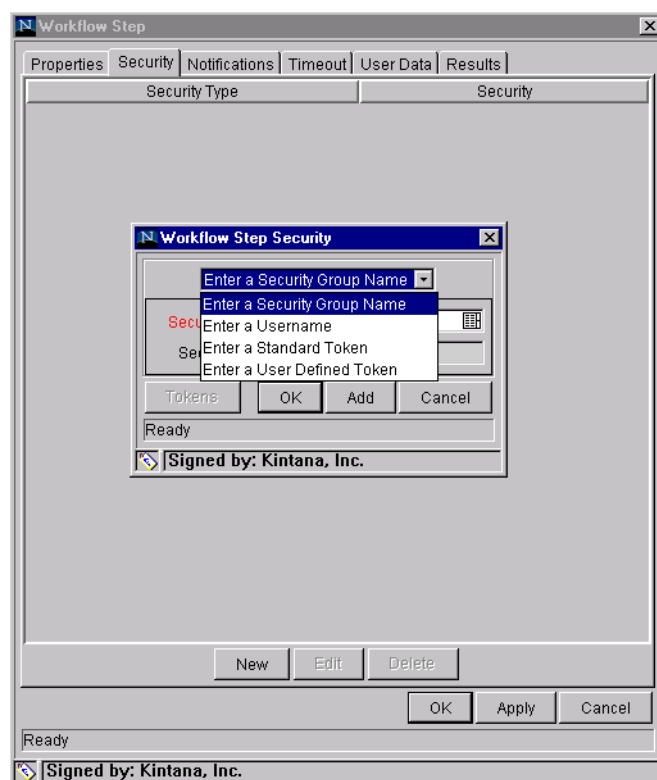
Setting	Value	Description
	Create: Manage Requests	<p>Perform advanced Request processing actions: creating, editing, deleting, changing the Request's workflow, and overriding references. This access enables different functions depending on your license (standard versus power).</p> <p>Standard License:</p> <ul style="list-style-type: none"> <li>• User can change the Workflow when creating and editing a Request.</li> <li>• User always has permission to edit the Request.</li> <li>• Override and/or remove any References on any Request.</li> </ul> <p>Power License:</p> <ul style="list-style-type: none"> <li>• Has the same permissions as those listed for Standard License.</li> </ul> <p>User always has permission to delete or cancel a Request.</p> <p>Access Grants are set in the SECURITY GROUP window.</p>
	Create: Allow Request Field Updates	<p>This Access Grant allows the user to view and update any Request regardless of whether the user is its creator or Contact.</p> <p>Access Grants are set in the SECURITY GROUP window.</p>
	Create: Override Participant Restriction	<p>This Access Grant allows the user to view a Request regardless of whether the user is its creator, the ASSIGNED TO user, a member of the ASSIGNED GROUP or a member of the Workflow Steps security group.</p> <p>Access Grants are set in the SECURITY GROUP window.</p>

## Enabling Users to Act on a Specific Workflow Step

You need to specify who can act on each step in the Request resolution Workflow. Only people who are specified on the **SECURITY** tab in the **WORKFLOW STEP** window will be able to process Packages and Package Lines at that step.

To specify the users who can act on a specific Workflow step:

1. Open the Workflow.
2. Click the **LAYOUT** tab.
3. Double click on the step that you would like to configure. The **WORKFLOW STEP** window opens. Note: the **WORKFLOW STEP** window also opens when first adding a step to the **LAYOUT** tab.
4. Click the **SECURITY** tab.
5. Click **NEW**. The **WORKFLOW STEP SECURITY** window opens.



6. Select the method for specifying the step security from the drop down list: Security Group Name, Username, Standard Token, User Defined Token.

Selecting a value from this field automatically updates the other fields on this window. For example, selecting **ENTER A USERNAME** will change the **SECURITY GROUP** field to **USERNAME**.

7. Specify the Security Groups, Usernames, or Tokens that will control the access to this step.
8. Click **OK**. The security specification is added to the **SECURITY** tab. You can add additional specifications to the step by clicking **NEW** and repeating the above process. You can therefore select to control the step's security using a combination of multiple Security Groups, Usernames and Tokens.
9. Click **OK** to save and close the window.



Tip

1. Consider assigning a Security Group to each decision, execution and condition step, even though many of these steps will proceed automatically. If a command fails, or a condition is not met, you may need to manually override the step.
2. You may also want to consider assigning a “Request Manager” Security Group to each step. That group could be configured with global access to act on every step in the process. Again, this could help avoid bottlenecks by providing a small group with permission to process stalled Requests.
3. You may want to avoid specifying a single user as the only person who can act on a Workflow step. This would require a process update (re-configuration) when that user changes roles or leaves the company. Better to grant access dynamically using a Token or Security Group.

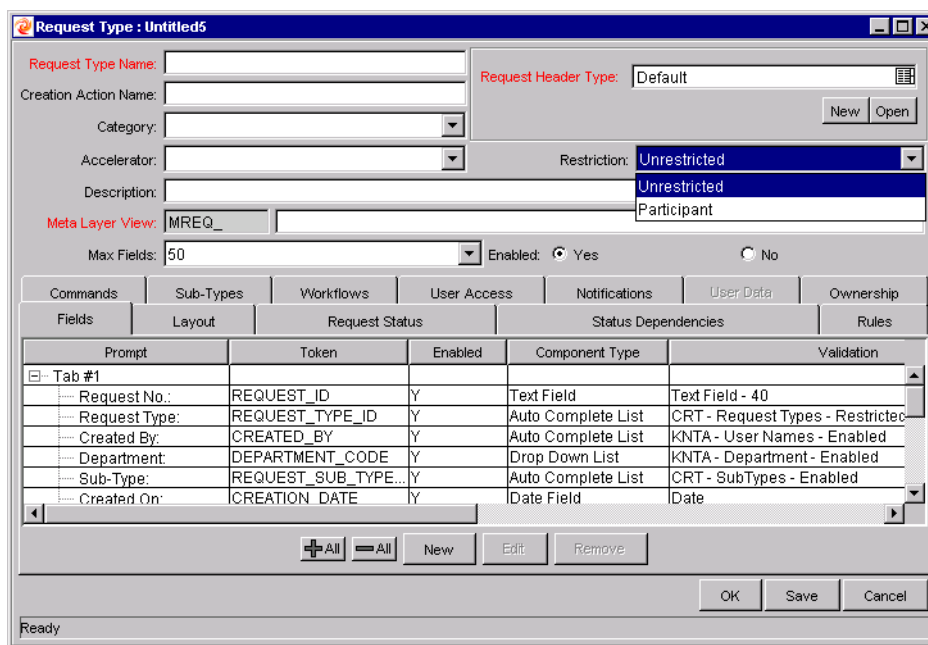
## Restricting Request Processing to Participants

The **RESTRICTION** drop down list in the **REQUEST TYPE** window lets you determine who can have access to Requests. Restricting access to Participants means that when non-Participant users search for Requests, they will not see a Request that uses the current Workflow. In this instance, Participants are defined as:

- The **ASSIGNED TO User**
- The creator of the Request
- Members of the **ASSIGNED GROUP**

- Any users who have access to the Workflow Step(s)

To let all Kintana users access Request using the current Workflow, select **UNRESTRICTED**.



To restrict the number of Kintana users who can access Requests using the current Workflow to Participants of the Requests, select **PARTICIPANT**.

## Setting Configuration Security

A critical part of ensuring successful Request resolution is ensuring that your resolution process is altered only by the correct people. Kintana allows you to set security around the Kintana configuration. You can establish configuration security around all of the Kintana configuration entities. This includes such activities as controlling:

- Who can change the Workflow.
- Who can change each Request Type.
- Who can change the Security Group definitions.

The following sections discuss some options for securing your Kintana configurations:

- [Setting Ownership for Kintana Configuration Entities](#)
- [Removing Access Grants](#)

## Setting Ownership for Kintana Configuration Entities

Different groups of Kintana users have ownership and control over Kintana entities. These groups are referred to as Ownership Groups. Unless a ‘global’ permission has been designated to all users for an entity, members of Ownership Groups are the only users who have the right to edit, delete or copy that entity. The Ownership Groups must also have the proper access grant for the entity in order to complete those tasks. For example, the `EDIT WORKFLOWS` Access Grant is needed to edit Workflows and Workflow Steps.

You can assign multiple Ownership Groups to the various entities. The Ownership Groups will have sole control over the entity, providing greater security. Ownership Groups are defined in the `SECURITY GROUP` window. Security Groups become Ownership Groups when used in the Ownership capacity.

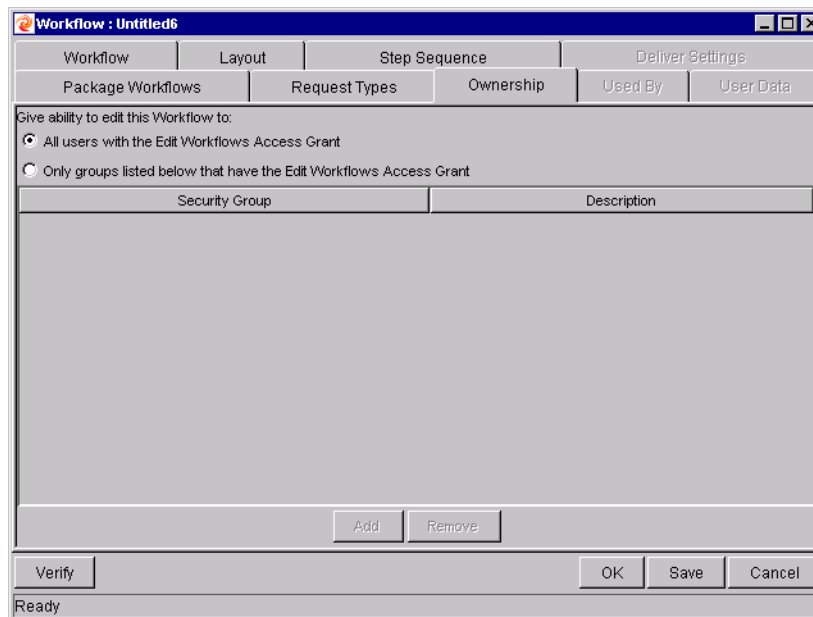
You can select to specify Ownership Groups for the following entities involved in your deployment process:

- Workflows
- Workflow Steps
- Request Types
- Request Header Types
- Security Groups
- User Definitions
- Report Types
- Validations
- Special Commands

The Ownership setting is accessed through the individual entity windows in the Kintana Workbench. For example, to set the Ownership for Workflows:

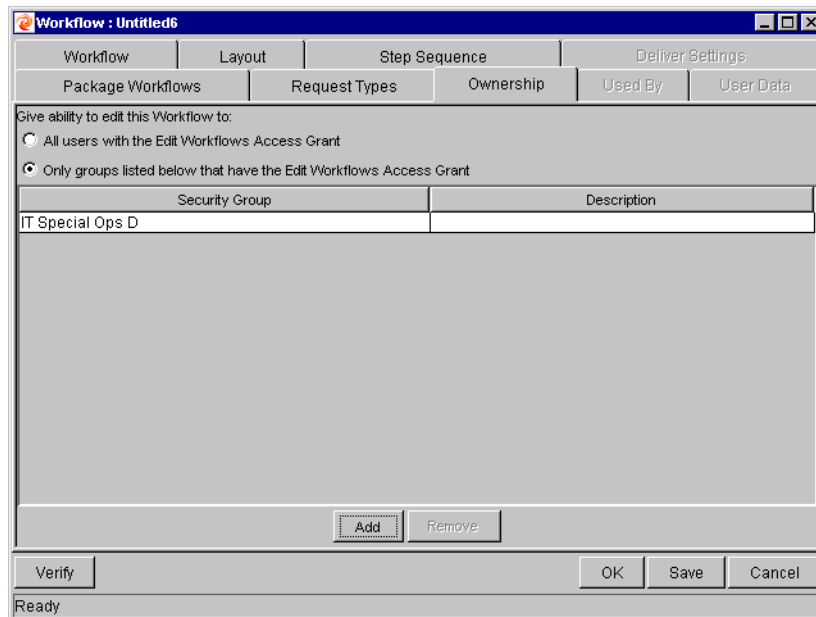
1. Open the Workflow.

2. Click the **OWNERSHIP** tab.



3. Click the **ONLY GROUPS LISTED BELOW THAT HAVE THE EDIT WORKFLOWS ACCESS GRANT** radio button.
4. Click **ADD**. The **ADD SECURITY GROUP** window opens.
5. Select the **SECURITY GROUP**.
6. Click **ADD** to add the current Security Group and continue adding more Security Groups. Click **OK** to add the current Security Group and close the **ADD SECURITY GROUP** window.

The Security Group(s) you selected displays in the **OWNERSHIP** tab under the **SECURITY GROUP** column.



7. Click **OK** to save the selection and close the **WORKFLOW** window. Click **SAVE** to save the selection and leave the **Workflow** window open.



Note

The **SYS ADMIN: OWNERSHIP OVERRIDE** access grant allows the user to access and edit configuration entities even if he is not a member of one of the entity's **Ownership Groups**.

## Removing Access Grants

You can also restrict the ability to modify Kintana configuration entities by removing the user from any **Security Group** that grants that access.

You can use the **ACCESS GRANTS** tabs in the **USER** window to see all **Security Groups** where specific access grants are included. You can then either:

- Remove the user from the **Security Group** (using the **SECURITY GROUP** tab on the **USER** window)
- Remove the **Access Grants** from the **Security Group** (in the **SECURITY GROUP** window). Note: you should only do this if no one in that **Security Group** needs the access provided in that **Access Grant**.

The following table lists the access grants that provide edit access to different Kintana configuration entities.



Table 8-5. Access Grants for editing Kintana configuration entities

Access Grant	Description
Create: Edit Contacts	Allows the user to generate, update and delete Contacts.
Config: Edit Report Types	Allows the user to generate, update and delete Report Types.
Create: Edit Request Header Types	Allows the user to generate, update and delete Request Header Types.
Create: Edit Request Types	Allows the user to generate, update and delete Request Types.
Sys Admin: Edit Security Groups	Allows the user to generate, update and delete Security Groups.
Config: Edit Special Commands	Allows the user to generate, update and delete Special Commands.
Config: Edit User Data	Allows the user to generate, update and delete User Data.
Sys Admin: Edit Users	Users Allows the user to generate, update and delete Users.
Config: Edit Validation Values	Allows the user to generate, update and delete Validation Values.
Config: Edit Validations	Allows the user to generate, update and delete Validations.
Config: Edit Workflows	Allows the user to generate, update and delete Workflows.



# Chapter 9

## Setting Up Communication Paths

This chapter provides an overview for different modes of communication that you can use in your Request resolution system. Kintana features three main devices for communicating status related to your Request resolution process:

- **Email Notifications:**  
Each Workflow step can be configured to send an email to specified users when the step becomes eligible, has a specific result, or encounters an error. Using notifications at key points in your process ensures a speedy resolution by notifying appropriate parties of actions required by them or complications during the process.

Email notifications can also be sent when a field in a Request changes.

- **Kintana Dashboard:**  
The Dashboard provides an interface through which you can quickly assess the current state of the deployments. Personalize your Dashboard to display status information that is most meaningful to your role. For example, Financial system users may only want to see the Requests that they submitted, whereas the Financial manager may want to have visibility into each critical Request currently in progress.
- **Reports:**  
Kintana includes a number of reports that can be used to assess Request status. Kintana also publishes a reporting meta layer that you can use to build your own custom reports. Kintana Reports can be scheduled to run periodically.

This chapter illustrates how notifications, Dashboard components, and reports can be used to monitor and control the resolution process. This chapter discusses the following topics:

- *Adding Notifications to Workflow Steps*
- *Setting Notifications on Request Field Changes*

- [Configuring Your Dashboard](#)
- [Configuring Reports](#)

## Adding Notifications to Workflow Steps

When configuring a Notification for a Workflow step, you need to consider the following:

- When to send it.
- Who should receive it.
- What the message should say.

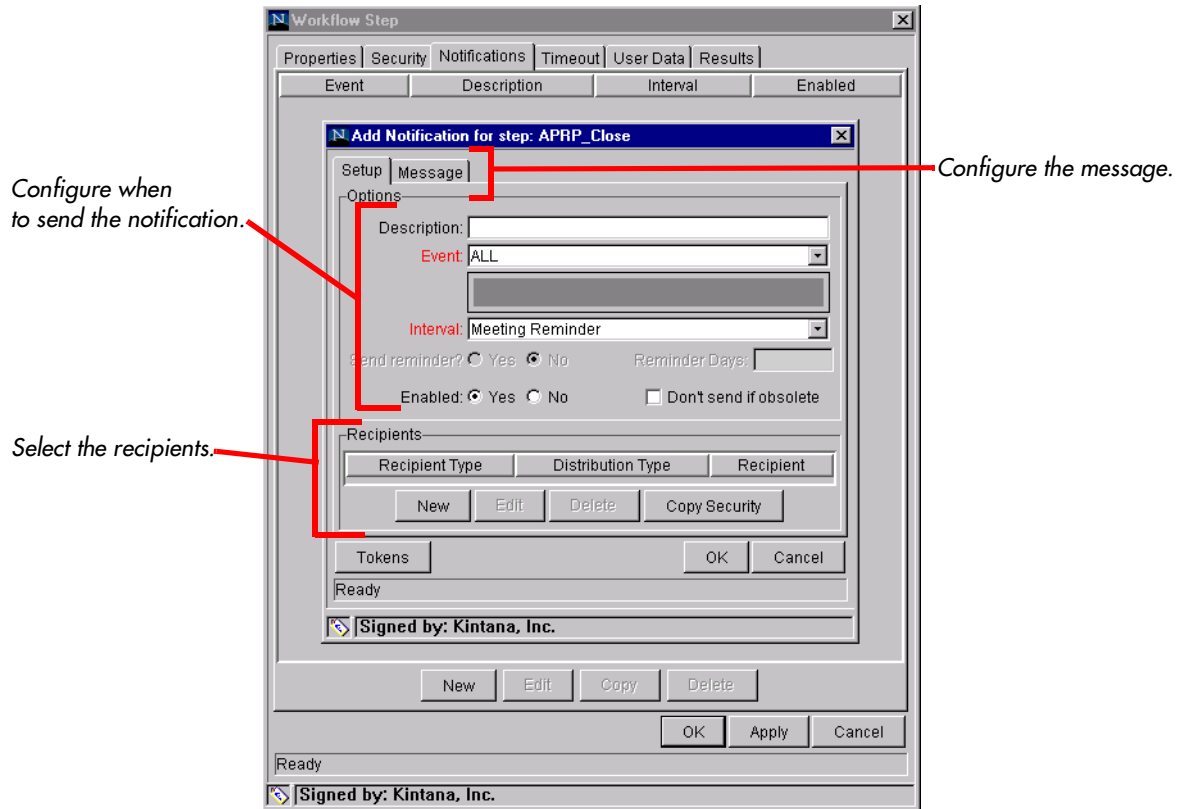
The following sections discuss different aspects of notifications.

- [Adding a Notification to a Workflow step - Overview](#)
- [Configuring When to Send a Notification](#)
- [Configuring the Notification Recipients](#)
- [Configuring the Notification Message](#)

### Adding a Notification to a Workflow step - Overview

To add a notification to a Workflow step:

1. Open the Workflow.
2. Click the **LAYOUT** tab.
3. Double click on the step that you would like to configure. The **WORKFLOW STEP** window opens. Note: the **WORKFLOW STEP** window also opens when first adding a step to the **LAYOUT** tab.
4. Click the **NOTIFICATIONS** tab.
5. Click **NEW**. The **ADD NOTIFICATION FOR STEP** window opens.



6. Configure the following:

- When the notification is sent (EVENT and INTERVAL)
- Who receives the notification (RECIPIENTS)
- The body of the notification (MESSAGE)

7. Click **OK**. The notification specification is added to the **NOTIFICATIONS** tab. You can add additional specifications to the step by clicking **NEW** and repeating the above process. You can therefore select to send a different notification to different recipients for different events.

8. Click **OK** to save and close the window.

## Configuring When to Send a Notification

Each Workflow step can be configured to send an email when the step becomes eligible, has a specific result, or encounters an error. The following topics are discussed:

- *Sending a notification when a step becomes eligible*
- *Sending a notification when a step has a specific result*
- *Sending a notification when the step has a specific error*
- *Configuring multiple notifications for a single step*
- *Specifying the Time the Notification is Sent*

### *Sending a notification when a step becomes eligible*

To send a notification when a Workflow step becomes eligible, configure the notification as indicated below.

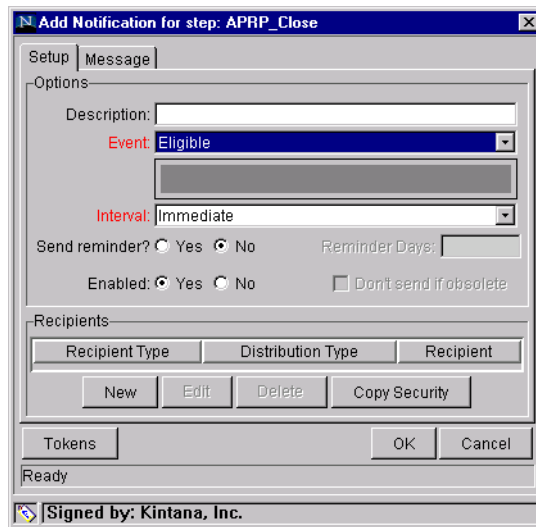


Table 9-1. Workflow step notification configuration - send on eligible

Field	Value	Notes
EVENT	ELIGIBLE	

Table 9-1. Workflow step notification configuration - send on eligible

Field	Value	Notes
INTERVAL	IMMEDIATE	<p>You can select to send the notification 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.</p> <p>Note also that multiple notifications to a single recipient can be batched and sent together. Selecting an interval other than "Immediate" will allow this batching to occur.</p> <p>See "<a href="#">Configuring the Notification Intervals</a>" on page 193 for instructions on configuring this.</p>
SEND REMINDER	Yes/No	<p>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.'</p>
ENABLED	YES	

### *Sending a notification when a step has a specific result*

You can configure the notification to be sent when a Workflow step results in a specific decision or execution result. The value for these events is determined by the Workflow step source's validation.

To send a notification when a Workflow has a specific result, configure the notification as indicated below.

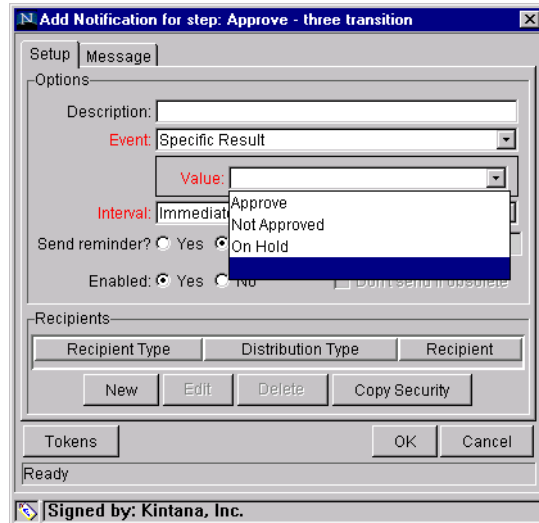


Table 9-2. Workflow step notification configuration - send on step result

Field	Value	Notes
EVENT	<b>SPECIFIC RESULT</b>	
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	<b>IMMEDIATE</b>	<p>You can select to send the notification 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.</p> <p>Note also that multiple notifications to a single recipient can be batched and sent together. Selecting an interval other than "Immediate" will allow this batching to occur.</p> <p>See <i>"Configuring the Notification Intervals"</i> on page 193 for instructions on configuring this.</p>



Table 9-2. Workflow step notification configuration - send on step result

Field	Value	Notes
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	

### *Sending a notification when the step has a specific error*

To send a notification when a Workflow has a specific error, configure the notification as indicated below.

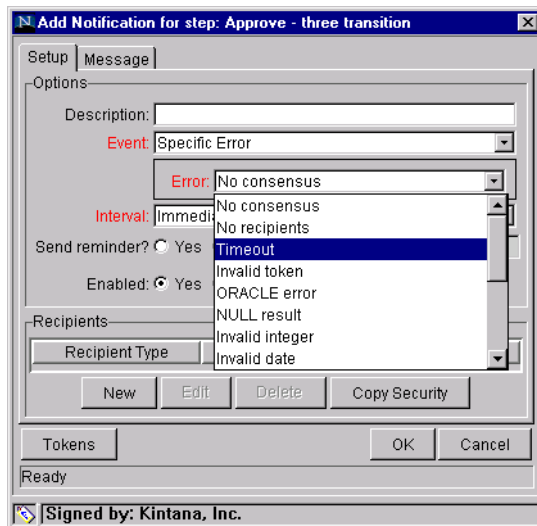


Table 9-3. Workflow step notification configuration - send on error

Field	Value	Notes
EVENT	SPECIFIC ERROR	
ERROR	Select the value to trigger the notification.	This is a standard set of errors. See <a href="#">“Specific Errors for Workflow Steps”</a> on page 190.

Table 9-3. Workflow step notification configuration - send on error

Field	Value	Notes
INTERVAL	IMMEDIATE	<p>You can select to send the notification 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.</p> <p>Note also that multiple notifications to a single recipient can be batched and sent together. Selecting an interval other than "Immediate" will allow this batching to occur.</p> <p>See "<a href="#">Configuring the Notification Intervals</a>" on page 193 for instructions on configuring this.</p>
SEND REMINDER	YES/NO	<p>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.'</p>
ENABLED	YES	

### Specific Errors for Workflow Steps

The following errors can cause a notification to be sent.

Table 9-4. 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.

Table 9-4. Specific Errors for Workflow Steps

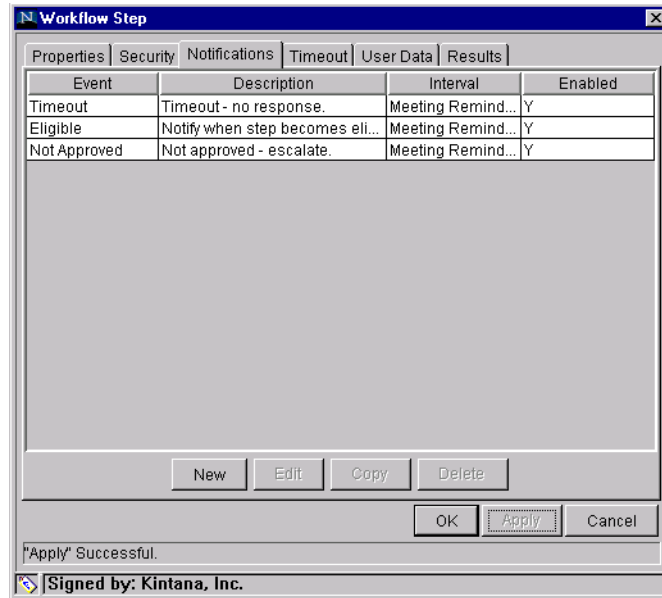
Specific Error	Meaning
<b>ORACLE ERROR</b>	Failed PL/SQL Execution.
<b>NULL RESULT</b>	No result is returned from the execution.
<b>INVALID INTEGER</b>	Validation includes an invalid value in the Integer field.
<b>INVALID DATE</b>	Validation includes an invalid value in the Date field.
<b>COMMAND EXECUTION ERROR</b>	Execution engine has failed or has a problem.
<b>INVALID RESULT</b>	Execution or Subworkflow has returned a result not included in the Validation.
<b>PARENT CLOSED</b>	For wf_receive or wf_jump steps, a Request is expecting a message from a Package Line that is cancelled or closed.
<b>CHILD CLOSED</b>	For wf_receive or wf_jump steps, a Package Line is expecting a message from a Request that is cancelled or closed.
<b>NO PARENT</b>	For wf_receive or wf_jump steps, a Request is expecting a message from a Package Line that has been deleted.
<b>NO CHILD</b>	For wf_receive or wf_jump steps, a Package Line is expecting a message from a Request that has been deleted.
<b>MULTIPLE JUMP RESULTS</b>	For wf_jump steps in a Package Line, different result values were used to transition to the step.
<b>MULTIPLE RETURN RESULTS</b>	When the Package Level subworkflow receives multiple results from Package Lines that traversed through it.

### Configuring multiple notifications for a single step

You can configure multiple notifications for each Workflow step. This can be useful in the following sample situations:

- Sending a different message depending on the result of the step
- Sending a different message depending on the type error
- Sending the notification 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

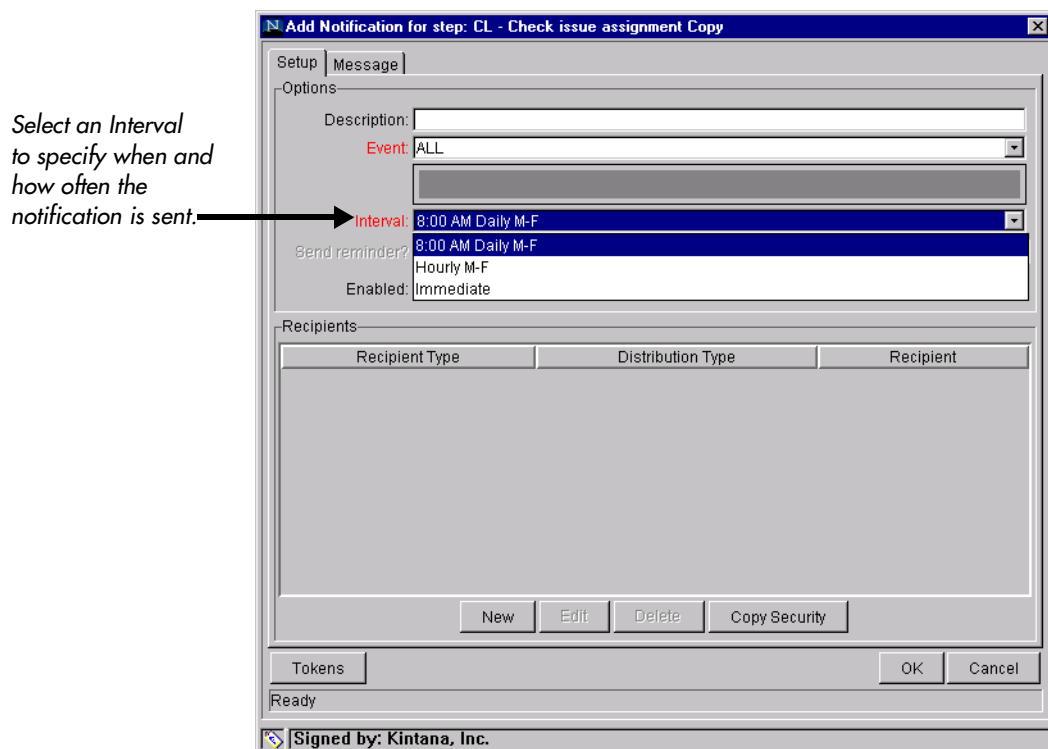
To configure multiple notifications for a Workflow step, simply add multiple notifications to the same Workflow step window.



In this example, one set of users is notified when the step becomes eligible. Then, depending on the outcome of the step, different groups are notified. If the step experiences a “TIMEOUT” error event, then the user responsible for acting on the step is notified. If the step results in the specific result of “NOT APPROVED,” then a notification is sent to the deployment manager.

### *Specifying the Time the Notification is Sent*

Use the INTERVAL field on the Workflow step to specify when the notification will be sent.



Select an Interval to specify when and how often the notification is sent.

The interval determines how frequently the notification will be sent. Kintana provides the following pre-configured intervals:

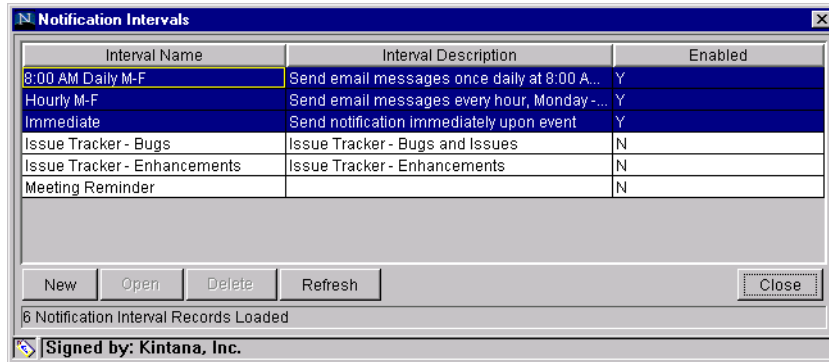
- **8:00 AM DAILY M-F:** This notification is sent every 8:00 AM on the next available work day after the notification event occurs.
- **HOURLY M-F:** This notification is sent every hour, starting on the next available work day after the notification event occurs.
- **IMMEDIATE:** This notification is sent immediately.

## Configuring the Notification Intervals

Notifications are configured on the NOTIFICATION TEMPLATES WORKBENCH. To configure the Notification Intervals:

1. Click the **CONFIGURATION** screen group and click the **NOTIFICATION TEMPLATES** icon.

2. Select **NOTIFICATION TEMPLATES ->INTERVALS** from the menu. The NOTIFICATION INTERVALS window opens.



3. Click **NEW** or **OPEN** to access the NOTIFICATION INTERVAL: NEW window. Enter the required information on the **INTERVAL** tab. These fields are defined in [Table 9-5](#).

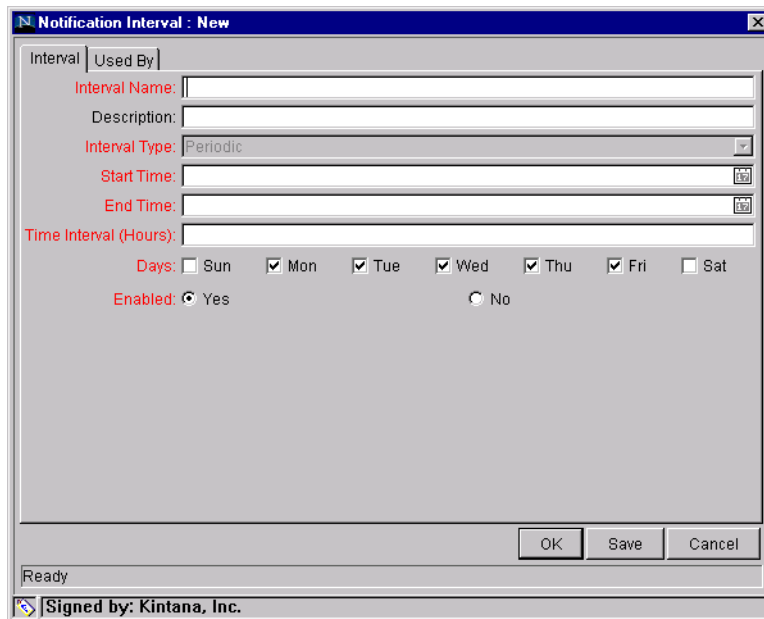


Table 9-5. Notification Intervals

Field Name	Description
Interval Name	This is the name assigned to the interval.
Description	Free form description of this interval.

Table 9-5. Notification Intervals

Field Name	Description
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 <b>Yes</b> is set, this interval is selectable. If <b>No</b> is set, this interval is unavailable.

4. Click **OK**. The new interval is added to the NOTIFICATION INTERVALS window.
5. Click **CLOSE** to close the window.

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 of the 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.

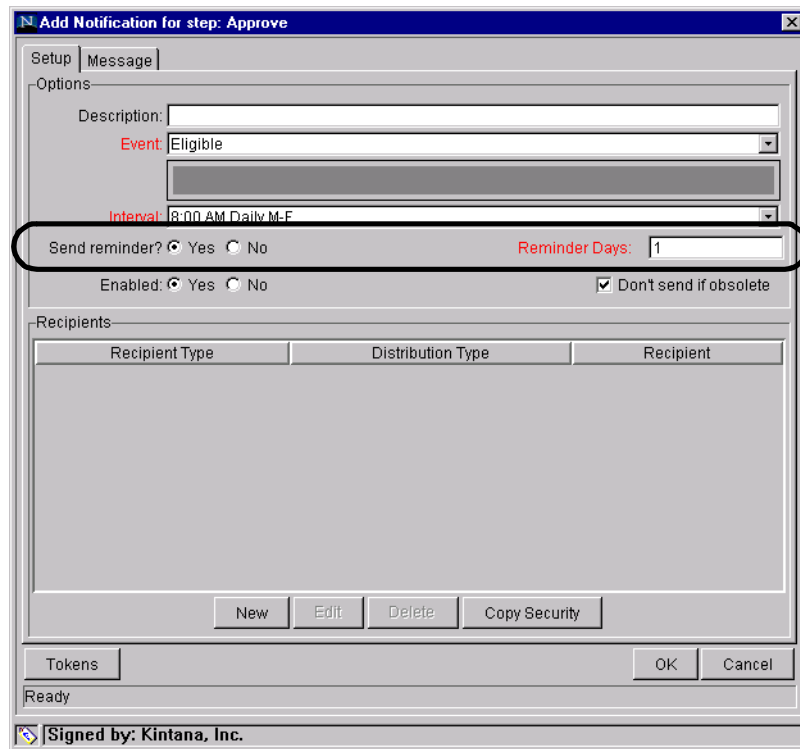
### *Sending a follow up notification (reminder)*

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 configure a notification to re-send after a period of time, configure the notification as indicated below.

Table 9-6. Workflow step notification configuration - send on error

Field	Value	Notes
Event		You can select any event except for <b>ALL</b> .
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.



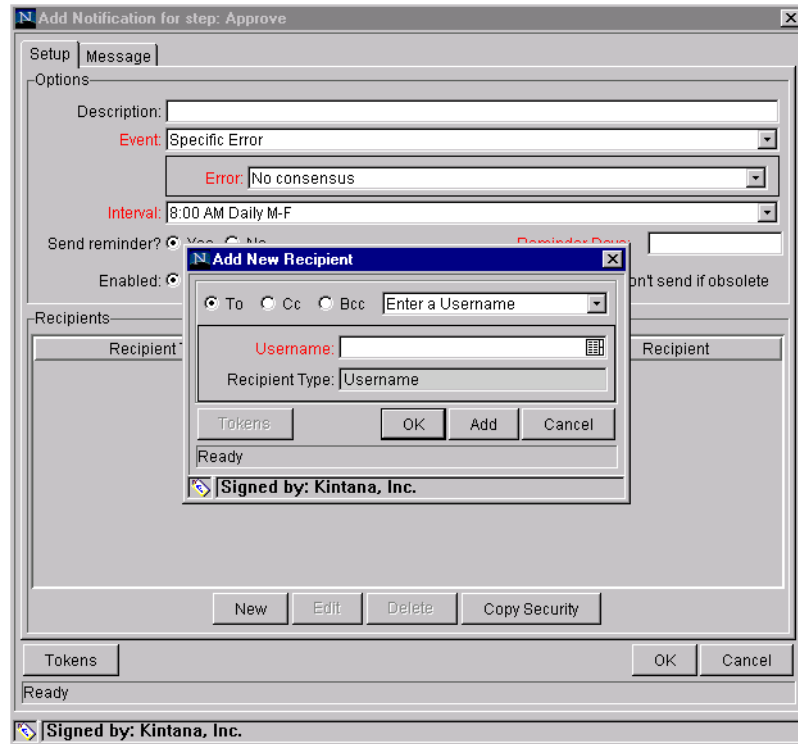
## Configuring the Notification Recipients

When creating a notification, at least one recipient must be added for the message. The recipient can be a specific Kintana user, all members of a Security Group, or any email address.

To add a recipient to a notification:



1. Click **NEW** in the ADD NOTIFICATION FOR STEP window. The ADD NEW RECIPIENT window opens.



2. Select how you would like to specify the recipient from the drop down list. You can select to:
  - **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 to send the notification to.
  - **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 field below. For example, selecting **ENTER A SECURITY GROUP** will change the field below to SECURITY GROUP.

3. Enter the specific value that corresponds to the recipient type selected above. This can be a Username, Email Address, Security Group, or a Token.

### *Recipient Configuration Tips*

#### Tip 1:

Kintana recommends using Security Groups or dynamic access (Tokens) to define the notification recipients whenever possible. You should 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), you would have to manually update that list. By using a Security Group instead of a list of users, you can update the Security Group once, and the changes are propagated throughout the Workflow steps.

#### Tip 2:

Use Tokens when sending a notification to an undetermined party. For example, you can configure the notification to be sent to the Assigned to User by specifying **[REQ.ASSIGNED\_TO\_USERID]** in the ADD NEW RECIPIENT window.

## Configuring the Notification Message

You can construct the notification's message to ensure that it contains the correct information or instructions for the recipient. For example, if the notification was sent to instruct the user that a Request requires his approval, the message should instruct him to log onto Kintana and update the Request's status. Additionally, the notification should include a link (URL) to the referenced Request.

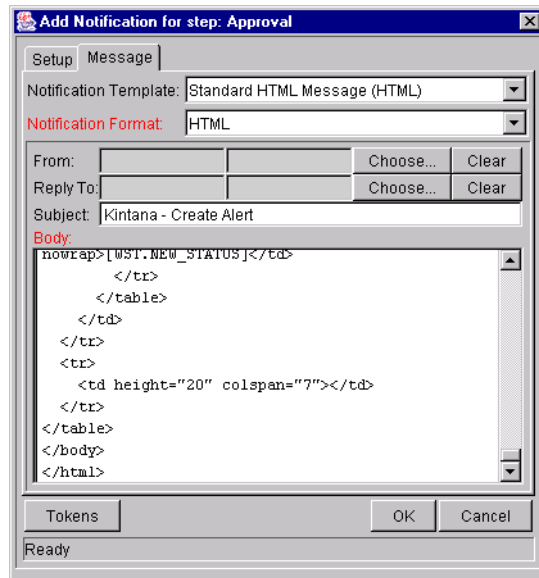
Kintana provides a number of features to make the notifications easier to configure and use:

- You can select from a number of pre-configured notification templates to more quickly construct the body of your message. See "[Configuration Workbench Reference](#)" for more detailed information on notification templates.
- The body of the notification can be plain text or HTML.

- You can include multiple Tokens 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. Click the **MESSAGE** tab on the ADD NOTIFICATION FOR STEP window.



2. Select a NOTIFICATION TEMPLATE. This updates the contents in the BODY section with the information defined for the selected template.
3. Select **HTML** or **PLAIN TEXT** from the NOTIFICATION FORMAT field.

Selecting **HTML** allows you more flexibility when formatting the look and feel of your notification. You can write and test the HTML code in any HTML editor and then paste the code into the BODY window.

4. Select values for the FROM and REPLY TO fields.
5. 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 9-7](#) for a list of these tokens.
  - Token for the URL to the Package (Workbench or Kintana interface). See [Table 9-7](#) for a list of these tokens.

- Tokens in the Body of the message:  
Click the **TOKENS** button to access the Token Builder window where you can select Tokens to add 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.
6. Click **OK** to save the notification specification.

### Using Tokens in the Message Body

You can select any of the available tokens accessed through the Token builder to include in the body of your message. You should note, however, that not all Tokens will resolve in all situations. As a general rule, Tokens associated with the Request or Workflow will resolve.

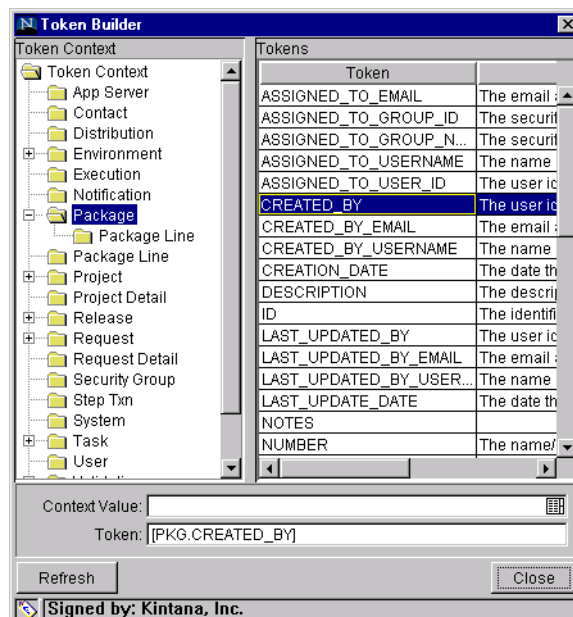


Figure 9-1 Token Builder Window

### Including URLs to Open the Request (Smart URLs)

Notifications can be configured in the body of the email notification to include the Web address (URL) for the following entities:

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

If end-users are viewing their mail with a Web-based mail reader (such as Microsoft Outlook or Netscape Messenger), they can then 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 the user 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 9-7](#).

*Table 9-7. Smart URL Tokens*

Smart URL Token	Description
PACKAGE_URL	Provides a URL that loads the Package Details page in the Kintana interface.
WORKBENCH_PACKAGE_URL	Provides a URL that loads the Package window in the Kintana Workbench.
REQUEST_URL	Provides a URL that loads the Request Details page in the Kintana interface.

### Smart URLs in HTML Formatted Messages

If you are using an HTML formatted message, you need to use an alternate Token to provide a link to the Request.

Table 9-8. Smart URL Tokens in HTML Format

Smart URL Token	Description
REQUEST_ID_LINK	Provides a link that loads the Request Detail page in the Kintana interface.

The Token will resolve to the following format:

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

In the Notification, the link would appear as:

Request Name



Note

These Tokens can also be used in plain-text formatted Notifications. They will appear with the HTML tags showing.

## Setting Notifications on Request Field Changes

Request Types can be configured to send Notifications when a particular field changes.



Example

A Request Type can be configured to send a Notification when the ASSIGNED GROUP field changes to **DEV MANAGERS**.



Note

If the Request Header Type is changed, the Notifications must be set up again.

The following fields are supported for Notification functionality:

- PRIORITY
- ASSIGNED TO
- CONTACT NAME
- ASSIGNED GROUP

- APPLICATION
- DEPARTMENT
- SUB-TYPE
- WORKFLOW
- REQUEST GROUP
- COMPANY



Though Field Groups, Request custom fields, and User Data fields are not directly supported, you can use a Token evaluation Execution step to evaluate the content of these fields and trigger Notifications based on the results. This may be a preferred method when you want to control when in the resolution process the Notification gets sent, rather than simply having a Notification sent every time a particular field changes.

Request Type Notifications are configured from the **NOTIFICATIONS** tab in the REQUEST TYPE window. Click New to open the ADD NOTIFICATION window.

## Configuring a Request Resolution System

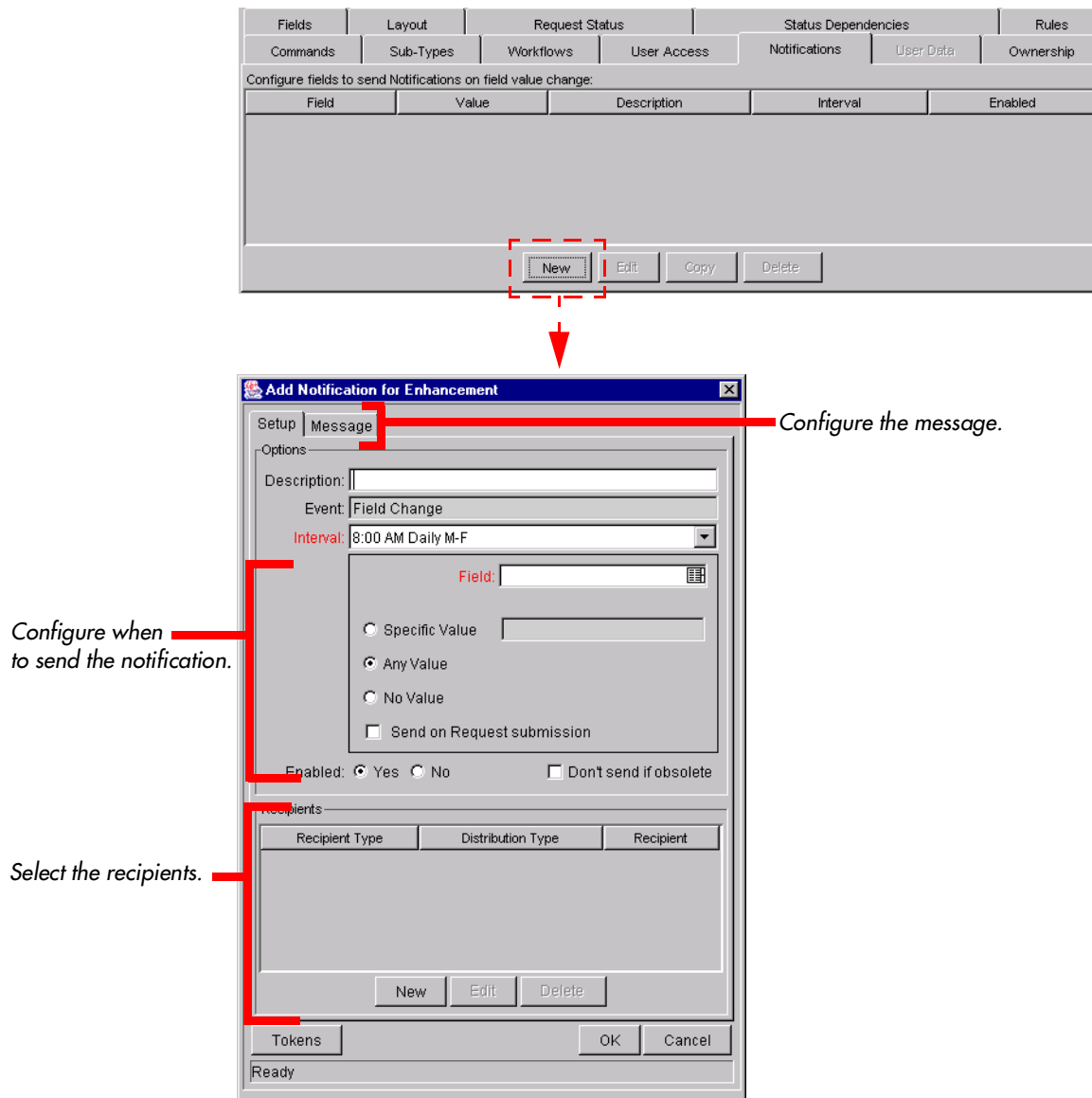


Figure 9-2 Request Type - Notifications Tab and Add Notification Window

The ADD NOTIFICATION window for Request Types is nearly identical to the ADD NOTIFICATION FOR STEP window for Workflow Steps. The only difference lies in the fields that configure when to send the Notification, described in [Table 9-9](#).

Table 9-9. Fields Controlling Request Type Field Notification

Field	Description
EVENT	Always set to <b>FIELD CHANGE</b> . Cannot be edited.



Table 9-9. Fields Controlling Request Type Field Notification

Field	Description
INTERVAL	<p>You can select to send the Notification at different intervals. For example, you might choose to send a notification of a field change at midnight so that it's ready to be reviewed in the morning.</p> <p>Note also that multiple notifications to a single recipient can be batched and sent together. Selecting an interval other than <b>IMMEDIATE</b> will allow this batching to occur.</p> <p>See "<a href="#">Configuring the Notification Intervals</a>" on page 193 for instructions on configuring this.</p>
FIELD	Specifies the field that will trigger the Notification.
SPECIFIC VALUE	You can choose to send the Notification when the chosen FIELD has a specific value. You must specify a FIELD first for this to have any meaning.
ANY VALUE	You can choose to send the Notification when the chosen FIELD has any value at all.
NO VALUE	You can choose to send the Notification when the chosen FIELD has no value.
SEND ON REQUEST SUBMISSION	Checking this box means the Notification will be send only when the Request is submitted and the chosen FIELD matches the value specified.
ENABLED	Turns the Notification on or off.
DON'T SEND IF OBSOLETE	Checking this box means that when the INTERVAL is not set to <b>IMMEDIATE</b> , Kintana will check if the chosen FIELD matches the value specified before sending the Notification.

To add a notification to a Request Type:

1. Open the Request Type.
2. Click the **NOTIFICATIONS** tab.
3. Click **NEW**. The ADD NOTIFICATION window opens.

The screenshot shows the 'Add Notification for Enhancement' dialog box. The 'Setup' tab is selected. The 'Options' section includes a 'Description' field, an 'Event' dropdown set to 'Field Change', and an 'Interval' dropdown set to '8:00 AM Daily M-F'. Below these is a 'Field' field with a list icon, and three radio button options: 'Specific Value', 'Any Value' (selected), and 'No Value'. There is also a checkbox for 'Send on Request submission'. At the bottom of the 'Options' section are 'Enabled' radio buttons for 'Yes' (selected) and 'No', and a checkbox for 'Don't send if obsolete'. Below the 'Options' section is a 'Recipients' section with a table with columns 'Recipient Type', 'Distribution Type', and 'Recipient'. The table is currently empty. Below the table are 'New', 'Edit', and 'Delete' buttons. At the bottom of the dialog are 'Tokens', 'OK', and 'Cancel' buttons. The status bar at the very bottom says 'Ready'.

4. Configure the following:
  - When the notification is sent (FIELD and VALUE choice)
  - Who receives the notification (RECIPIENTS)
  - The body of the notification (MESSAGE)
5. Click **OK**. The notification specification is added to the **NOTIFICATIONS** tab. You can add additional specifications to the step by clicking **NEW** and repeating the above process. You can therefore select to send a different notification to different recipients for different events.
6. Click **OK** to save and close the window.

## Configuring Your Dashboard

The Dashboard provides an interface through which you can quickly assess the current state of Requests. Personalize your Dashboard to display status information that is most meaningful to your role. For example, Financial system users may only want to see the Requests that they submitted, whereas the Financial manager may want to have visibility into each critical Request currently in progress.

Each user can personalize their own Dashboard to display only information relevant to their role. When configuring your Request resolution system, you need to consider the following configuration topics:

- [Controlling User Access to Portlets](#)
- [Creating and Distributing a Default Dashboard](#)
- [Creating Custom Portlets](#)

### Related Documents:

- ["Using the Kintana Dashboard"](#)
- ["Configuring the Kintana Dashboard"](#)
- ["Kintana Security Model"](#)



Note

Users are required to have a Dashboard license to add portlets to their Dashboard.

## Controlling User Access to Portlets

You can control portlet user access at two levels:

- [Disabling Portlets](#)
- [Restricting User Access](#)

### Disabling Portlets

You can disable custom-built portlets at your site. To disable a portlet:

1. Click the **DASHBOARD** screen group and click the **PORTLETS** icon.
2. Search for and open the custom Portlet that you would like to disable.

Note that you can not disable Kintana system portlets. To control access to these portlets, you can restrict user access. See [Restricting User Access](#) for details.

The screenshot shows a configuration window for a portlet titled "Finance - My Expenses for Last 2 Periods". The window includes the following fields and controls:

- Portlet Name:** Finance - My Expenses f
- Product Scope:** Kintana Deliver
- Default Title:** Finance - My Expenses f
- Portlet Category:** (empty)
- Default Max Rows Displayed:** 10
- Portlet Width:** Wide
- Description:** (empty)
- Enabled:** Radio buttons for Yes (selected) and No.
- Time-Out:** Use Default (dropdown), 20 (input), Seconds.
- Currently Used By:** 47 User(s)
- User Access:** A section with a checked checkbox "Allow all users to add this portlet to their dashboard".
- Security:** A table with columns "Security Type" and "Security".
- Remove:** A button to remove the portlet.
- Security Group:** A text input field with an "Add Security Group(s)" button.
- User:** A text input field with an "Add User(s)" button.
- Buttons:** Verify, OK, Save, and Cancel.

System portlets only allow editing of user access, and cannot be copied or deleted.

3. Click **ENABLED = No**.



Note

If there are any users currently using the portlet on their Dashboard, disabling the portlet will delete it from their Dashboards.

4. Click **SAVE**.

### Related Topics:

- ["Using the Kintana Dashboard"](#)
- ["Configuring the Kintana Dashboard"](#)

## Restricting User Access

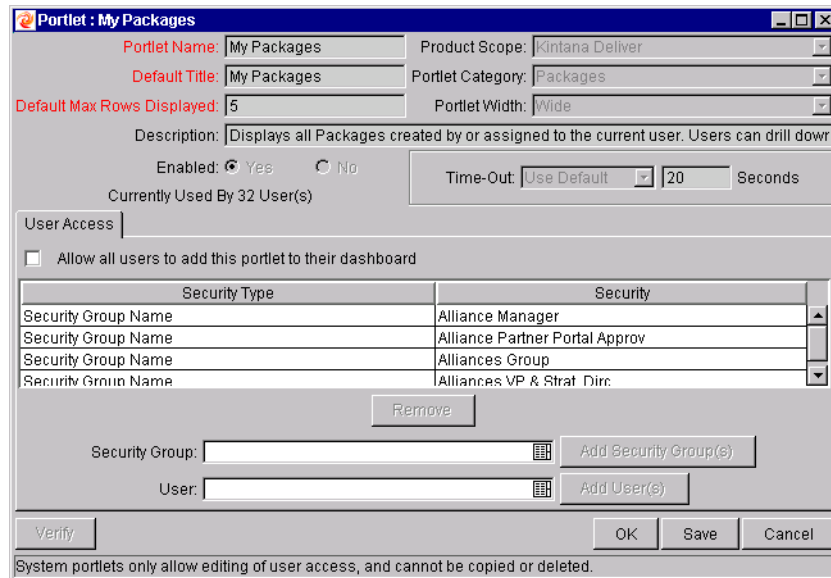
You can control which users can add a portlet to their Dashboard. For example, you may want to restrict the Request-related portlets to only members involved in resolution processes. Enabling only the portlets that a specific user needs will make it easier for that user to personalize their Dashboard, because there are less (non-relevant) portlets to choose from.

To specify which users can use the portlet on their Dashboard:

1. Click the **DASHBOARD** screen group and click the **PORTLETS** icon.
2. Search for and open the Portlet that you would like to configure.
3. Click the **USER ACCESS** tab. For system portlets (such as My Packages), the **USER ACCESS** tab is the only displayed tab.

The screenshot shows the configuration window for the 'My Packages' portlet. The 'User Access' tab is active, and the checkbox 'Allow all users to add this portlet to their dashboard' is checked. Below this, there are fields for 'Security Type' and 'Security'. A 'Remove' button is present. At the bottom, there are fields for 'Security Group' and 'User', each with an 'Add' button. The 'Time-Out' is set to 20 seconds. The status bar at the bottom indicates 'System portlets only allow editing of user access, and cannot be copied or deleted.'

4. Un-check the **ALLOW ALL USERS TO ADD THIS PORTLET TO THEIR DASHBOARD** field. The **SECURITY GROUP** and **USER** fields are enabled.
5. Select the desired **Security Groups** or **Users** and click the respective **ADD** button. They are added to the **USER ACCESS** tab.



Security Type	Security
Security Group Name	Alliance Manager
Security Group Name	Alliance Partner Portal Approv
Security Group Name	Alliances Group
Security Group Name	Alliances VP & Strat Dir:

### 6. Click **SAVE**.

You can restrict access by specifying multiple Security Groups and Users for each portlet. Only members of the specified Security Group or the specified users can add this portlet to their Dashboard.



Note

You can restrict user access for both custom and system portlets.

## Creating and Distributing a Default Dashboard

You can configure a Default Home page that all Kintana users will see when they log into Kintana for the first time. This saves time and allows first-time users to more quickly and easily integrate the Dashboard into their business processes. The Default Dashboard is configured using the Kintana HTML interface. See "[Configuring the Kintana Dashboard](#)" for detailed instructions.



Note

Users must have the **EDIT DEFAULT USER HOMEPAGE** Access Grant to configure the default Dashboard.

## Creating Custom Portlets

Portlets are visual displays that act as windows into different aspects of Kintana data. While Kintana's system portlets (provided at the time of installation) are personalizable by end-users and provide wide access to your Kintana data, you can also create custom portlets to access additional information in Kintana or in other databases. These custom portlets behave the same as the system portlets, using filter fields to limit the displayed data. You can create textual or graphical portlets.

Because custom portlets are data-driven entities and require extracting information stored in the database, knowledge of SQL is required for users who wish to create or configure portlets.

See the "[Configuring the Kintana Dashboard](#)" for detailed instructions on creating custom portlets.



Note

Before creating custom portlets, consider using one of Kintana's system portlets. Review the business and data presentation portlet requirements and compare against the following list of system portlets.

## Configuring Reports

Kintana features a pre-defined set of HTML-based reports that are accessed through a Web browser. The reports allow users to view the current detailed status of their Kintana data at any point in time. Kintana's Decision Support System (DSS) reports provide users with a high level overview of their initiatives through graphical summary reports.

Kintana also provides a Reporting Meta Layer, which allows users to build their own custom reports using third-party reporting tools.

See "[Kintana Reports](#)" for a full list of available reports and information on configuring them.





## Appendix



## Advanced Workflow Topics

Workflows are discussed in the Kintana Business Application Guides as they relate to business processes. This chapter provides additional instructions for advanced Workflow configurations. It is organized into the following sections:

- *Using Subworkflows*
- *Package - Request Workflow Integration*
- *Using Condition Steps*
- *Setting the Reopen Step for Request Workflows*
- *Modifying Workflows in Use*
- *Using Workflow Parameters*

### Using Subworkflows

A Kintana Subworkflow is any Workflow that is referenced from within another Workflow. Subworkflows allow you to model complex business processes into logical, more manageable and reusable sub-processes.

Subworkflows are defined in the same manner as typical Kintana Workflows. Two things to keep in mind when working with Subworkflows:

- The WORKFLOW window contains a SUBWORKFLOW 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.



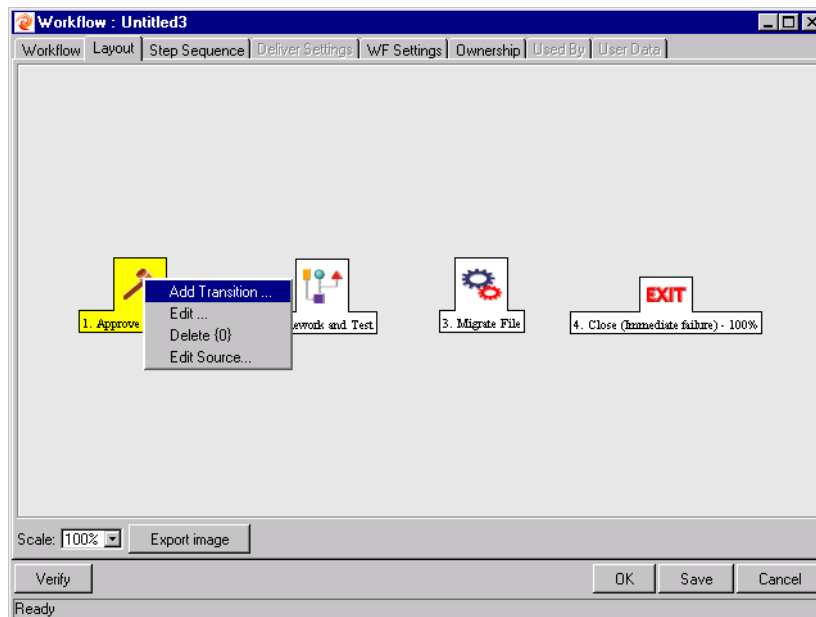
Subworkflows can also be generated by copying and renaming an existing Subworkflow.

A Subworkflow can be selected from the **WORKFLOW STEP SOURCES** window and dragged onto the **LAYOUT** tab. When the Package, Request, or Release Distribution 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.

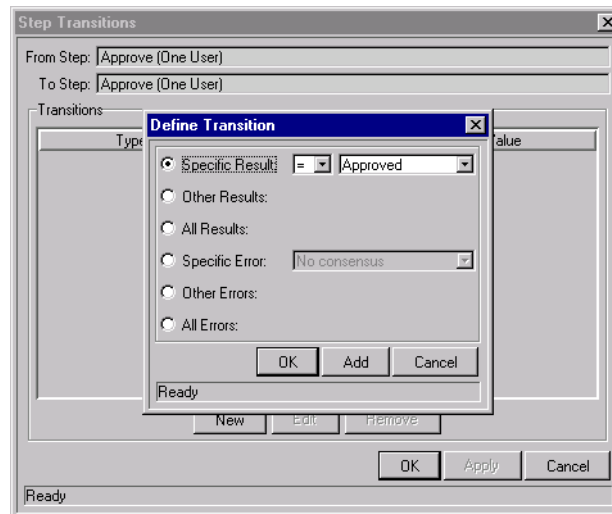
## Transitioning to a Subworkflow

A transition to a Subworkflow Step is made in the same way as a transition to any other Workflow Step (Execution, Decision or Condition):

1. Open a Workflow and click the **LAYOUT** tab.
2. Right-click the source Workflow Step and select **ADD TRANSITION** from the pop-up menu. This creates an arrow from the source Workflow Step.



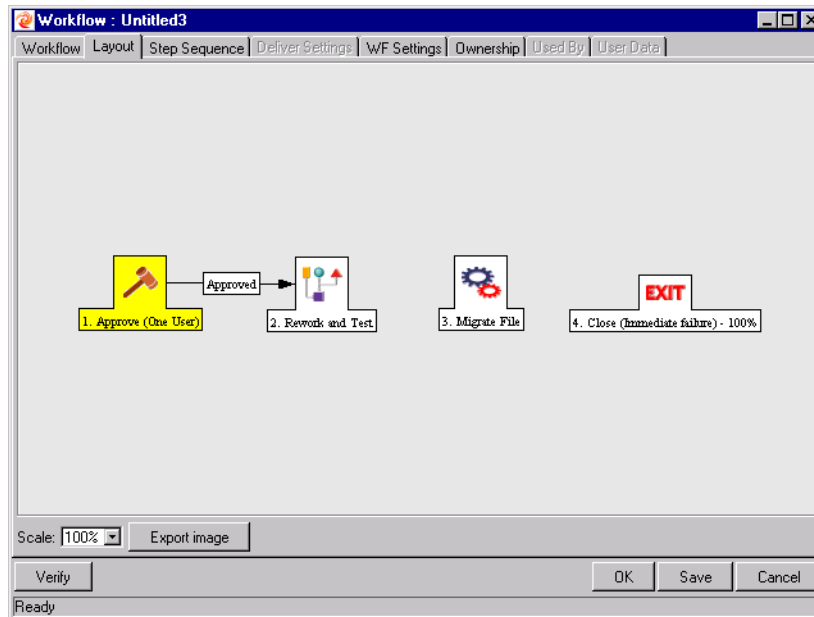
3. Drag the arrow to the destination Step and left-click. The **STEP TRANSITIONS** window opens.
4. Click **NEW**. The **DEFINE TRANSITION** window opens.



5. Define the desired transaction result by:
  - a. Clicking one of the radio buttons for results (SPECIFIC RESULT, OTHER RESULTS, etc.).
  - b. Selecting = or != (does not equal) from the drop down list.
  - c. Selecting **YES** or **NO** from the drop down list.

If the Workflow Step results in this value, the Request or Package proceeds along this path.

6. Click **OK** to close the DEFINE TRANSITION window.
7. Click **OK** to finalize the step transition definition.

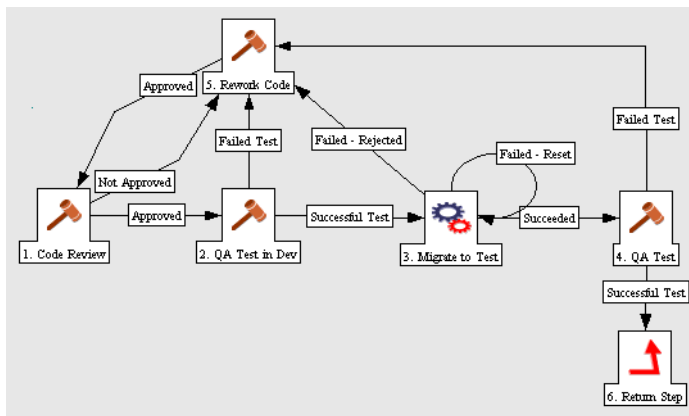
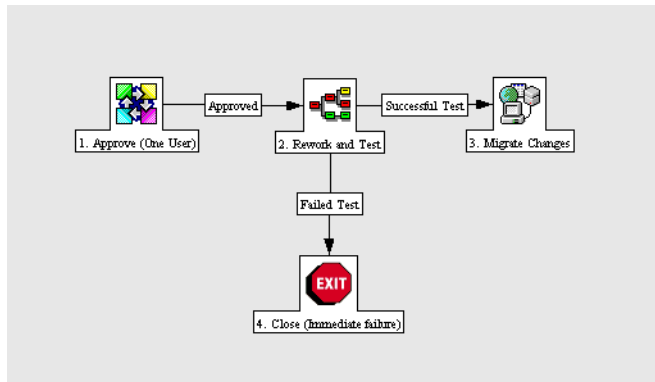


The transition is graphically represented by an arrow between the two steps. The Package Line or Request proceeds to the First Step designated in the Subworkflow definition.

## Transitioning From a Subworkflow

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.



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.



Note

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.

## Package - Request Workflow Integration

Kintana Request and Package Workflows 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.



Note

Kintana also supports another level of Request - Package integration that does not rely on the Workflow configuration. You can attach Packages and Requests to each entity as References. You can then set dependencies 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 that the Package will not continue until the Request closes.

Kintana has provided two built-in Workflow Events available in the Workflow Workbench to facilitate the cross-product Workflow integration. These steps are **WF\_JUMP** and **WF\_RECEIVE**. Jump (**WF\_JUMP**) and Receive (**WF\_RECEIVE**) steps can be created that define the points of interaction between Workflows. Each Jump step must be coupled with a Receive step. Workflows can communicate through these Jump and Receive 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 in Deliver.
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. Because the step has succeeded, the process can now migrate the code changes to the Production Environment.

This process is graphically represented in [Figure A-1](#).

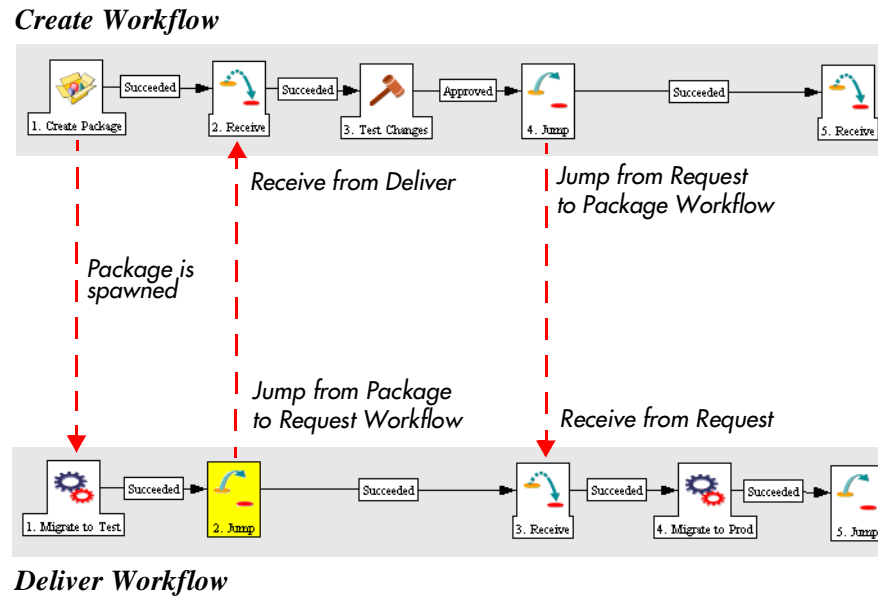


Figure A-1 Jump/Receive Workflow Steps

The Jump and Receive pair must be carefully coordinated. Each JUMP step must have an associated RECEIVE step, linked together by a common JUMP/RECEIVE STEP LABEL defined in the WORKFLOW STEP window. The transition values for entering into and exiting the JUMP and RECEIVE steps must also be coordinated.

This section details the process for setting up a successful Request - Package Workflow integration. To establish communication between Requests and Packages:

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 step. The selected JUMP/RECEIVE STEP LABEL must match in the paired Jump and Receive Workflow Step windows. See [“Setting Up the ‘WF - Jump/Receive Step Labels’ Validation”](#) on page 220.
2. Create a JUMP step using the **WF\_JUMP** BUILT-IN WORKFLOW EVENT. See [“Generating a Jump Step Source”](#) on page 222.
3. Create a RECEIVE step using the **WF\_RECEIVE** BUILT IN WORKFLOW EVENT. See [“Generating a Receive Step Source”](#) on page 223.
4. Verify that both the JUMP and RECEIVE steps specify the same JUMP/RECEIVE STEP LABEL. See [“Including the Jump/Receive pair in Workflows”](#) on page 225.

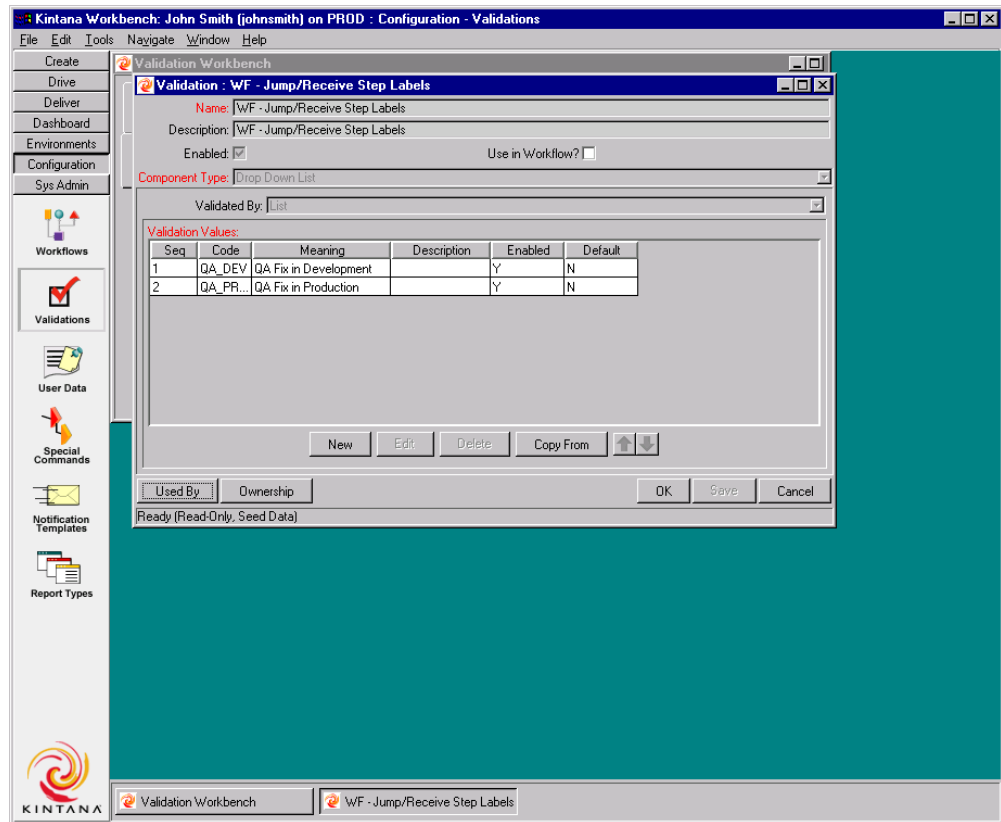
5. Verify that the transitions exiting the JUMP and RECEIVE steps match the possible values entering the JUMP step.

### *Setting Up the 'WF - Jump/Receive Step Labels' Validation*

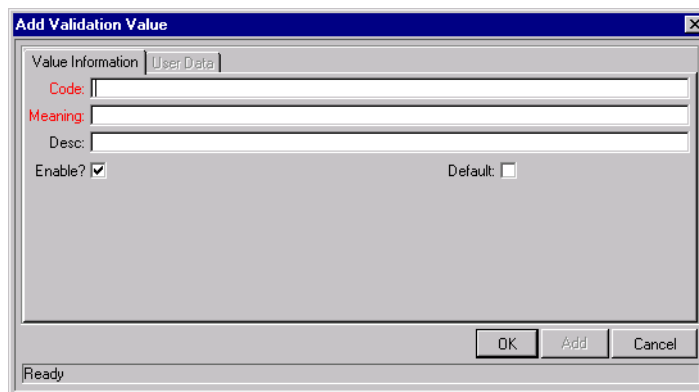
To set up the WF - JUMP/RECEIVE STEP LABELS Validation:

1. Click the **CONFIGURATION** screen group and click the **VALIDATIONS** icon. The **VALIDATION WORKBENCH** opens.
2. On the **QUERY** tab, enter **WF - JUMP/RECEIVE STEP LABELS** in the **VALIDATION NAME** field.
3. Click **LIST**.
4. Click the **RESULTS** tab. The WF - JUMP/RECEIVE STEP LABELS is listed in the **RESULTS** tab.
5. Click **OPEN**. The **VALIDATION** window opens.





6. Click **NEW** to define a new Validation Value that is used to link two Workflows together. The ADD VALIDATION VALUE window opens.



7. Enter the desired CODE, MEANING and DESCRIPTION in the appropriate fields.
8. Click **OK** to close the ADD VALIDATION VALUE window.
9. Click **OWNERSHIP** to select which Ownership Groups will have the ability to edit this Validation.

10. Click **OK** to close the VALIDATION window.

The new Validation Value is now included in the JUMP/RECEIVE STEP LABEL drop down list in the WORKFLOW STEP window.

### Generating a Jump Step Source

To create a Jump step using the **WF\_JUMP** BUILT-IN WORKFLOW EVENT:

1. Click the **CONFIGURATION** screen group and click the **WORKFLOWS** icon. The WORKFLOW WORKBENCH and WORKFLOW STEP SOURCES window open.
2. Select the WORKFLOW STEP SOURCES window.
3. Select the EXECUTIONS folder.
4. Click **NEW**. The EXECUTION window opens.

5. Select either **PACKAGES** or **REQUESTS** from the WORKFLOW SCOPE drop down list, depending on the desired application of the Workflow. Package Level Subworkflows can not include jump and receive steps.
6. Select **BUILT-IN WORKFLOW EVENT** from the EXECUTION TYPE drop down list.

7. Select **WF\_JUMP** from the **WORKFLOW EVENT** drop down list.
8. Select or create a **Validation** from the **VALIDATION** drop down list which will be used to transition out of this **Workflow Step**.

Note

The **Validation** values exiting the **Jump** step must match the possible **Validation** values entering the **Jump** step.

9. Fill in any other required or optional information in the **EXECUTION** window (such as **NAME**, **DESCRIPTION** or **PROCESSING TYPE**).
10. Click the **OWNERSHIP** tab to select which **Ownership Groups** will have the ability to edit this **Execution** step.
11. Click **OK**. The **Workflow Step** is added to the **WORKFLOW STEP SOURCES** window.

This **Workflow Step** can now be used in any new or existing **Workflow** within the step's defined **Workflow Scope**. Remember that every **JUMP** step must have a paired **RECEIVE** step in another **Workflow**.

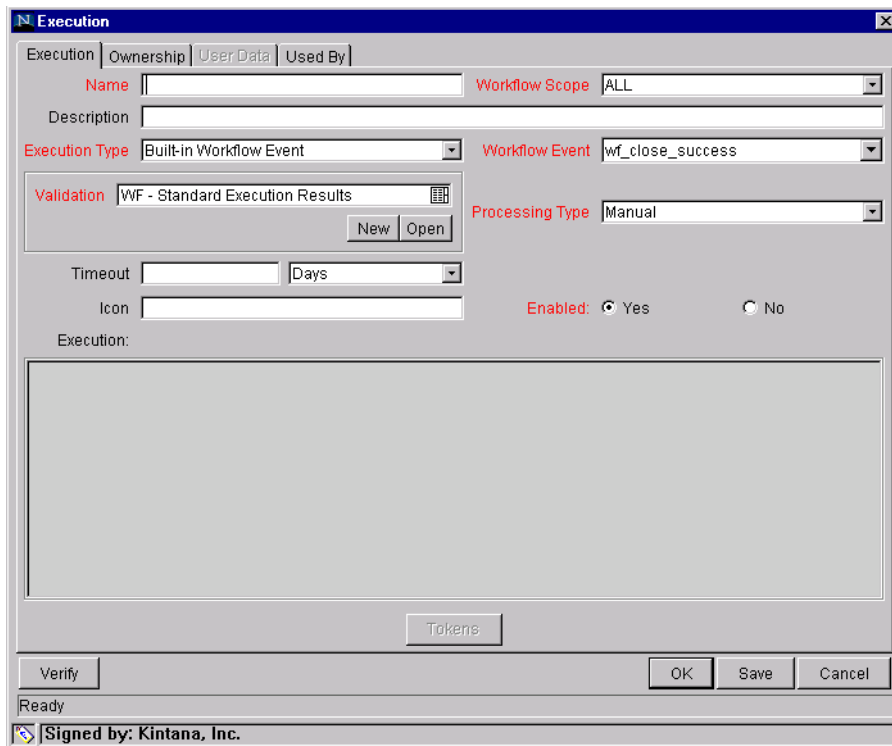
### *Generating a Receive Step Source*

To create a **RECEIVE** step using the **WF\_RECEIVE BUILT-IN WORKFLOW EVENT**:

1. Click the **CONFIGURATION** screen group and click the **WORKFLOWS** icon. The **WORKFLOW WORKBENCH** and **WORKFLOW STEP SOURCES** window open.
2. Select the **WORKFLOW STEP SOURCES** window.



3. Select the EXECUTIONS folder.
4. Click **NEW**. The EXECUTION window opens.



5. Select either **PACKAGES** or **REQUESTS** from the WORKFLOW SCOPE drop down list, depending on the desired application of the Workflow.

6. Select **BUILT-IN WORKFLOW EVENT** from the EXECUTION TYPE drop down list.
7. Select **WF\_RECEIVE** from the WORKFLOW EVENT drop down list.
8. Select or create a Validation which will be used to transition out of this Workflow Step.



Note

The Validation values exiting the RECEIVE step must match the possible Validation values entering and exiting the JUMP step.

9. Fill in any other required or optional information (such as NAME, DESCRIPTION OR PROCESSING TYPE).
10. Click the **OWNERSHIP** tab to select which Ownership Groups will have the ability to edit this Execution step.
11. Click **OK**. The Workflow Step is added to the WORKFLOW STEP SOURCES window.

This Workflow Step can now be used in any new or existing Workflow within the step's defined Workflow Scope. Remember that every RECEIVE step must have a paired JUMP step in another Workflow.

### *Including the Jump/Receive pair in Workflows*

1. Drag either the JUMP or RECEIVE step from the WORKFLOW STEP SOURCES window into the Workflow's **LAYOUT** tab. The WORKFLOW STEP window opens.

The screenshot shows the 'Workflow Step' dialog box with the following fields and values:

- Step Number: 1
- Step Name: Create Package Immediate
- Action Button Label: (empty)
- Description: (empty)
- Source Type: Execution
- Source Name: Create Package Immediate
- Enabled:  Yes  No
- Display: Always (dropdown)
- Jump/Receive Step Label: QA Fix in Development (dropdown)
- Workflow Parameter: NONE (dropdown)
- Source Environment: (empty)
- Source Environment Group: (empty)
- Dest Environment: (empty)
- Dest Environment Group: (empty)
- Save to O\*M/GL\*M Archive?  Yes  No
- Avg Lead Time: (empty)
- Request Status: (empty)
- Current % Complete: (empty)
- Parent Assigned To User: (empty)
- Parent Assigned To Group: (empty)
- Workflow Step Information: (empty)

Buttons: OK, Apply, Cancel

Status: Ready

2. Select an item from the JUMP/RECEIVE STEP LABEL drop down list. This item must be the same for a paired Jump/Receive Step.



Note

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.

3. Enter any additional Workflow Step information.
4. Click **OK**.
5. Repeat the above process for the other paired Workflow Step (Jump or Receive), depending on which one was configured first

## Using Condition Steps

Kintana can perform complex routing based on the status of multiple Workflow Steps using Condition steps. There are five Condition steps available:

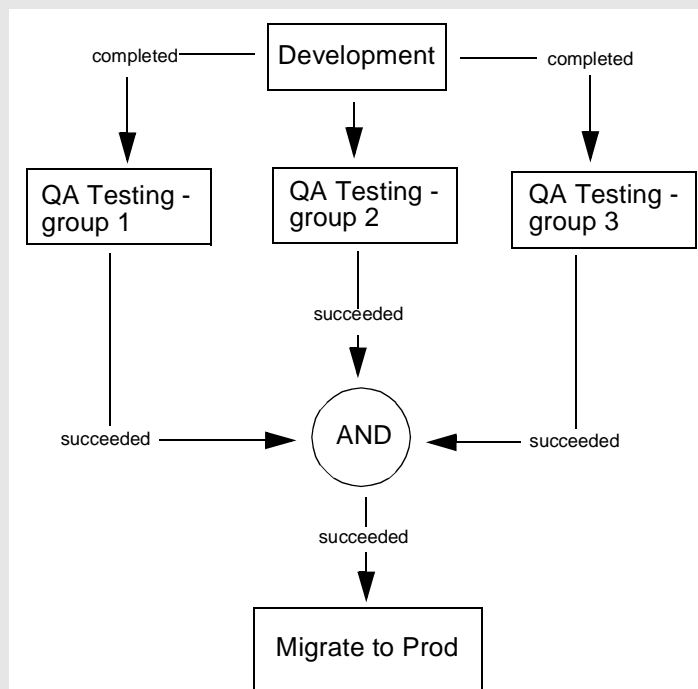
- *AND*
- *OR*
- *SYNC*
- *FIRST LINE*
- *LAST LINE*

### AND

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



The AND step becomes successful only if 'QA Testing - group 1,' 'QA Testing - group 2' and 'QA Testing - group 3' are successful. At that point, the following step 'Migrate to Prod' becomes eligible.

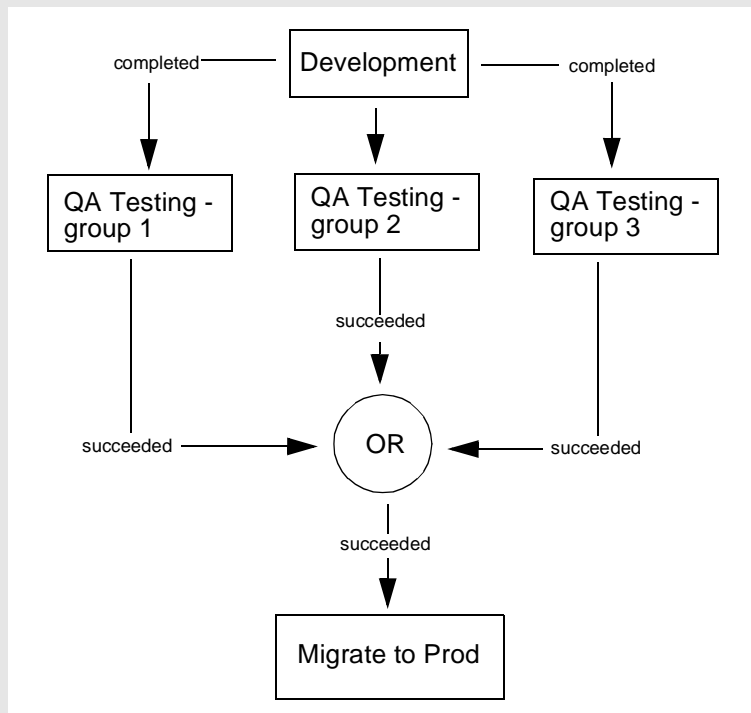


## OR

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



The OR step becomes successful if any one of 'QA Testing - group 1,' 'QA Testing - group 2' and 'QA Testing - group 3' is successful. At that point, the following step 'Migrate to Prod' becomes eligible.



## SYNC

A SYNC step is valid only for Kintana Packages. A SYNC step is successful only if all the Package Lines of that Package reach the status expected for the Workflow Step right before the SYNC step.

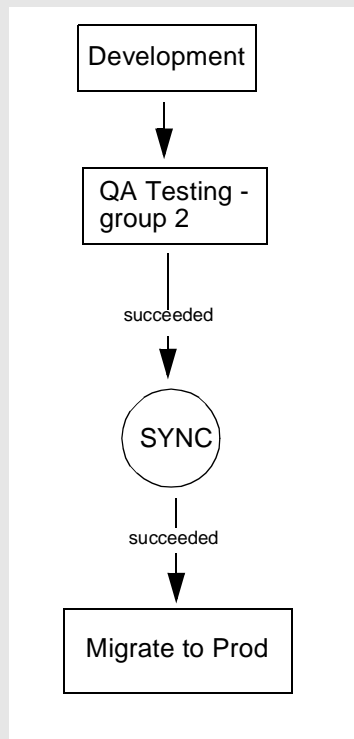




Example

Consider the business process outlined in the following flow chart. According to the flow chart, when 'QA Testing' is successful for all Package Lines, SYNC becomes successful and the next step, 'Migrate to Prod' becomes eligible.

This business process could be part of a software development life cycle. Consider a case where three Java files are being processed on three respective Package Lines in a single Package. By including a SYNC step, even if the first two Java files pass 'QA Testing,' they must wait for the third Java file to succeed 'QA Testing' before 'Migrate to Prod' becomes eligible for any of these Package Lines.



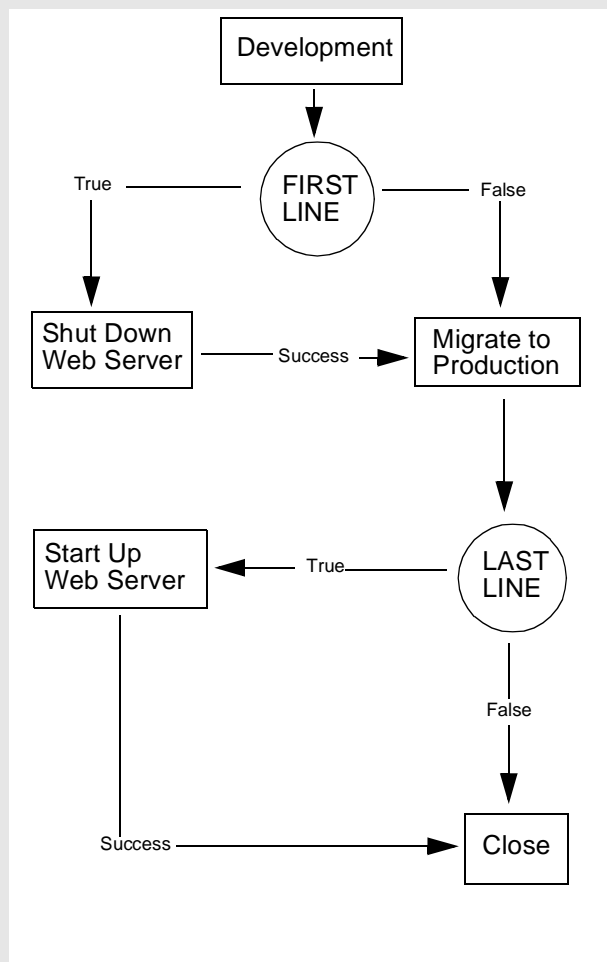
### FIRST LINE

A FIRST LINE step is valid only for Kintana Packages. Only the first line to reach the Condition step takes the 'True' transition. All successive lines take the 'False' transition.

Example

Consider the business process outlined by the following flow chart. This business process could be part of a Website maintenance life cycle. As part of this life cycle, three HTML files are being processed on three respective Package Lines in a single Package. The Website updates are large enough to warrant shutting down the Web server while migrating the changes.

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



## LAST LINE

A LAST LINE step is valid only for Kintana Deliver. Only the last active line to reach the Condition step takes the 'True' transition. All previous lines take the 'False' transition. See the example of a LAST LINE step shown in the previous flow chart.

## Setting the Reopen Step for Request Workflows

Closed Requests can be re-opened by users with the proper access grants. A re-opened Request begins at the pre-defined Reopen Step in its Workflow, and begins processing normally.

The Reopen Step is defined from the WORKFLOW window.

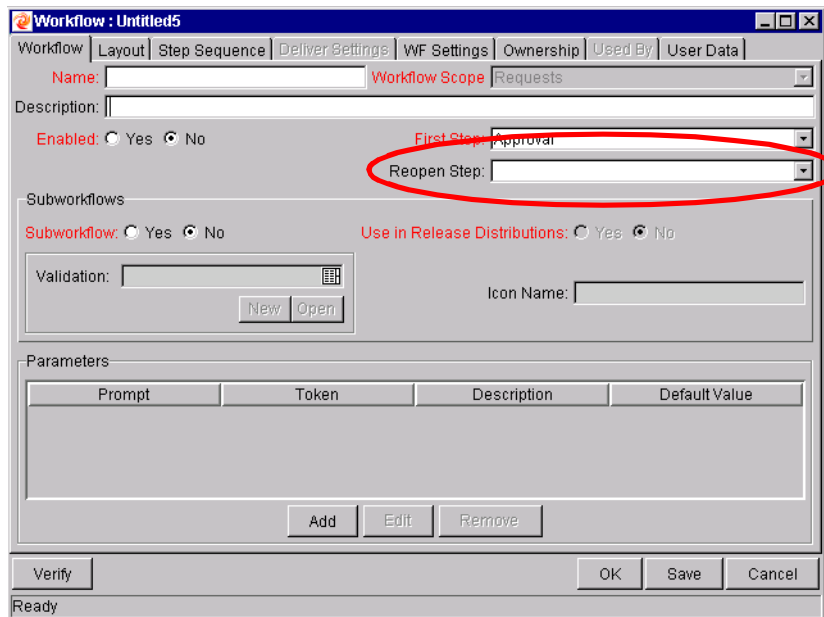


Figure 0-1 Workflow Window Reopen Step Drop Down

To specify the Reopen Step for the Workflow, select the desired step from the REOPEN STEP drop down list.

## Modifying Workflows in Use

Kintana Workflows can be modified while they are going through their Workflow steps in a Package Line or Request that has been initiated. These modifications include adding new Workflow Steps, as well as changing the transitions, security assignments and notifications from within the Workflow.

You 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.

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 0-1](#).

*Table 0-1. 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.



Note

When a Workflow that is in use is modified and saved, the changes take effect in Kintana immediately. Any changes made to Workflow Steps are applied to all open Package Lines, Requests, and Distributions.

Changes to a Workflow can have undesirable effects on Requests or Packages currently in progress and are using that Workflow.



The information included here also applies when migrating Workflows between installations (instances) of Kintana.

When you modify 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, you might remove a transition from a Workflow Step and find that the Requests or Package Lines are stuck in that Step. While no one solution covers all situations, the following sections describe possible solutions for common problems when modifying Workflows:

- [\*Copying and Testing the Workflow\*](#)
- [\*Moving Requests Out of a Step\*](#)
- [\*Disabling a Workflow Step\*](#)
- [\*Setting Up Execution Steps\*](#)
- [\*Verifying Workflow Logic\*](#)

### *Copying and Testing the Workflow*

To modify a Workflow that is being used, make a copy of the original Workflow in a Development environment. Then modify the copied version of the Workflow. Test the copied version of the Workflow to make sure it works correctly.

After verifying that the modified Workflow functions as it is supposed to, make the same changes to the original Workflow and move it through the same cycle of DEVELOPMENT -> TEST -> PRODUCTION environments.

### *Moving Requests Out of a Step*

If your Requests are stuck in a step after you remove a transition from a Workflow in use, add the deleted transition back to the Workflow. After the Requests have flowed out of the step, delete the transition again.

To determine when the Requests have flowed out of the step, run the WORKFLOW DETAIL REPORT. This report indicates if the step you want 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.

## Disabling a Workflow Step

As mentioned in [Table 0-1](#), you cannot delete a step from a Workflow that is in use; you can only disable it. However, you may want to change the process that is routed through the Workflow. Any changes to the process must be reflected in the Workflow. This will require disabling existing steps and adding new steps.

You can effectively disable a step that you no longer want to use and add a new step by following this process:

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 the Workflow

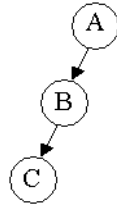
If you disable a Workflow step that is currently ‘Eligible’ for user action, the Requests or Package Lines in that step will become “stuck”. Since the step is now disabled, the user cannot take action on it and will not be able to progress any further through the Workflow.

To determine which steps are currently Eligible, remove the incoming transition to the step you want to delete and then run the Packages Pending Report in Deliver or the Workflow Detail Report in Create. The reports will indicate if the step you want to delete is ‘Eligible’ for action by Package Lines or Requests.

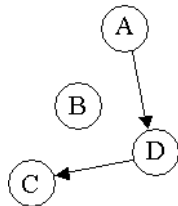
The outgoing transition to be deleted is still intact, so the eligible Package Lines and Requests will eventually be acted upon and flow out of the Workflow step.

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 below.



After removing the incoming and outgoing transitions to B, you would 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 the sequence shown below.



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.

### *Setting Up Execution Steps*

When setting up Execution steps in a Workflow Step, be sure to include Workflow Events 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.

### *Modifying Workflow Step Security – Performance Consideration*

Updating an existing Workflow's step security with a specific configuration can impact system performance. If you add dynamic security to a step (i.e. based on a Standard or User Defined Token) in the **WORKFLOW STEP** window on the **LAYOUT** tab, tables in the Kintana database are updated to handle this new configuration. Because of the scope of database changes, you should re-run the Database Statistics on your Kintana Database. Instructions for this are included in the "Kintana System Administration Guide." Contact your System Administrator for help with this procedure.



This also applies if you migrate a Workflow with these types of changes into an instance of Kintana.

### *Verifying Workflow Logic*

A Workflow can also become stuck if the logic behind it is faulty. Plan the steps of your Workflow process carefully before actually defining it. After configuring your Workflow, click the **VERIFY** button in the Workflow window to ensure that the logic of your Workflow is correct. Any mistakes in the Workflow's logic will be highlighted.

## Using Workflow Parameters

You can use Workflow parameters to store the result of a workflow step. This value can then be used later to define a transition.



#### Workflow Parameters:

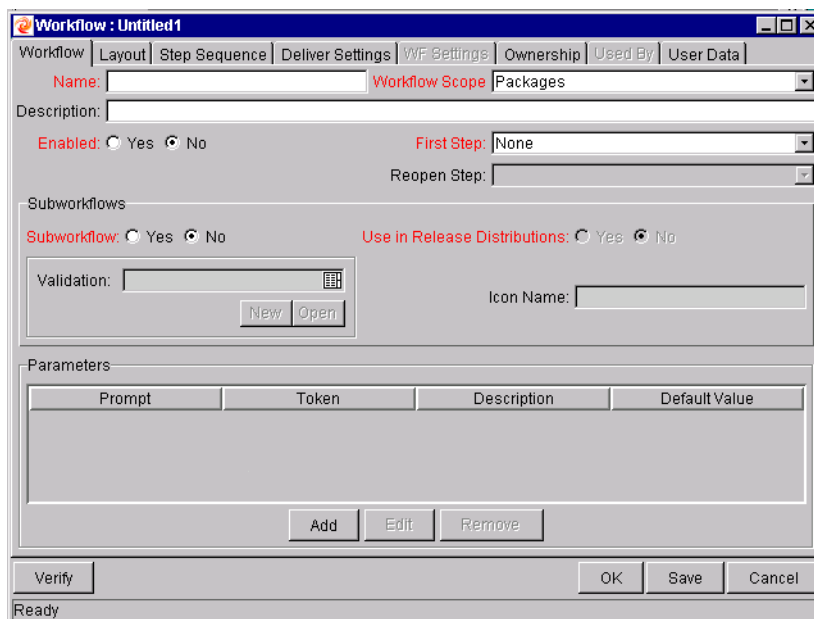
- Can be referenced using the WFI.P Token prefix.
- Can be used in PL/SQL and SQL Workflow Step executions.

### *Creating a Workflow Parameter*

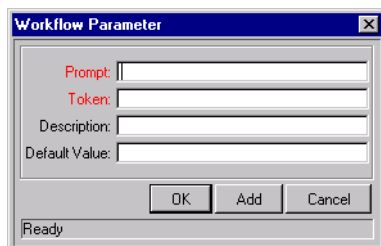
To create a Workflow parameter:

1. From the WORKFLOW WORKBENCH, query and open the Workflow to be modified.





2. In the **WORKFLOW** tab, click **ADD**. The **WORKFLOW PARAMETER** window opens.

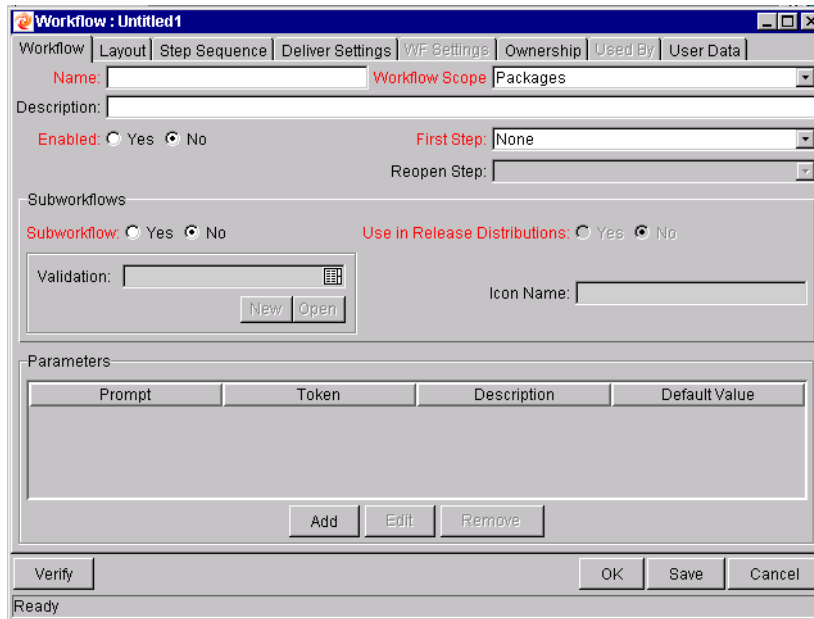


3. Enter information in the required fields.
4. Click **OK**.

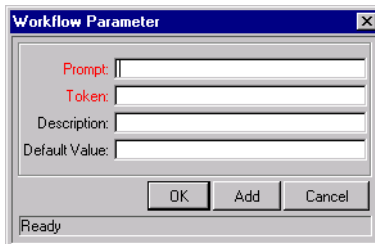
### *Example: Building a Loop Counter*

Workflow parameters can be used to generate a counter for the number of times a Workflow Step is in a certain state. To build a loop counter using Workflow parameters:

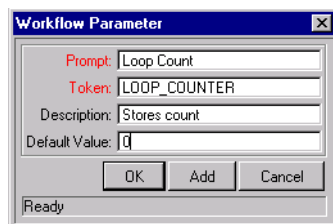
1. From the **WORKFLOW WORKBENCH**, open the Workflow to which the loop counter is to be added.



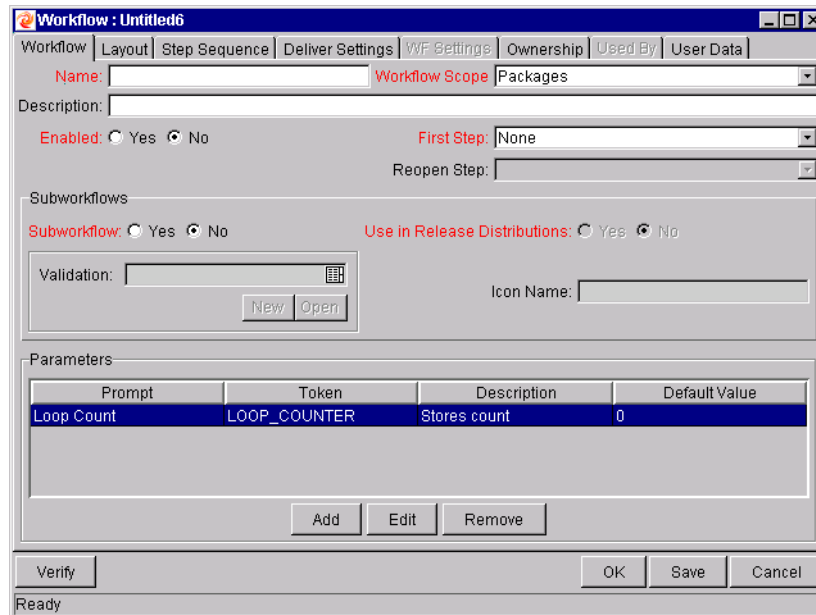
2. In the **WORKFLOW** tab, click **ADD**. The **WORKFLOW PARAMETER** window opens.



3. Generate the Workflow parameter by entering information in the fields of the **WORKFLOW PARAMETER** window. In this example, the parameter is named **LOOP\_COUNTER**.



4. Click **OK**. The **LOOP COUNT** parameter is added to the Workflow window.



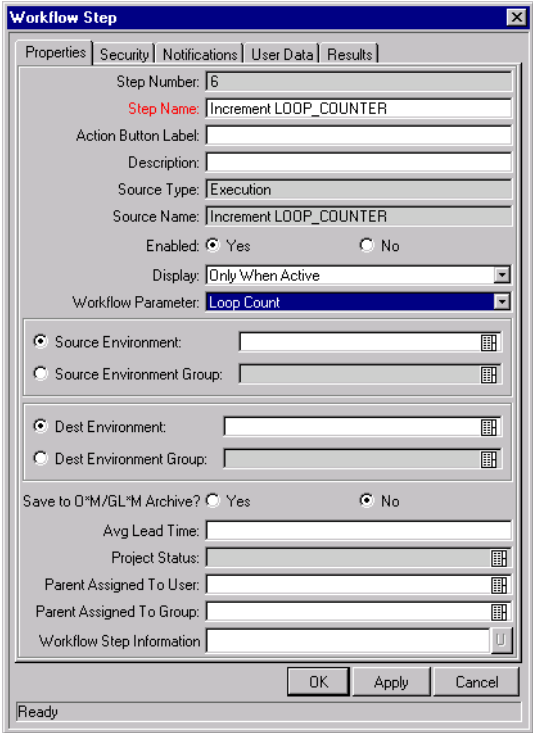
5. Generate a new Immediate SQL Execution Step. There are two key concepts to note about the new step definition.
  - The result of the SQL Execution step returns the result **LOOP\_COUNTER + 1**. This return value is linked back into the parameter when the Workflow Step is generated on a Workflow.
  - A Validation for a **NUMERIC** Text Field is used. This allows  $\leq$ ,  $<$ ,  $\geq$ , and  $>$  comparisons to be used in transitions off this step.

The screenshot shows the 'Execution' configuration window. The 'Name' field is 'Increment Loop Counter' and the 'Workflow Scope' is 'Packages'. The 'Description' is 'Increments count by 1.'. The 'Execution Type' is 'SQL Statement' and the 'Workflow Event' is 'None'. The 'Validation' is set to 'Numeric Text Field'. The 'Processing Type' is 'Immediate'. The 'Enabled' checkbox is checked. The 'Execution' section contains the following SQL statement:

```
Select [WF.F.P.LOOP_COUNTER]+1
from dual
```

At the bottom of the dialog, there are buttons for 'Verify', 'OK', 'Save', and 'Cancel'. A status bar at the very bottom of the window reads 'Signed by: Kintana, Inc.'

6. 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.



Example

The loop counter can be incremented each time a Kintana Deliver 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.



# Appendix B Validations

This chapter provides an overview for how to use Validations in your Kintana system. Validations determine the acceptable input values for user-defined fields (such as Object Type or Request Type fields). Validations also determine the possible results that a Workflow step can return. This appendix discusses the following topics:

- *What are Validations*
- *Validation Component Types - Overview*
- *Creating a Validation*
- *Editing Validations*
- *Deleting Validations*
- *Static List Validations*
- *Dynamic List Validations*
- *Using Auto-Complete Validations*
- *Using Directory and File Choosers*
- *Creating 1800 Character Text Areas*
- *Configuring the Table Component*
- *Package and Request Group Validations*
- *Validation Special Characters*
- *System Validations*

## What are Validations

Validations are used in two main ways in Kintana:

- **Fields:**  
Validations determine the field’s component type (text field, drop down list, etc.) and the fields possible values. Fields can be created for a number of Kintana entities: Object Types, Request Types, Request Header Types, and User Data.
- **Workflow step results:**  
Validations determine the possible results exiting a Workflow step. For example, the validation WF - STANDARD EXECUTION RESULTS contains the possible execution step results of **SUCCEEDED** or **FAILED**.

Kintana provides a number of pre-seeded (system) Validations with every installation or upgrade. When configuring your system, you can select to use these system Validations. If no Validation exists that meets your specific requirements, you can create a new Validation using the VALIDATION WORKBENCH. See [“Creating a Validation”](#) on page 247 for details.

## Validation Component Types - Overview

The following table summarizes the types of field components that can be used in Kintana. Note that only certain component types can be used in a Workflow step source’s Validation.

Table 0-2. Component Types




Component Type	Use In Workflow?	Example**	Description
Text Field	Yes		Text entry fields displayed on a single line.
Drop down list	Yes		Field showing a column of choices.
Radio Button	No		Field providing a Yes/No input.



Table 0-2. Component Types







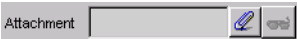





Component Type	Use In Workflow?	Example**	Description
Auto-complete list	Yes		Field showing list of choices with multiple columns.
Text Area	No		Text entry field that can span multiple lines.
Date Field	No		Supports a variety of date and time formats: long, medium, and short.
Web Address (URL)	No		Text entry field for entering a URL. Pressing the U button opens a browser window to the specified web address.
File Chooser	No		Used only in Object Types. Requires that two fields be defined with the following Tokens: P_FILE_LOCATION and P_SUB_PATH. See <i>“Using Directory and File Choosers”</i> on page 268 for configuration details.
Directory Chooser	No		Used only in Object Types. Requires that a parameter field be defined with the Token P_FILE_LOCATION.
Attachment	No		Field for indicating file attachments. Comes with buttons for locating files for previewing contents of the selected file.
Password field	No		Field for capturing passwords.

Table 0-2. Component Types

Component Type	Use In Workflow?	Example**	Description
Table Component	No	<p><b>Table Component</b> (No Entries) </p>	<p>Used to enter multiple records into a single Kintana component. The table component can be configured to include multiple columns of varied data types. Additionally, this component supports rules for populating elements within the table and provides functionality for capturing column totals. See <a href="#">“Configuring the Table Component”</a> on page 272 for details.</p> <p>Fields of this component can only be added to Request Types, Request Header Types and Request User Data.</p>
Budget	No	<p><b>Budget</b> (No Budget) </p>	<p>Field that can be added to the Request Type to enable access to view, edit or create Budgets associated with a Request or Project.</p> <p>Fields of this component can only be added to a Request Type.</p>
Staffing Profile	No	<p>(No Staffing Profile) </p>	<p>Field that can be added to the Request Type to enable access to view, edit or create Staffing Profiles associated with a Request or Project.</p> <p>Fields of this component can only be added to a Request Type.</p>
Resource Pool	No	<p><b>Resource Pool</b> (No Resource Pool) </p>	<p>Field that can be added to the Request Type to enable access to view, edit or create Resource Pools associated with a Request or Project.</p> <p>Fields of this component can only be added to a Request Type.</p>

## Creating a Validation

Generating certain Workflow steps may require specific validations to ensure that business procedures are being followed. It is necessary to have both the Validation Editor and the Validation Values Editor access grants to add a new validation. See "[Kintana Security Model](#)" for a discussion of security groups and access grants.

To define a new Validation:

1. Click **NEW VALIDATION** on the VALIDATION WORKBENCH or select **FILE -> NEW -> VALIDATION** from the menu. The VALIDATION window opens.
2. Enter the name of the new Validation in the NAME field.
3. Enter a description of the new Validation in the DESCRIPTION field.
4. Select whether the Validation is enabled or not in the ENABLED check box.
5. In the USE IN WORKFLOW checkbox, specify whether or not this Validation can be used in a Workflow step source. You can only use Text Field, Drop Down List and Auto-Complete component types within Workflow step sources.
6. Select the desired type of Validation from the COMPONENT TYPE drop down list. Enter any additional information required for the component type selected. See the Validation chapter in "[Configuration Workbench Reference](#)" for descriptions of the fields required to define each component type.
7. Click **OWNERSHIP** to select which users will be able to edit, copy and delete this validation.
8. To save changes to the Validation without closing the window, click **SAVE**. To save changes and close the window, click **OK**.

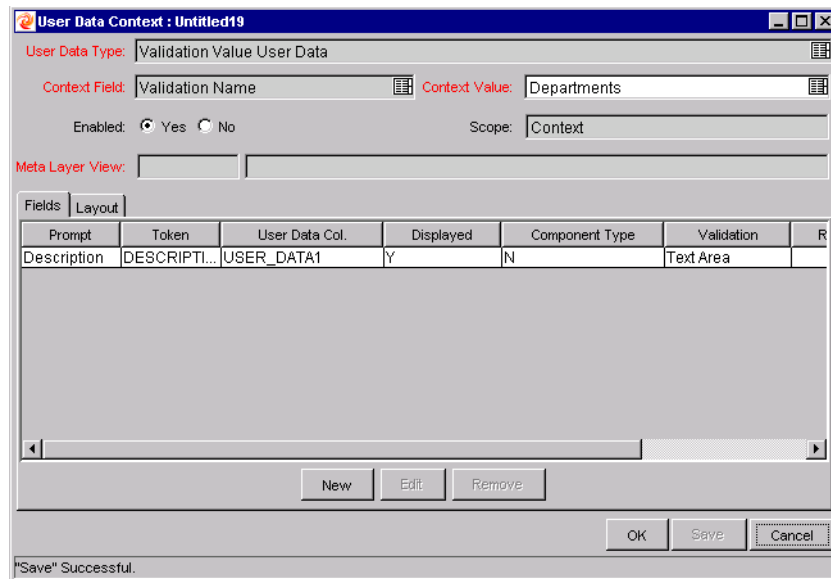
## User Data on the Validation Value

You can enable the **USER DATA** tab to capture more information related to an individual Validation value within a specific Validation. For example, you can create a DESCRIPTION user data field that is associated with the DEPARTMENTS Validation. When you add new values to the validation, you can click on the **USER DATA** tab and enter a description for that value.

The **USER DATA** tab can only be used when creating a drop down or an auto-complete validated by a list.

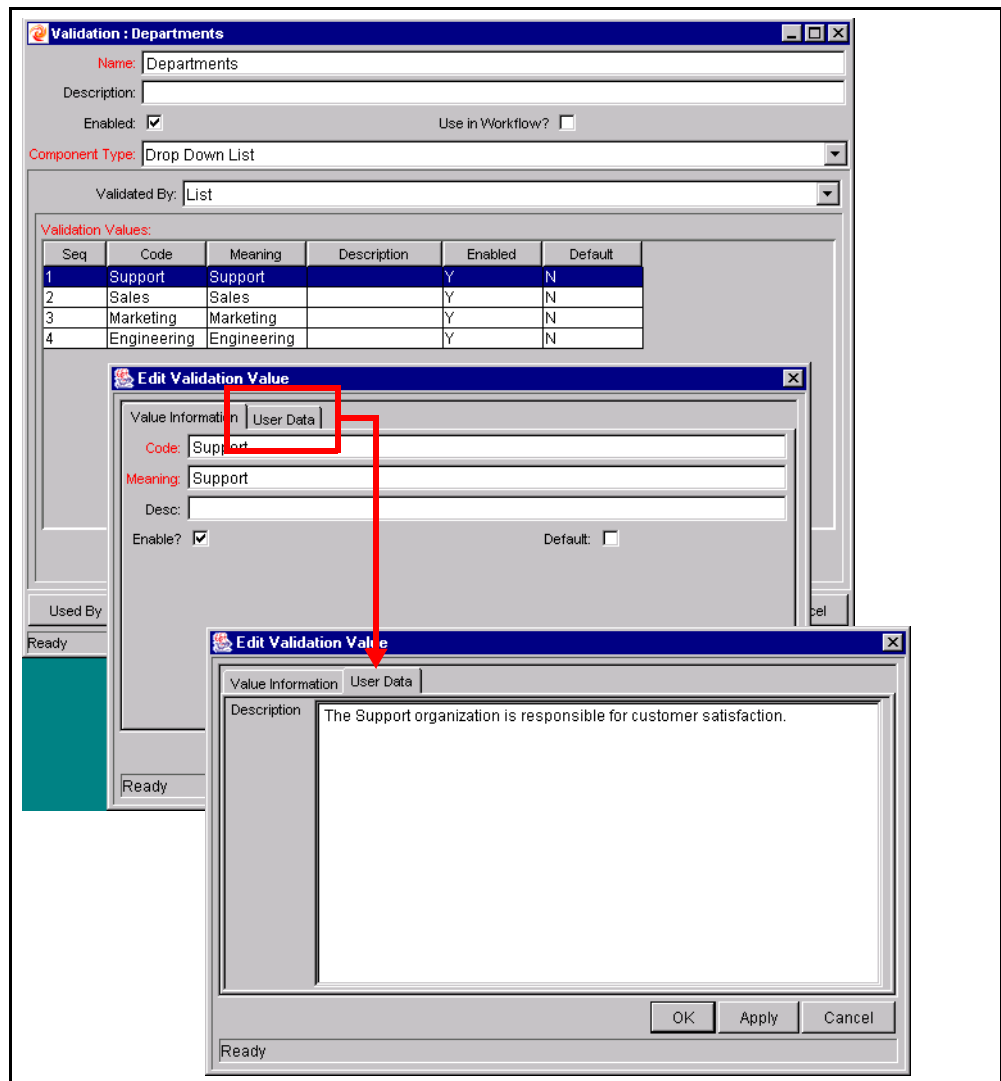
To enable the **USER DATA** tab in the EDIT VALIDATION VALUE window:

1. Create the Validation and note its name.
2. Open the USER DATA workbench.
3. Click **NEW USER DATA CONTEXT**.
4. Select **VALIDATION VALUE USER DATA** from the USER DATA TYPE field.
5. Click **NEW** to create a User Data field.



Prompt	Token	User Data Col.	Displayed	Component Type	Validation	R
Description	DESCRIPTI...	USER_DATA1	Y	N	Text Area	

6. Save the settings in the USER DATA window.
7. On the VALIDATION window, add or edit a Validation value. The **USER DATA** tab is now enabled. You can select the tab and enter information in the newly defined user data field.



See the User Data appendix in *"Configuring a Request Resolution System"* for more details on using User Data in Kintana.

## Editing Validations

You can open and edit Validations using the Kintana Workbench. You should exercise caution when editing Validations that are currently used by fields or

Workflow step sources. Both field and Workflow step validations can be tied to Workflow logic. Changing the Validation values can invalidate a process.

For example, ACME changes the PRIORITY field Validation to include a new value **VERY EASY**. ACME uses a deployment system Workflow that has an EVALUATE PRIORITY step that routes the Package based on the value in the Priority field (using a Token execution type). ACME, however, did not update the Workflow to enable a transition out of the step for the case when PRIORITY = **VERY EASY**. When a **VERY EASY** Package enters the EVALUATE PRIORITY step, it will get stuck.

The following restrictions apply to editing Validations:

- User must have the following Access Grants:
  - Edit Validations
  - Edit Validation Values
- User must be a member of the Ownership Group for the Validation
- You can not change which Validation is associated with a Workflow step source after a Package has traversed that step. You can, however, still edit the values within that Validation.

## Creating a URL to Open the Validation Window

You can create a URL that opens a specific Validation in the Kintana Workbench. This can provide a quick link to the configuration screen for a Validation that is expected to change frequently. This URL can be included on your internal or external Web pages or a list of browser Favorites to provide convenient access to the Validation's definition.

Use the following URL format to access a specific VALIDATION window:

```
http://host:port/kintana/servlet/SmartURL?screen=VAL&pkname=<ValidationName>
```



Note

The following URL opens the VALIDATION window for the Validation named "Development Priorities."

```
http://host:port/kintana/servlet/SmartURL?screen=VAL&pkname=Development+Priorities
```

---

## Deleting Validations

Validations can be deleted from the Kintana Workbench. To delete a Validation, you must be a member of the Validation's Ownership Group and have the EDIT VALIDATIONS access grant.

A Validation can not be deleted when:

- It is a system Validation (a Validation that is delivered with Kintana as seed-data)
- It is being used by a Workflow step source. Validations referenced by Workflow step sources can only be disabled. A disabled Validation continues to function in existing Workflow steps, but can not be used when defining a new step source.
- It is being used by a field in a Kintana entity (Object Type, Request Type, User Data, Report Type, or Project Template field). Validations referenced by entity fields can only be disabled. A disabled Validation continues to function in existing fields, but can not be used when defining a new field.



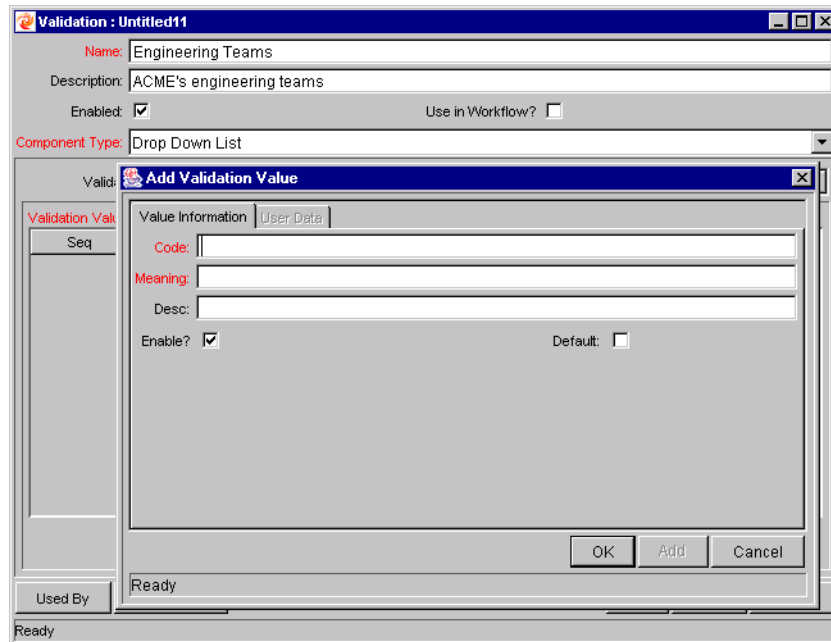
Although you may not be able to delete a custom Validation in all cases, you can disable it. This will allow the Validation to be used in any active Workflows or Kintana entities, but will keep it from being used in any new Workflow or entity definitions.

## Static List Validations

You can create Validations that provide a static list of options to the user. For example, ACME, Inc. can create a Validation for their engineering teams. They create a Validation called ENGINEERING TEAMS, consisting of the following values: **NEW PRODUCT INTRODUCTION**, **PRODUCT ONE**, and **PRODUCT TWO**.

A static list validation can be a drop down or an auto-complete list component. To add values to the Validation list:

1. In the VALIDATION window, select **DROP DOWN LIST** or **AUTO COMPLETE LIST** from the COMPONENT TYPE field.
2. Select **LIST** from the VALIDATED BY field.
3. Click **NEW** and add a value. The ADD VALIDATION WINDOW opens.



4. Enter the CODE, MEANING and DESCRIPTION of the value. See "[Configuration Workbench Reference](#)" for definitions of these fields.
5. Optionally set the Validation value as the default by checking the DEFAULT field. The default option is only available for drop down lists.
6. Click OK to close the window and add the value to the Validation. Click **ADD** to add the value and keep the ADD VALIDATION VALUE open.

Validation values can be re-ordered using the up and down arrow buttons. The sequence of the Validation values determines the order that the values are displayed in the list.



Tip

You can copy existing values defined in other Validations using the **COPY FROM** button. Click **COPY FROM** and query an existing list-validated Validation and choose any of the Validation values. Click **ADD** or **OK** in the COPY FROM window and the selected value or values are added to the list.



Note

Be careful when creating Validations (drop down lists and auto-complete lists) that are validated by lists. Each time the set of values changes, you will be forced to update the Validation. Consider, instead, validating using a SQL query or PL/SQL function to obtain the values from a database table.



## Dynamic List Validations

You can create Validations that provide a dynamic list to the user. This is often a better approach than defining static list validations. Each time a static list Validation needs to be updated, a manual update has to occur. Dynamic list Validations can often be constructed in such a way as to automatically pick up and display the altered values.

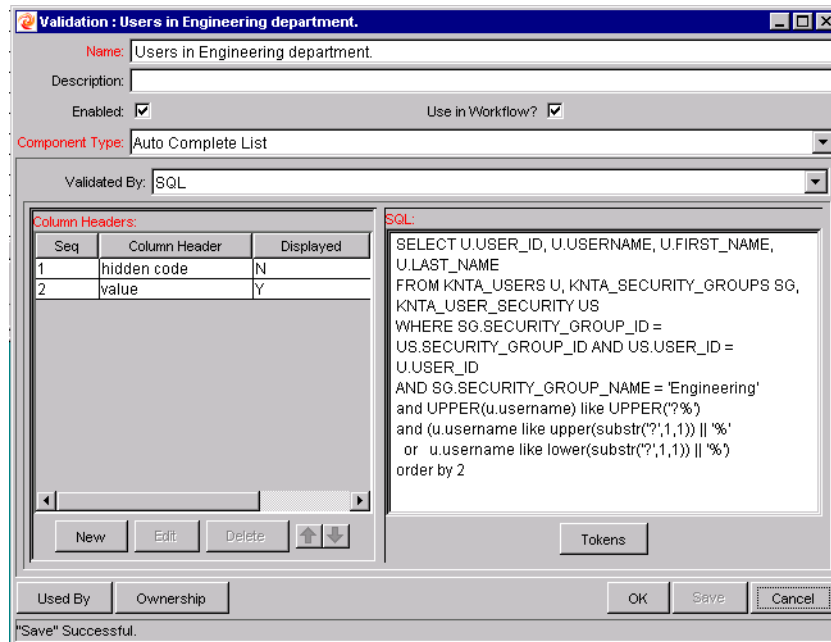
For example, ACME needs a field validation that will list all Kintana users who are on their Support Team. They could construct a Validation that is validated by a list of users, but any time the Support Team changed (members join or leave the department) the list would have to be manually updated. ACME decides instead to create a dynamic list validation. They create an auto-complete list validation that is validated by a SQL statement. The SQL statement returns all users who are a member of the **SUPPORT TEAM** Security Group. When the Security Group membership is altered, the Validation is automatically updated with the correct values.

A dynamic list validation can be created using a drop down or an auto-complete list component. The lists are dynamically generated using either:

- [SQL Validation](#)
- [Command Validation](#)

## SQL Validation

You can use a SQL statement to generate the values in a Validation. SQL can be used as a validation method for drop down lists and auto-complete lists. To define a dynamic list of choices, set a drop down list or auto-complete list to VALIDATED BY - **SQL**. Then in the SQL area, enter the Select statement that queries the necessary database. See "[Configuration Workbench Reference](#)" for an explanation of each screen and field in the VALIDATION window.



Example

ACME, Inc. creates an auto-complete field that lists all Kintana users in the “Engineering” department. They choose to validate the list by SQL.

```
SELECT U.USER_ID, U.USERNAME, U.FIRST_NAME, U.LAST_NAME
FROM KNTA_USERS U, KNTA_SECURITY_GROUPS SG,
KNTA_USER_SECURITY US
WHERE SG.SECURITY_GROUP_ID = US.SECURITY_GROUP_ID AND
US.USER_ID = U.USER_ID
AND SG.SECURITY_GROUP_NAME = 'Engineering'
and UPPER(u.username) like UPPER('?%')
and (u.username like upper(substr('?',1,1)) || '%'
or u.username like lower(substr('?',1,1)) || '%')
order by 2
```

When a new user is added to Kintana and included in the “Engineering” Security Group, that user will automatically be included in the auto-complete list.



Tip

Kintana may already have a Validation that meets your process requirements. If it does, consider using that Validation in your process. Also consider copying and modifying Validations that are similar to the desired Validation. See *“System Validations”* on page 287 for a complete list of Validations that are delivered with Kintana.

## SQL Validation Tips

The following guidelines are helpful when writing a SQL statement for a SQL-validated Validation:

- The SQL statement must query at least two columns. The first column is a hidden value which is never displayed, and is often stored in the database or passed to internal functions. The second column is the value that is displayed in the field. All other columns are for information purposes and are only displayed in the auto-complete window. Extra columns are not displayed for drop down lists.
- When something is typed into an auto-complete list field, the values in the auto-complete window that appear are constrained by what was first typed in the field. Generally, the constraint is case insensitive. This is accomplished by writing the SQL statement to query only values that match what was typed.

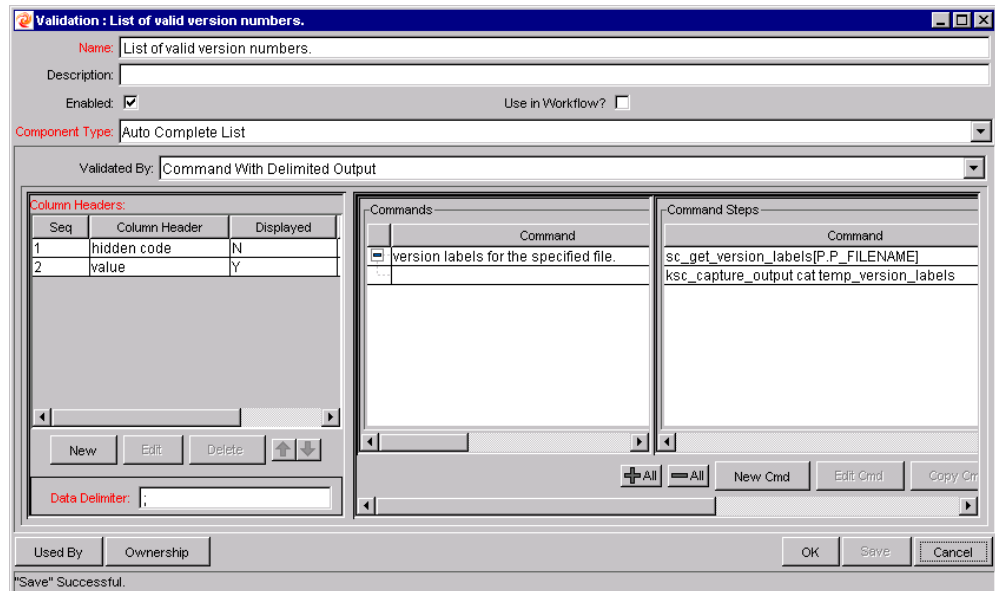
Before the auto-complete window is displayed, all question marks in the SQL statement are replaced by the text that the user typed. In general, if the following conditions are added to the WHERE clause in a SQL statement, the values in the auto-complete window are constrained by what the user typed.

```
where UPPER(<displayed_column>) like UPPER('?%')
and (<displayed_column> like upper(substr('?',1,1)) || '%')
or <displayed_column> like lower(substr('?',1,1)) || '%')
```

Any column aliases included directly in the SQL statement are not used. The names of the columns, as displayed in auto-complete lists, are determined from the Column Headers. Drop down lists do not have column headers

## Command Validation

An auto-complete list can contain command line executions that return and display a list of values. To define a dynamic list of choices, set an auto-complete list to VALIDATED BY - **COMMAND WITH DELIMITED OUTPUT** or **COMMAND WITH FIXED WIDTH OUTPUT**. Then enter commands the COMMANDS area.



## Using Auto-Complete Validations

The values in an auto-complete list can be specified in the following ways. In the VALIDATE BY field, select one of the following:

- **LIST:** Used to enter specific values.
- **SQL:** Uses a SQL statement to build the contents of the list.
- **COMMAND WITH DELIMITED OUTPUT:** Uses a system command to produce a character-delimited text string and uses the results to define the list.
- **COMMAND WITH FIXED WIDTH OUTPUT:** uses a system command to produce a text file and parses the result on the basis of the width of columns, as well as the headers.

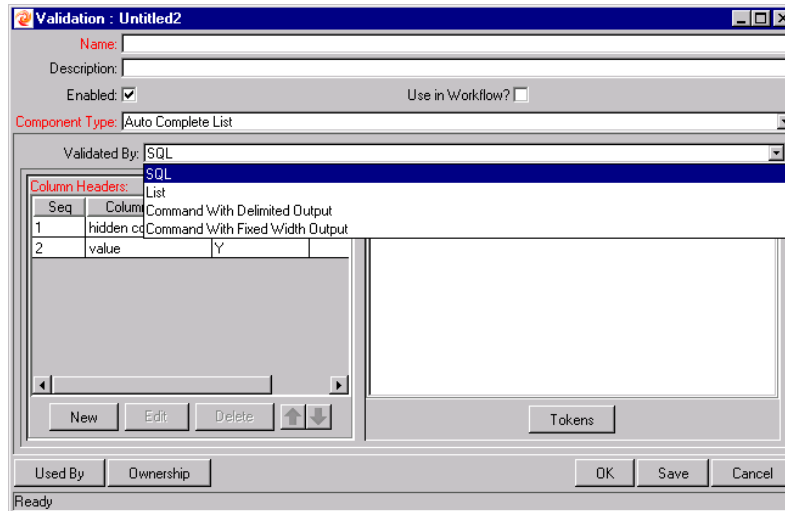


Figure 0-2 Auto-Complete List

The following sections discuss the following topics:

- [Validation by Command With Delimited Output](#)
- [Validation by Command With Fixed Width Output](#)
- [User-Defined Multi-Select Auto-Complete Fields](#)
- [Example: Token Evaluation and Validation by Command with Delimited Output](#)
- [Special Case - Limiting the Number of Returned Rows](#)

For more information on creating auto-completes validated by List or SQL, refer to the following sections:

- [“Static List Validations”](#) on page 251
- [“Dynamic List Validations”](#) on page 253

## Validation by Command With Delimited Output

Validations by Command with Delimited Output can be used to get data from an alternate source, and use that data to populate an auto-complete field. This functionality provides additional flexibility when designing auto-complete lists.

Many enterprises need to use alternate sources of data within their applications. Examples of these sources are a flat file, an alternate database source, or output from a command line execution. Special commands may be used in conjunction with these alternate data sources, in the context of a Validation, to provide a list of values.

To configure a validation by command with delimited output:

1. In the VALIDATION WORKBENCH, under VALIDATED BY, choose **COMMAND WITH DELIMITED OUTPUT** and input the delimiting character.
2. Under NEW COMMAND, enter in the command steps to be executed. These can include Kintana Special Commands. Your commands should include the Special Command `ksc_capture_output`, which captures and parses the delimited command output. If the `ksc_capture_output` Special Command is surrounded by the `ksc_connect` and `ksc_disconnect` commands, the command will be run on the remote system. Otherwise, the command will be run locally on the Kintana server (similar to `ksc_local_exec`).



Example

The simple example below uses a comma for a delimiter and has the validation values red, blue and green. The script places the validations into the `newfile.txt` file, and then uses the Special Command `ksc_capture_output` to process the text of the file.

```
ksc_begin_script [AS.PKG_TRANSFER_PATH]newfile.txt
red,red
blue,blue
green,green
ksc_end_script
ksc_capture_output cat [AS.PKG_TRANSFER_PATH]newfile.txt
```

*Table 0-3* shows the VALIDATION window for COMMAND WITH DELIMITED OUTPUT.

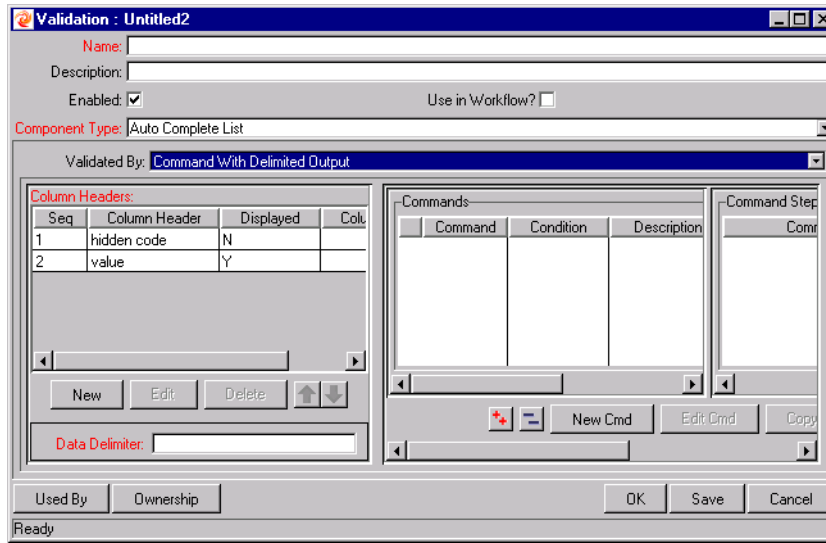


Figure 0-3 Validation by Command with Delimited Output

Table 0-3. Validation by Command With Delimited Output

Field	Definition
COMMAND PANEL	Panel where new commands can be added to capture Validation values.
DATA DELIMITER	Indicates the character or key by which the file will be separated into the Validation columns.

Headers can also be defined for the columns selected. These column headers are used in the window that opens when a value is selected from an auto-complete list. To define a new header, click **NEW** under COLUMN HEADER. [Table 0-4](#) shows the fields that can be entered for a column header. If a column header is not defined for each column in a Command, a default name is used.

Table 0-4. Column Headers

Field	Definition
COLUMN HEADER	The name of the column that is displayed in the auto-complete window.
DISPLAY	Determines whether or not the header is displayed in the Validation.

## Validation by Command With Fixed Width Output

Validations by `COMMAND WITH FIXED WIDTH OUTPUT` can be used to obtain data from an alternate source, and use that data to populate an auto-complete field. This functionality provides additional flexibility when designing auto-complete lists.

Many enterprises need to use alternate sources of data within their applications. Examples of these sources are a flat file, an alternate database source, or output from a command line execution. Special commands may be used in conjunction with these alternate data sources, in the context of a Validation, to provide a list of values on the fly.

In the `VALIDATION WORKBENCH`, under `VALIDATED BY`, choose **COMMAND WITH FIXED WIDTH OUTPUT** and input the appropriate width information.

Then, under `NEW COMMAND`, enter in the command steps to be executed. These can include Kintana Special Commands. Your commands should include the Special Command `ksc_capture_output`, which captures and parses the delimited command output. If the `ksc_capture_output` Special Command is surrounded by the `ksc_connect` and `ksc_disconnect` commands, the command will be run on the remote system. Otherwise, the command will be run locally on the Kintana server (similar to `ksc_local_exec`).



Example

The example below has the validations red, blue and green. The column width is set to a value of 6. The script places the validations into the `newfile.txt` file.

```
ksc_begin_script [AS.PKG_TRANSFER_PATH]newfile.txt
red      red
blue     blue
green    green
ksc_end_script
ksc_capture_output cat [AS.PKG_TRANSFER_PATH]newfile.txt
```



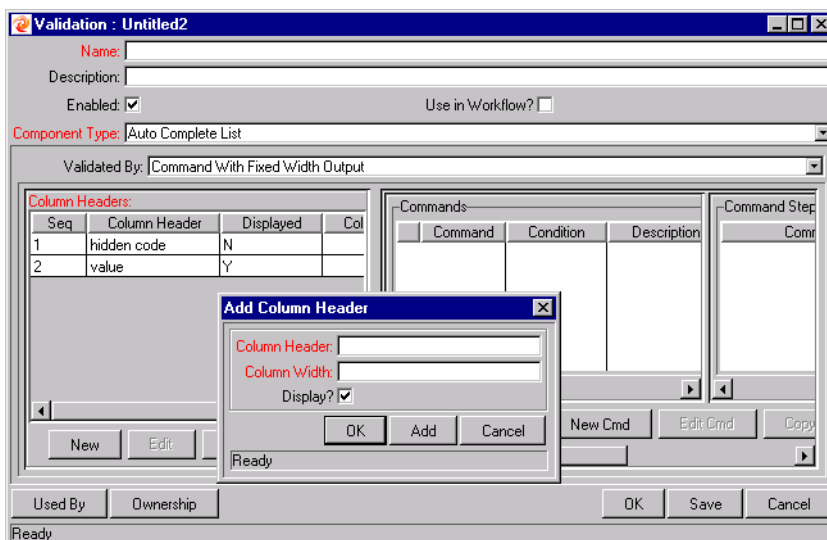


Figure 0-4 Validation by Command with Fixed Width Output

Table 0-5. Validation by Command With Fixed Width Output

Field	Definition
COMMAND PANEL	The panel where new commands can be added to capture Validation values.

Headers can also be defined for the columns selected. These column headers are used in the window that opens when a value is selected from an auto-complete list. To define a new column header, click **NEW** under COLUMN HEADER. [Table 0-6](#) shows the fields can be entered for a column header. If a column header is not defined for each column in a Command, a default name is used.

Table 0-6. Column Headers

Field	Definition
COLUMN HEADER	The name of the column that is displayed in the Auto Complete dialog.
DISPLAY	Whether or not the column is displayed. The first column is never displayed and the second column is always displayed.
COLUMN WIDTH	The number of characters in each column of the output generated as a result of the command.

## User-Defined Multi-Select Auto-Complete Fields

A number of auto-complete fields in the Workbench have been configured by Kintana to allow users to open a separate window for selecting multiple values from a list. Users can also define custom auto-complete fields to have multi-select capability when creating various Kintana entities.

The user-defined multi-select capability is supported for:

- User Data fields
- Report Type fields
- Request Type fields
- Project Template fields

The user-defined Multi-Select capability is **not** supported for:

- Request Header Types
- Object Types

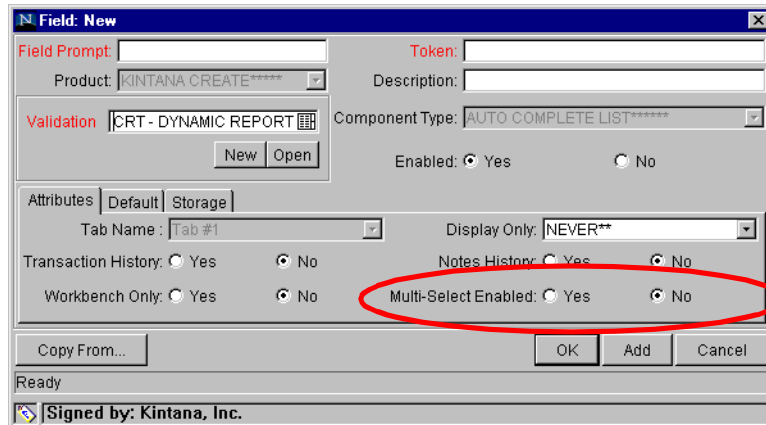
In order to use this feature when creating a new entity, users must:

- Select a Validation for the new entity that has **AUTO-COMPLETE LIST** as the Component Type. This enables the **MULTI-SELECT ENABLED** field in the **FIELD: NEW** window.
- In the **FIELD: NEW** window, users must click **YES** for the **MULTI-SELECT ENABLED** radio button.

The step-by-step procedure for defining multi-select capability in User Data, Report Type, Request Type Project Template fields is very similar. The procedure for enabling this capability for Request Type field is shown below as an example.

To define a multi-select auto-complete field for a Request Type:

1. Click the **CREATE** screen group and click the **REQUEST TYPES** screen. The **REQUEST TYPE WORKBENCH** opens.
2. Click **NEW REQUEST TYPE**. The **REQUEST TYPE** window opens.
3. Click **NEW**. The **FIELD: NEW** window opens.



4. Click the auto-complete icon for the **VALIDATION** field. The VALIDATE window opens.
5. In the VALIDATE window, select a Validation that has **AUTO-COMPLETE LIST** as the Component Type.
6. Click **OK** in the VALIDATE window. The VALIDATE window closes.

The **VALIDATION** field is populated with the selection from the VALIDATE window. The **MULTI-SELECT ENABLED** option is now enabled.

7. Click the **YES** radio button for the **MULTI-SELECT ENABLED** option.
8. The **POSSIBLE CONFLICTS** window opens. It warns you not to use a multi-select auto-complete for Advanced Queries, Workflow Transitions and Reports. If this field is not going to be used in Advanced Queries, Workflow Transitions or Reports, click **YES** to continue.
9. Configure the other options in this window for the new Request Type.
10. Click **OK**.

The field is now enabled for multi-select auto-complete.

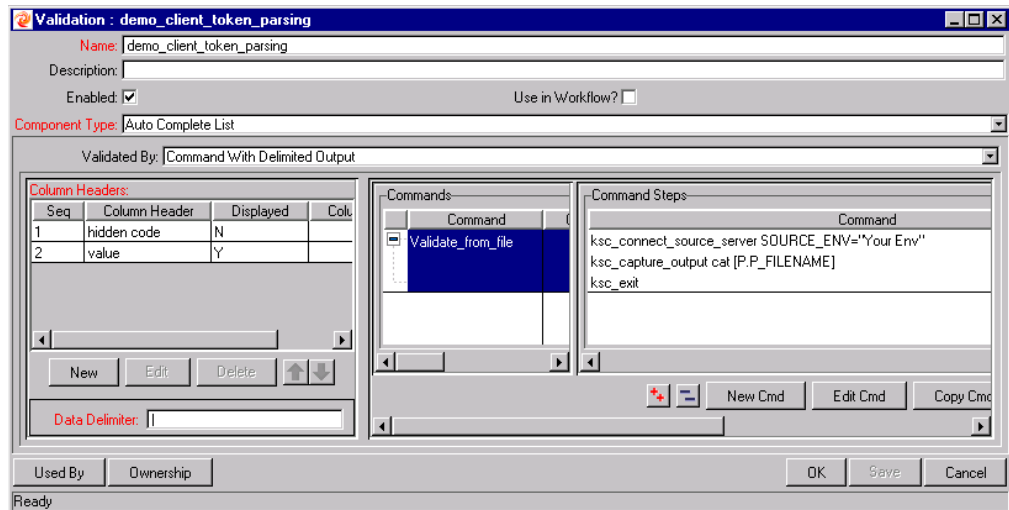
## Example: Token Evaluation and Validation by Command with Delimited Output

The Validation functionality can be extended to include field dependent token evaluation. Validations can be configured to dynamically change, depending on the client-side value entered in another field.

To use field dependent token evaluation, it is necessary to configure a Validation in conjunction with an Object Type, Request Type, Report Type, Project Template, or User Data definition. Consider the following example for setting up an Object Type using field dependent tokens.

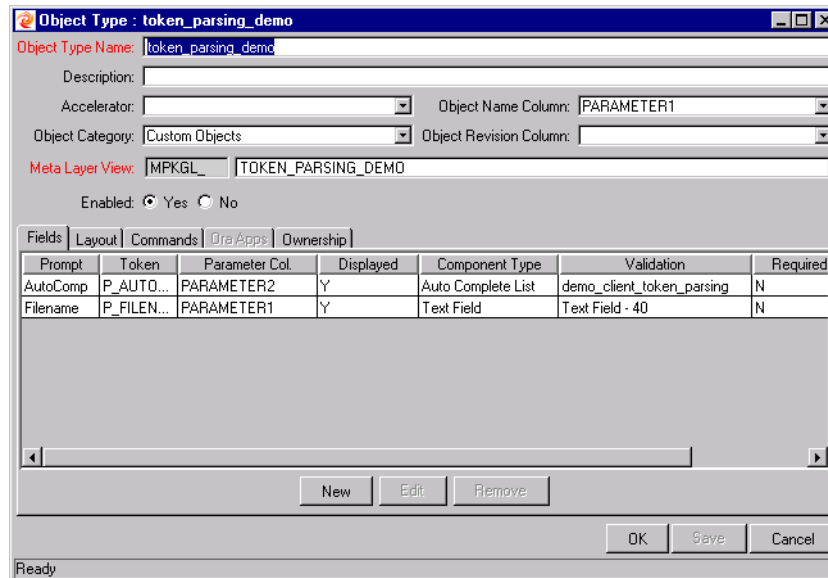
1. Generate a Validation and set the following parameters as shown:
  - a. NAME: **DEMO\_CLIENT\_TOKEN\_PARSING**
  - b. COMPONENT TYPE: **AUTO COMPLETE LIST**
  - c. VALIDATED BY: **COMMAND WITH DELIMITED OUTPUT**
  - d. DATA DELIMITER: | (bar)
  - e. COMMAND
    - o COMMAND: **VALIDATE\_FROM\_FILE**
    - o STEPS

```
ksc_connect_source_server SOURCE_ENV="Your Env"  
ksc_capture_output cat [P.P_FILENAME]  
ksc_exit
```



When called, this Validation will connect to an Environment called 'YOUR ENV' and retrieve data from a file specified by the token P\_FILENAME. The file should be located in the directory specified in the BASE PATH in the ENVIRONMENT window.

2. Generate an Object Type named **TOKEN\_PARSING\_DEMO**.



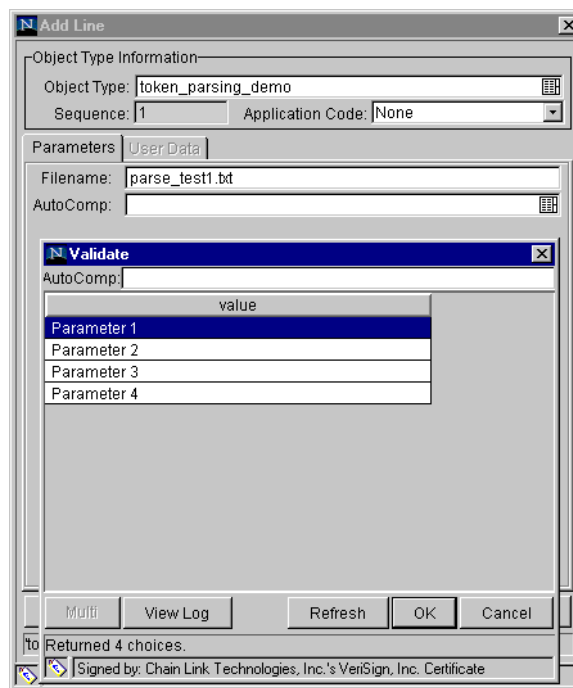
- a. Generate a new field with the following parameters:
  - o NAME: **FILENAME**
  - o TOKEN: **P\_FILENAME**
  - o VALIDATION: **TEXT FIELD - 40**
- b. Generate a new field with the following parameters:
  - o NAME: **AUTOCOMP**
  - o TOKEN: **P\_AUTOCOMP**
  - o VALIDATION: **DEMO\_CLIENT\_TOKEN\_PARSING** (this is the Validation that was defined above)
3. For this example to return any values in the auto-complete, a file must be generated in the directory specified in the Base Path in the Environment Detail of 'YOUR ENV' Environment. Generate a file named 'parse\_test1.txt' with the following delimited data:

```

DELIMITED_TEXT1 | Parameter 1
DELIMITED_TEXT2 | Parameter 2
DELIMITED_TEXT3 | Parameter 3
DELIMITED_TEXT4 | Parameter 4
    
```

The Object Type 'token\_parsing\_demo' is now enabled to use this token evaluation. To test the above configuration sample:

1. Generate a new Package.
2. Select a Workflow and click **ADD LINE**.
3. Select **TOKEN\_PARSING\_DEMO** from the OBJECT TYPE drop down list. The following fields are displayed:
  - FILENAME
  - AUTOCOMP
4. Type 'parse\_test1.txt' in the FILENAME field.
5. Click on the auto-complete box in the AUTOCOMP field. The following VALIDATION window opens, displaying the contents of the 'parse\_test1.txt' file.



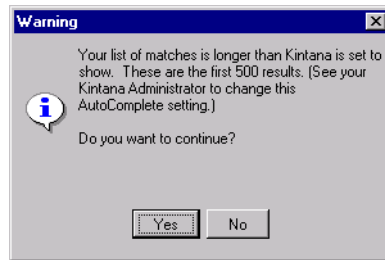
### Special Case - Limiting the Number of Returned Rows

Fields that must be validated against a list of pre-defined values use auto-complete lists. When users type a partial value into an auto-complete field and try to tab out of the field, an auto-complete list opens that shows all values

matching the partial value. If no values match the partial value entered, the full list is returned.

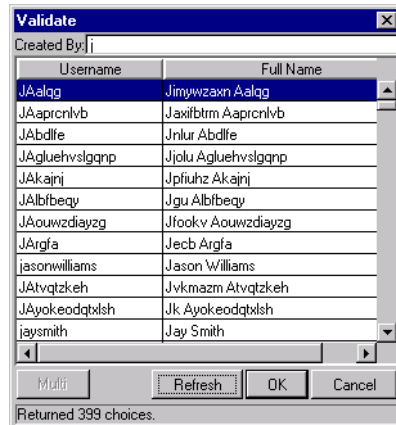
The number of results returned in the list affects how quickly the results are displayed. The larger the number of results, the longer it takes to load. When the number of returned results becomes very large, it becomes advantageous to narrow the search by typing in some limiting text. For example, if you type 'm' into the ASSIGNED TO USER field in the PACKAGE window, a list of all users whose Username starts with 'm' appears.

The auto-complete component can be configured to provide a warning when the number of rows returned from a search will exceed a certain amount. For example, assume that you have 2000 entries in a CREATED BY auto-complete list. When you click on the CREATED BY auto-complete list (without limiting the search), Kintana will display the following message:



You can then either:

- Click **YES** to display the first 500 (or other preconfigured number) results. You can then refine your search in the auto-complete's validate window. For example, if you are searching for a user named 'John Smith,' type 'J' to limit the search to only those users whose names start with the letter 'J.' You may have to click the **REFRESH** button if 'J' was not included in the first 500 rows.



- Click **No** to abort the auto-complete search.

Kintana System Administrators can choose at which number of rows the warning will appear. This is set by creating a `MAX_AUTOCOMPLETE_ROWS` parameter in the `server.conf` file located in the `<KNTA_Home>` directory on the Kintana server.

To enable this auto-complete feature, type the following text into your `server.conf` file:

```
com.kintana.core.server.MAX_AUTOCOMPLETE_ROWS=X
```

where `X` is the number of rows above which the warning will appear. The auto-complete is set to `MAX_AUTOCOMPLETE_ROWS=500` by default.

For more information on setting up the `server.conf` parameters, refer to the “Kintana System Administration Guide” available on the Kintana Download Center.



Note

This is a site-wide setting.

## Using Directory and File Choosers

Directory and File Choosers are only used with Object Types. The following sections discuss them in more detail:

- [Directory Chooser](#)



- *File Chooser*

## Directory Chooser

The DIRECTORY CHOOSER field can be used to select a valid directory from an Environment. Kintana Deliver connects to the first Source Environment on a Workflow and allows navigation through the directory structure and the selection of a directory from the list.

- The Directory Chooser field can only be used on an Object Type in Kintana Deliver.
- On every Object Type that a Directory Chooser is chosen, it is also necessary to have a field whose token is 'P\_FILE\_LOCATION' and whose validation is 'Kintana Deliver - File Location'. The possible values for this field are **CLIENT** and **SERVER**. If **CLIENT** is chosen, the Directory Chooser connects to the Client Base Path of the Source Environment. If **SERVER** is chosen, the Directory Chooser connects to the Server Base Path of the Source Environment.

## File Chooser

A FILE CHOOSER field can be used by Object Types to select a valid file from an Environment. Kintana connects to the first Source Environment on a Workflow and provides the ability to view all files within a specific directory and select one from the list.

On every Object Type that a File Chooser is chosen, it is necessary to have two other fields defined.

1. The first is a field for the File Location for the directory chooser, described in the previous section.
2. The second is a field whose token is 'P\_SUB\_PATH'. This field is the directory from which the file is selected and is usually a Directory Chooser field.

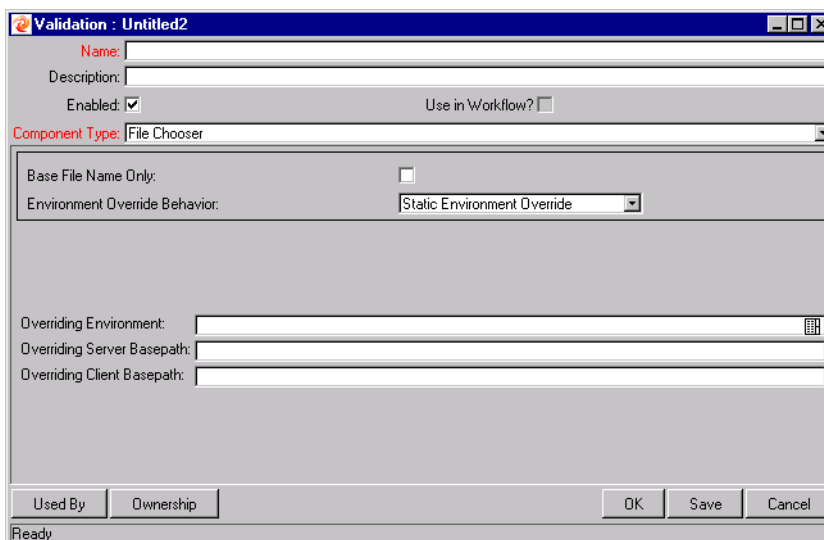


Figure 0-5 Validation Window for Static Environment Override in File Chooser.

Table 0-7. File Chooser Field

Field	Definition
BASE FILE NAME ONLY	Defines whether the base file name only (without its suffix) or the complete name is displayed.
ENVIRONMENT OVERRIDE BEHAVIOR	Used to select files from a specific environment other than the default environment.

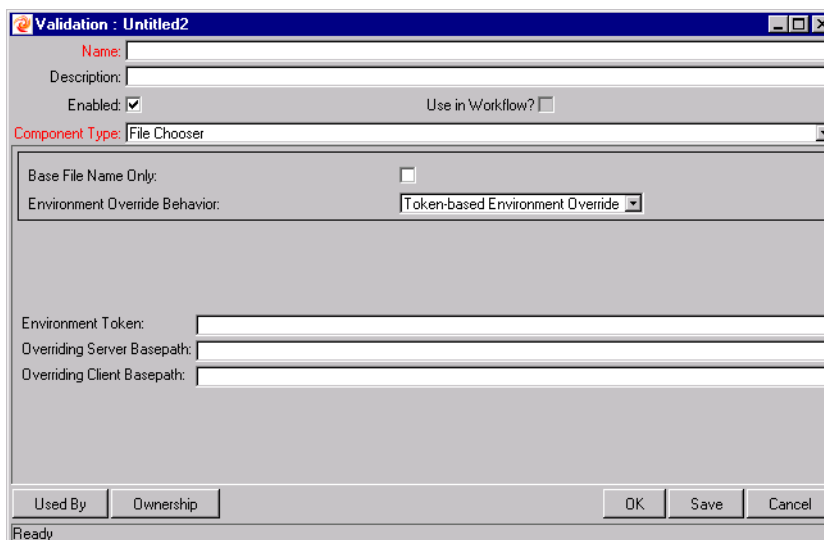
The ENVIRONMENT OVERRIDE BEHAVIOR drop down list contains three options: **DEFAULT BEHAVIOR**, **STATIC ENVIRONMENT OVERRIDE**, and **TOKEN-BASED ENVIRONMENT OVERRIDE**.

**STATIC ENVIRONMENT OVERRIDE** provides the ability to override one Environment at a time. The fields for Static Environment Override are pictured in [Figure 0-5](#) and described in [Table 0-8](#).

Table 0-8. Static Environment Override

Field	Definition
OVERRIDING ENVIRONMENT	Selects the Environment to be overridden.
OVERRIDING SERVER BASEPATH	The server basepath of the Environment may be overridden.
OVERRIDING CLIENT BASEPATH	The client basepath of the Environment may be overridden.

**TOKEN-BASED ENVIRONMENT OVERRIDE** provides the ability to select a token that will resolve to the overriding Environment. The fields for **TOKEN-BASED ENVIRONMENT OVERRIDE** are shown in *Figure 0-6* and defined in *Table 0-9*.



*Figure 0-6 Validation Window for Token-Based Environment Override in File Chooser.*

*Table 0-9. Token-Based Environment Override*

Field	Definition
ENVIRONMENT TOKEN	Select the token that will resolve to the overriding Environment.
OVERRIDING SERVER BASEPATH	The server basepath of the Environment that is to be resolved by the token may be overridden.
OVERRIDING CLIENT BASEPATH	The client basepath of the Environment that is to be resolved by the token may be overridden.

## Creating 1800 Character Text Areas

Standard Text Areas are between 1 and 200 characters. Kintana does, however, allow you to create a Text Area Validation with a character length of 1800. To create this Validation:

1. Open the Validation Workbench.
2. Search for “**TEXT AREA - 1800.**”
3. In the results tab, select **TEXT AREA - 1800.**
4. Click **COPY.**
5. Rename the Validation.

The new Text Area Validation (with a length of 1800) can be used when defining a custom field in Kintana.

## Configuring the Table Component

The table component is used to enter multiple records into a single Kintana field on a Kintana Request. The table component can be configured to include multiple columns of varied data types. Additionally, this component supports rules for populating elements within the table and provides functionality for capturing column totals.

1. Click the Table Component icon to open the Table Component entry page.

2. Add, edit, or delete entries in the list.

Seq	Column 1	Column 2	Column 3
<input type="checkbox"/> 1	Entry 1	Entry 2	Entry 3

Fields of this component can only be added to Request Types, Request Header Types and Request User Data.

To configure and use a Table Component:

- [Define the Table Component in the Validation Workbench](#)
- [Add the Table Component to a Request Type](#)



Example

ACME creates a Request Type to request quotes and parts for hardware. Each entry of this type has four elements: PART, SUB-TYPE, PART NUMBER, and UNIT PRICE. ACME creates a Table Component field called HARDWARE INFORMATION to collect this information.

When the user logs a request for new hardware, the Request displays the HARDWARE INFORMATION field. The user opens the field. He selects a PART, which triggers a rule to populate the PART NUMBER and UNIT PRICE. He submits the Request, which now contains all of the information required to successfully order the hardware.

## Define the Table Component in the Validation Workbench

To create a Table Component field:

1. Open the VALIDATION WORKBENCH in the CONFIGURATION screen group.
2. Click **NEW VALIDATION**. The VALIDATION window opens.
3. Select **TABLE COMPONENT** from the COMPONENT TYPE drop down list.

The screenshot shows the 'Validation : Untitled1' window. The 'Name' field is empty. The 'Description' field is empty. The 'Enabled' checkbox is checked. The 'Use in Workflow?' checkbox is unchecked. The 'Component Type' dropdown is set to 'Table Component'. The 'User Instructions' field is empty. The 'Meta Layer View' field contains 'MREQ\_'. Below this is a table with columns: Column Seq., Column Header, Column Token, Parameter Col., Enabled, and Component. The table is currently empty. At the bottom of the table area are buttons for 'New', 'Edit', and 'Remove'. At the bottom of the window are buttons for 'Used By', 'Ownership', 'OK', 'Save', and 'Cancel'. The status bar at the bottom left says 'Ready'.

4. Enter a Validation NAME and DESCRIPTION.

5. Enter any **USER INSTRUCTIONS**. This text will appear on the top of the table entry page.
6. Create the Table Columns.
  - a. Click **NEW** in the **TABLE COLUMNS** tab. The **FIELD** window opens.
  - b. Define the type of information that will be stored in that column's entries. This may require you to create a new **Validation** for the column.



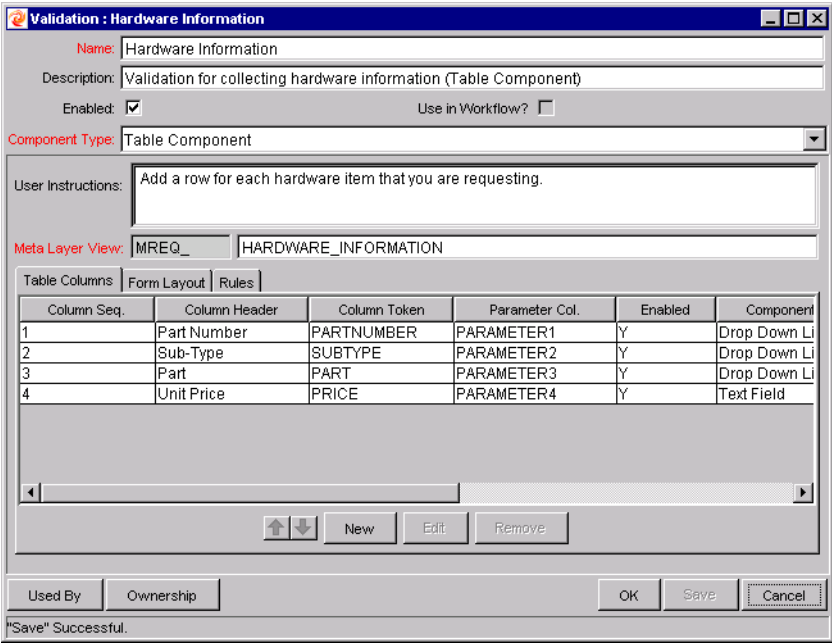
Note

File attachments can not be used in a Table component column.

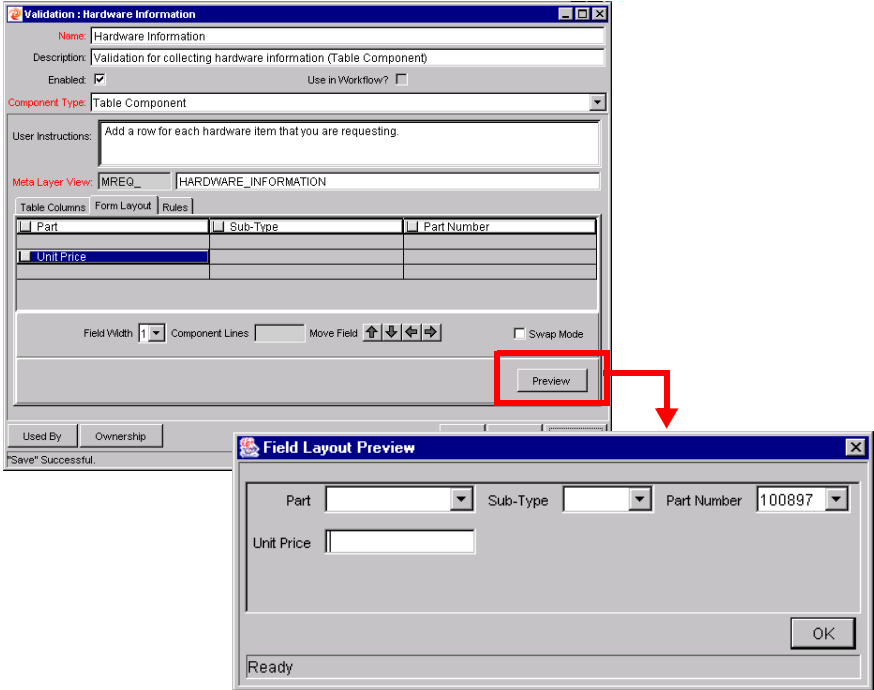
The screenshot shows the 'Field: New' dialog box with the following settings:

- Column Header: Part Number
- Column Token: PARTNUMBER
- Description: (empty)
- Enabled:  Yes  No
- Validation: Hardware Part Numbers (with New and Open buttons)
- Component Type: Drop Down List
- Multi-Select Enabled:  Yes  No
- Attributes: Default | Storage (selected)
- Editable:  Yes  No
- Required: Never
- Display Total:  Yes  No
- Buttons: Copy From..., OK, Add, Cancel
- Status: Ready

- c. Specify the **ATTRIBUTES** (**EDITABLE** OR **REQUIRED**) and any **DEFAULT** behavior.
- d. Click **ADD** to save the column information and add another column. When you are finished adding columns, click **OK** to close the **FIELD** window.



- 7. Configure the Form Layout.
  - a. Click the **FORM LAYOUT** tab.
  - b. Select the fields and move their positions using the arrow buttons.



- c. Click **Preview** to see a representation of the final positioning. Note that the **Preview** loads a window in the **Workbench**, but the actual table component will only be available to users in the standard **Kintana** interface (**HTML**).
8. Configure any **Table** logic in the **RULES** tab. Rules are used for advanced defaulting behavior and calculating column totals.
  - a. Click the **RULES** tab.
  - b. Click **NEW** to define a new rule. See *“Creating a Table Rule”* on page 276 for detailed instructions.
9. Click **OK** to save the **Validation**.

The new **Table Component** field can be included on a **Request Type**, **Request Header Type** or **Request User Data** field.

### *Creating a Table Rule*

Table rules are configured in the same manner as advanced **Request Type** rules. Essentially, you can configure fields (columns) in the table to default to certain values based on an event or value in another field in the table. Because the table component rules are configured using a **SQL** statement, you are given enormous flexibility for the data that is populated in the table cells.

Table rules are configured using the **RULES** tab on the **VALIDATION** window.



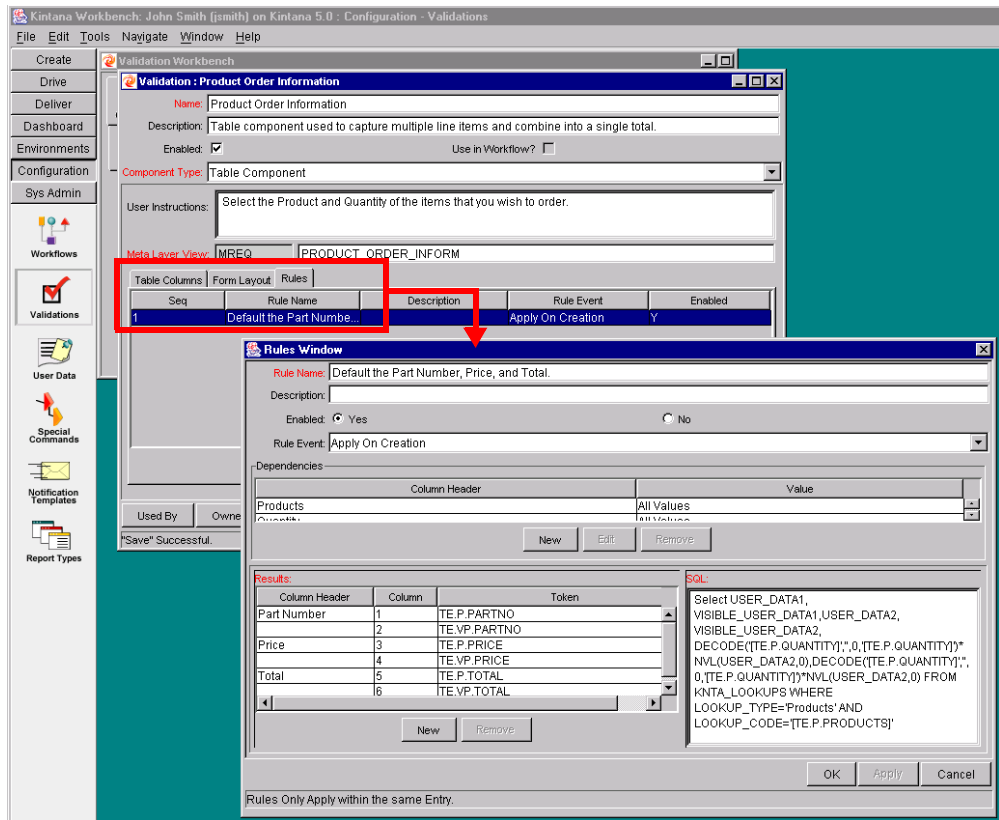


Figure 0-7 Rules window accessed from the Rules tab

### Example: Using a Table Component on an Order Form

The following example illustrates the table component rules functionality.

ACME, Inc. uses a Request for creating and tracking employee computer hardware equipment orders. ACME has included a table component field on their Request Type for gathering the order information. When the employee selects a Product, the Unit Price is automatically updated. Then, when they update the Quantity, the total line cost is automatically calculated and displayed in the table.

To enable this functionality, ACME first has to configure a new Validation with the following specifications:

Table 0-10. Example - Table Component Validation Settings

Setting	Value / Description
Validation Name	Product Order Information
Component Type	Table Component
Column 1	Column Header = Products Column Token = PRODUCTS Validation = Auto complete list with the following list values: PC, MOUSE, MONITOR, KEYBOARD
Column 2	Column Header = Quantity Column Token = QUANTITY Validation = Numeric Text Field
Column 3	Column Header = Price Column Token = PRICE Validation = Numeric Text Field
Column 4	Column Header = Total Column Token = TOTAL Validation = Numeric Text Field

**Validation : Product Order Information**

Name:

Description:

Enabled:  Use in Workflow?

Component Type:

User Instructions:

Meta Layer View:

Table Columns | Form Layout | Rules

Column Seq.	Column Header	Column Token	Parameter Col.	Enabled	Component Type	Validation
1	Products	PRODUCTS	PARAMETER5	Y	Auto Complete List	Product List for Order Form
2	Quantity	QUANTITY	PARAMETER3	Y	Text Field	Numeric Text Field
3	Price	PRICE	PARAMETER1	Y	Text Field	Numeric Text Field
4	Total	TOTAL	PARAMETER4	Y	Text Field	Numeric Text Field

Buttons: ↑ ↓ New Edit Remove

Used By: Ownership

Buttons: OK Save Cancel

Ready

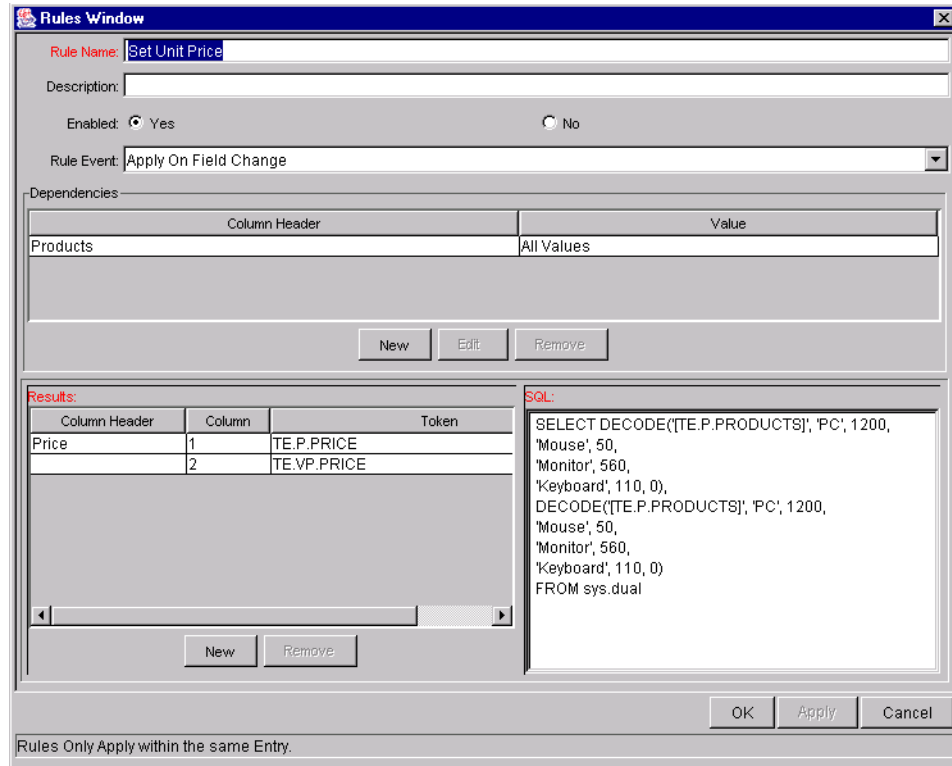
Once the Validation's columns have been defined, the Rules can be configured:

**Rule 1: Set Unit Price.**

ACME uses the following rule to set the default unit price in the PRICE cell based on the PRODUCT selection.

*Table 0-11. Example - Set Unit Price Rule Settings*

Setting	Value / Description
Rule Name	Set Unit Price
Rule Event	Apply on Field Change
Dependencies	Column = Products All Values = Yes
Results	Column Header = Price
SQL	<pre>SELECT DECODE(['TE.P.PRODUCTS'], 'PC', 1200, 'Mouse', 50, 'Monitor', 560, 'Keyboard', 110, 0), DECODE(['TE.P.PRODUCTS'], 'PC', 1200, 'Mouse', 50, 'Monitor', 560, 'Keyboard', 110, 0) FROM sys.dual</pre>



**Rule 2: Set Unit Price.**

ACME uses the following rule to set the calculate and display the total line price in the TOTAL column based on the values in the PRODUCTS and QUANTITY cells.

*Table 0-12. Example - Calculate Total Rule Settings*

Setting	Value / Description
Rule Name	Set Unit Price
Rule Event	Apply on Field Change
Dependencies	Column = Price [All Values = Yes] Column = Quantity [All Values = Yes]
Results	Column Header = Total
SQL	SELECT [TE.P.PRICE] * [TE.P.QUANTITY], [TE.P.PRICE] * [TE.P.QUANTITY] from sys.dual

## Using the Table Component

Add a field to a Request Type that is validated by this Table Component Validation. When a user opens the field to enter information, the table rules will be applied to each row that is created.

## Tokens in the Table Components

Each column included in the table component has an associated Token. These Tokens can be used in the same manner as other field tokens in Kintana, such as for commands, notifications or advanced field defaulting. See ["Using Commands and Tokens"](#) for details on referencing Tokens related to Table Components.

## Calculating Column Totals

You can configure columns that are validated by a number to calculate the total for that column. This is configured in the Validation's FIELD window. The following example illustrates how to configure a column to calculate and display the column total.

ACME, Inc. uses a Request for creating and tracking simple employee equipment orders. ACME has included a table component field on their Request Type for gathering the order information. Employee enter the Purchase Items and Cost for each item. The table component automatically calculates the total cost for the Cost column.

ACME creates a Validation with the following settings:

- COMPONENT TYPE = **TABLE COMPONENT**.
- Column 1 = Purchase Item (text field)
- Column 2 = Cost (number). In the FIELD window for the COST column, select DISPLAY TOTAL = **Yes**. The DISPLAY TOTAL field is only enabled if the field's validation is a number.

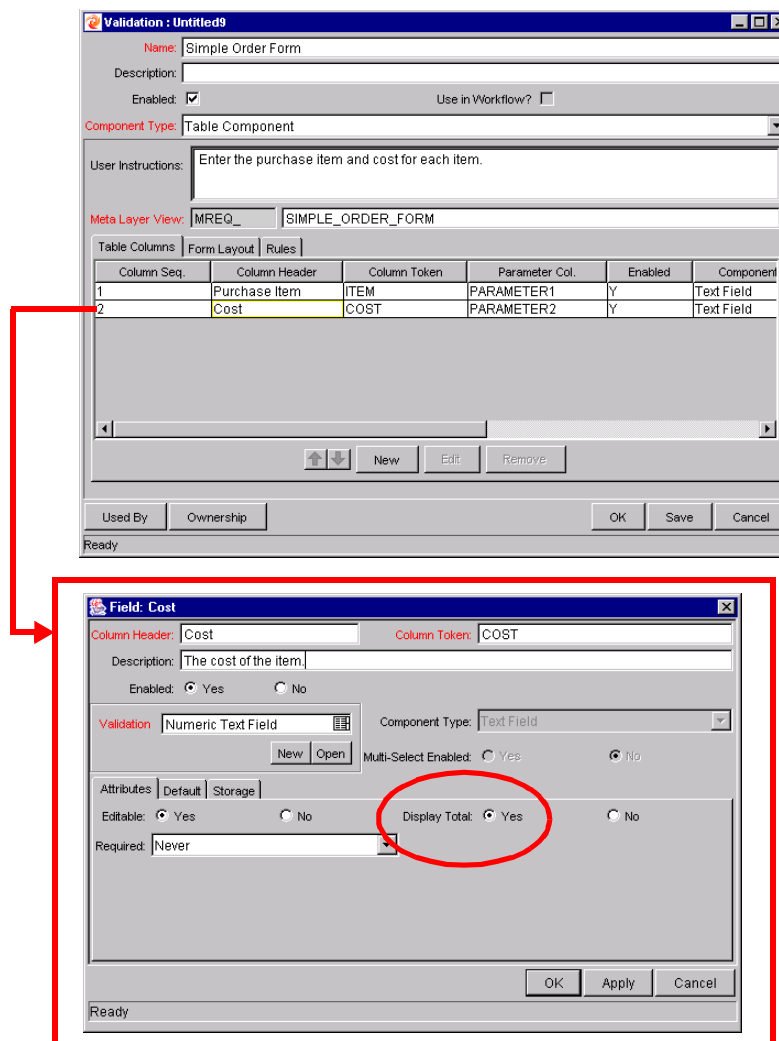


Figure 0-8 Sample validation for a Simple Order table component.

ACME includes adds a field to their Order Request Type that uses this Validation. When a user creates a Request using that Request Type, he can

click on the table component icon next to the field to open the order form. The total for the COST column is displayed at the bottom of the table.

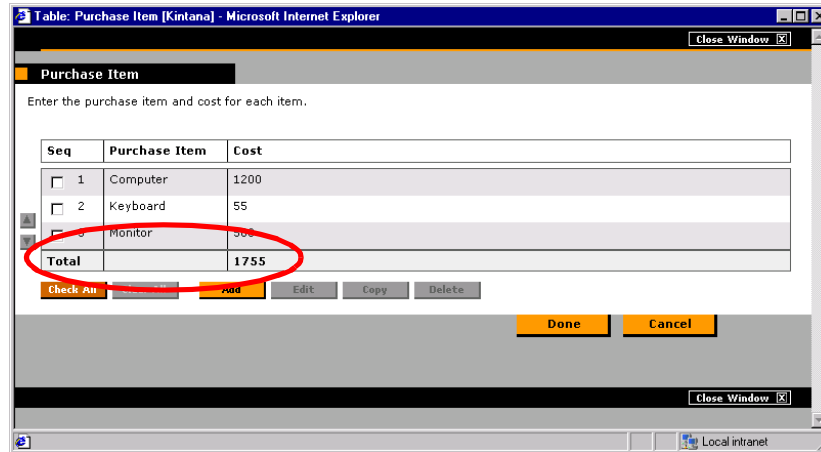


Figure 0-9 Sample table component displaying a column total.

## Add the Table Component to a Request Type

Table Component fields can be included on a Request Type, Request Header Type or Request User Data field.

To add a Table Component field to a Request Type:

1. Open the REQUEST TYPE window.
2. Click NEW in the **FIELDS** tab. The FIELD window opens.
3. Enter the FIELD PROMPT, TOKEN, and DESCRIPTION.
4. In the Validation field, select a table component Validation. If you have not created a table component Validation, click **NEW** to create one. See *“Define the Table Component in the Validation Workbench”* on page 273 for instructions.

## Configuring a Request Resolution System

Field: New

Field Prompt: Hardware Information Table      Token: HARDWAREINFO

Description: Table Component for entering hardware information.

Enabled:  Yes     No

Validation: Hardware Information      Component Type: Table Component

New    Open

Multiselect:  Yes     No

Attributes    Default    Storage    Security

Section Name: Request Type Fields      Display Only:  Yes     No

Transaction History:  Yes     No      Notes History:  Yes     No

Display on Search and Filter:  Yes     No      Display:  Yes     No

Copy From...      OK    Add    Cancel

Ready

5. Click **OK** to add the field to the Request Type.

6. Save the Request Type.

The table component field will now appear on Requests of this Request Type.

KINTANA

Create A Request > Create New Hardware Request Type

Welcome John Smith

Create New Hardware Request Type

Expand All    Collapse All

Submit    Cancel

Header

Tab #1

Created By: jsmith

Department: Manufacturing      Sub-Type:

Workflow:

Priority: High      Application:

Assigned To:      Assigned Group:

Request Group:

Contact Name:

Contact Phone:

Contact Email:

Description: Requesting new hardware for John Smith.

Request Type Fields

Hardware Information Table    2 Entries

Details

Notes

Hardware Information Table

Add a row for each hardware item that you are requesting.

Seq	Part Number	Sub-Type	Part	Unit Price
<input type="checkbox"/> 1	100897	18 inch	Monitor	\$320.00
<input type="checkbox"/> 2	899768		Keyboard	\$60.00

Check All    Clear All    Add    Edit    Copy    Delete

Done    Cancel



## Package and Request Group Validations

Two particular entity-specific Validations can be accessed in the Kintana Workbench without entering the VALIDATIONS screen group:

- *Package and Request Groups*
- *Request Type Category*

### Package and Request Groups

The KNTA-Package and Request Groups Validation can be accessed directly from the **PACKAGE** screen. To specify that a Package belongs to a new or unique Package Group that is not named in the auto-complete Validation list, it is not necessary to proceed through the VALIDATION WORKBENCH.

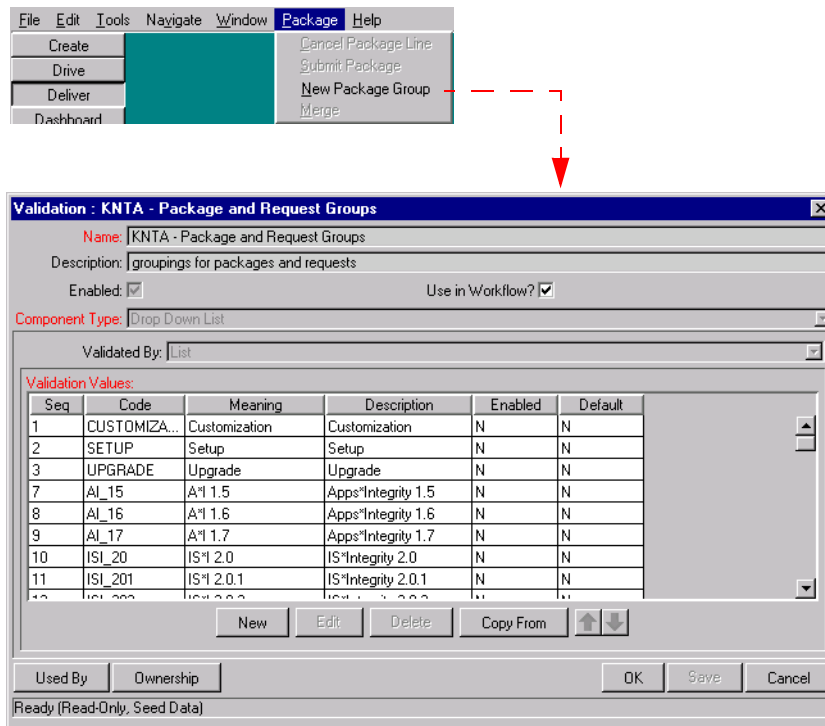
To access the KNTA-Package and Request Groups Validation window from the **PACKAGE** screen:

Select **NEW PACKAGE GROUP** from the **PACKAGE** menu. The VALIDATION window will appear, listing the existing Kintana Deliver Package Groups.



Note

All users are granted read access to this screen, but only users with appropriate security privileges can alter the KNTA-Package and Request Groups Validation list.



## Request Type Category

The CRT - REQUEST TYPE CATEGORY Validation can be accessed directly from the **REQUEST TYPES** screen.

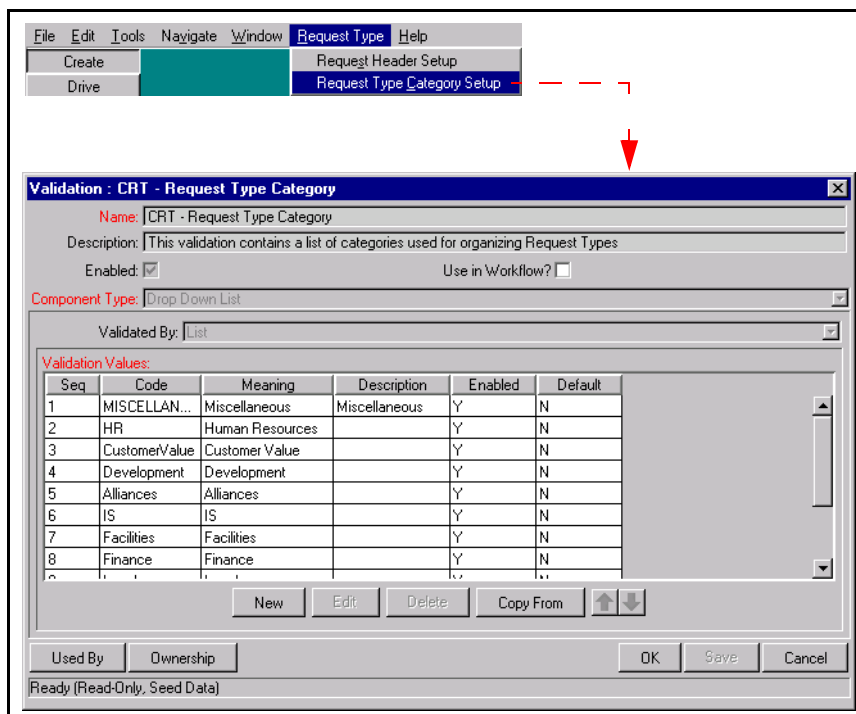
To access the CRT - REQUEST TYPE CATEGORY Validation window from the **REQUEST TYPES** screen:

Select **REQUEST TYPE CATEGORY SETUP** from the **REQUEST TYPE** menu. The **VALIDATION** window will appear, listing the existing Kintana Request Type Categories.



Note

All users are granted read access to this screen, but only users with appropriate security privileges can alter the CRT - Request Type Category Validation list.



## Validation Special Characters

The VALIDATION NAME field for all Validations cannot contain a question mark ('?'). The Kintana Workbench prevents this character from being entered into the field, but all previously configured Validation Names (Validations entered before Kintana release 4.5) should be checked and corrected.

## System Validations

The following is a list of the default validations that are installed with Kintana. Note that many of these validations may have been altered to better match your company's specific business needs. The table contains the following data:

- Validation Name
- Component Type: the type of field that the Validation represents
- Use in Workflows: whether the Validation is currently enabled for use in Workflows.



Use the Validations report to get a list of all validations currently in your system. This includes information such as the validations' values and any SQL used to generate the values.

Table 0-13. Kintana System Validations

Validation Name	Component Type	Use in Workflow?
All Projects - Cost Enabled and Not New	Auto Complete List	N
Application Module	Drop Down List	N
Attachment	Attachment	N
CONNECTION_PROTOCOL	Drop Down List	N
CRT - All DSS Report Types	Auto Complete List	N
CRT - All Regular Report Types	Auto Complete List	N
CRT - Application - Enabled	Auto Complete List	N
CRT - Assigned Group - All	Auto Complete List	N
CRT - Assigned Group - Enabled	Auto Complete List	N
CRT - Assigned Group - Participant	Auto Complete List	N
CRT - Assigned To - Enabled	Auto Complete List	N
CRT - Assigned To - Participant	Auto Complete List	N
CRT - Company	Auto Complete List	N
CRT - Company - All	Auto Complete List	N
CRT - Contact Email - Enabled	Auto Complete List	N
CRT - Contact Email - Restricted Enabled	Auto Complete List	N
CRT - Contact Email by Company - Enabled	Auto Complete List	N
CRT - Contact Name - All	Auto Complete List	N
CRT - Contact Name - Enabled	Auto Complete List	N
CRT - Contact Name - Restricted	Auto Complete List	N
CRT - Contact Name - Restricted Enabled	Auto Complete List	N
CRT - Contact Name by Company - Enabled	Auto Complete List	N
CRT - Contact Phone - Enabled	Auto Complete List	N
CRT - Contact Phone - Restricted Enabled	Auto Complete List	N
CRT - Contact Phone by Company - Enabled	Auto Complete List	N
CRT - Contact Synch Driver	Drop Down List	N
CRT - DSS Report Types	Auto Complete List	N
CRT - Department - Enabled	Drop Down List	N
CRT - Difficulty	Drop Down List	N
CRT - Dynamic Reporting Column List	Auto Complete List	N
CRT - Enhancement Request Category	Drop Down List	N
CRT - Impact	Drop Down List	N
CRT - Max Custom Fields	Drop Down List	N
CRT - Modification Type	Drop Down List	N

Table 0-13. Kintana System Validations

Validation Name	Component Type	Use in Workflow?
CRT - Platform	Drop Down List	N
CRT - Priority - All	Drop Down List	N
CRT - Priority - Enabled	Drop Down List	Y
CRT - Priority - No Default	Drop Down List	Y
CRT - Priority AutoComplete	Auto Complete List	N
CRT - Prj Code - Enabled	Auto Complete List	N
CRT - Regular Report Types	Auto Complete List	N
CRT - Request Activity Portlet Group By	Drop Down List	N
CRT - Request Analyzed	Drop Down List	Y
CRT - Request Assigned	Drop Down List	Y
CRT - Request Detail Report Order By	Drop Down List	N
CRT - Request Group - All	Auto Complete List	N
CRT - Request Group - Enabled	Auto Complete List	N
CRT - Request Header Type Name (Migrator Source)	Auto Complete List	N
CRT - Request Header Types - All	Auto Complete List	N
CRT - Request Header Types - Enabled	Auto Complete List	N
CRT - Request Held	Drop Down List	Y
CRT - Request In Progress	Drop Down List	Y
CRT - Request Info Required	Drop Down List	Y
CRT - Request List	Auto Complete List	N
CRT - Request List Narrow Sort By	Drop Down List	N
CRT - Request List Wide Sort By	Drop Down List	N
CRT - Request List Wide Status	Auto Complete List	N
CRT - Request Listing Report Columns	Auto Complete List	N
CRT - Request Quick View Order By	Drop Down List	N
CRT - Request Reference Type	Drop Down List	N
CRT - Request Reviewed	Drop Down List	Y
CRT - Request Summary Group By	Auto Complete List	N
CRT - Request Summary Group By Types	Drop Down List	N
CRT - Request Type Category	Drop Down List	N
CRT - Request Type Fields	Auto Complete List	N
CRT - Request Type Fields - All	Auto Complete List	N
CRT - Request Type Name (Migrator Source)	Auto Complete List	N
CRT - Request Type Names - All	Auto Complete List	N
CRT - Request Type Names - Restricted by Package Workflow	Drop Down List	N
CRT - Request Type Notification Fields	Auto Complete List	N
CRT - Request Type Prompt - All	Auto Complete List	N
CRT - Request Type Restriction	Drop Down List	N
CRT - Request Type Status (Partial)	Auto Complete List	N
CRT - Request Type Status (Partial) REQ tokens	Auto Complete List	N
CRT - Request Type Status - All	Auto Complete List	N

Table 0-13. Kintana System Validations

Validation Name	Component Type	Use in Workflow?
CRT - Request Types - All	Auto Complete List	N
CRT - Request Types - All.	Auto Complete List	N
CRT - Request Types - Enabled	Auto Complete List	N
CRT - Request Types - Enabled.	Auto Complete List	N
CRT - Request Types - Interface Restricted	Auto Complete List	N
CRT - Request Types - Restricted	Auto Complete List	N
CRT - Request Types - Restricted Enabled	Auto Complete List	N
CRT - Request Types w. Description - Interface Restricted	Auto Complete List	N
CRT - Requests - All	Auto Complete List	N
CRT - Resolution	Drop Down List	Y
CRT - Rule Dependencies Fields	Auto Complete List	N
CRT - Rule Results Fields	Auto Complete List	N
CRT - Security Group With Description	Auto Complete List	N
CRT - Sort By	Drop Down List	N
CRT - Sub Types - All	Auto Complete List	N
CRT - SubTypes - All	Auto Complete List	N
CRT - SubTypes - Enabled	Auto Complete List	N
CRT - Validations - Enabled	Auto Complete List	N
CRT - Workflow Id - All	Auto Complete List	N
CRT - Workflow With Description	Auto Complete List	N
CRT - Workflows - Enabled	Auto Complete List	N
CRT - Workflows - Restricted	Auto Complete List	N
CRT Request - Queryable Fields	Auto Complete List	N
CST - All Budget Status	Drop Down List	N
CST - Budget Entities	Auto Complete List	N
CST - Budget Fiscal Periods	Auto Complete List	N
CST - Budget For Types	Drop Down List	N
CST - Budget Line Type	Drop Down List	N
CST - Budget Programs	Auto Complete List	N
CST - Budget Projects	Auto Complete List	N
CST - Budget Rolls Up To Types	Drop Down List	N
CST - Budget Search Sort By	Drop Down List	N
CST - Budgets Line Category	Auto Complete List	N
CST - Fiscal Quarters	Drop Down List	N
CST - Parent Org Unit Budgets	Auto Complete List	N
CST - Parent Program Budgets	Auto Complete List	N
CST - Parent Project Budgets	Auto Complete List	N
CST - Program Names	Auto Complete List	N
CST - Specific Entity Linked Budgets	Auto Complete List	N
Component Type	Drop Down List	N
DEM - All Assignable Users	Auto Complete List	N

Table 0-13. Kintana System Validations

Validation Name	Component Type	Use in Workflow?
DEM - Assignment Queue Order By	Auto Complete List	N
DEM - Demand Categories with Disposition	Auto Complete List	N
DEM - Demand Disposition	Auto Complete List	N
DEM - Demand Disposition Open and Satisfied	Auto Complete List	N
DEM - Demand Disposition Outstanding	Auto Complete List	N
DEM - Demand Fields	Auto Complete List	N
DEM - Demand Fields only from Demand Set view name	Auto Complete List	N
DEM - Demand Fields with Disposition	Auto Complete List	N
DEM - Demand Fields with Disposition & Request Type	Auto Complete List	N
DEM - Demand Set Views	Auto Complete List	N
DEM - Demand Sets - Enabled	Drop Down List	N
DEM - Demand Sets Request Types	Auto Complete List	N
DEM - Request Held	Drop Down List	N
DEM - Request Type Statuses	Auto Complete List	N
DEM - Request Types of a Demand Set	Auto Complete List	N
DEM - SLA Level	Auto Complete List	N
DEM - Search Validations - All	Auto Complete List	N
DEM Filter - User Id - with empty - when no category	Auto Complete List	N
DIST - Workflow Id - Enabled	Auto Complete List	N
DLV - Accelerator - Enabled	Drop Down List	N
DLV - Accelerator Panel, Env Screen - Enabled	Auto Complete List	N
DLV - All DSS Report Types	Auto Complete List	N
DLV - All Regular Report Types	Auto Complete List	N
DLV - Assigned Group - Enabled	Auto Complete List	N
DLV - Assigned Group - Participant	Auto Complete List	N
DLV - Assigned Group - Restricted	Auto Complete List	N
DLV - Assigned To - Participant	Auto Complete List	N
DLV - Assigned To - Restricted	Auto Complete List	N
DLV - DSS Report Types	Auto Complete List	N
DLV - Database Type	Drop Down List	N
DLV - Execution Order	Drop Down List	N
DLV - File Location	Drop Down List	N
DLV - File Type	Drop Down List	N
DLV - Files of Type	Drop Down List	N
DLV - Kintana Server Directory Chooser	Directory Chooser	N
DLV - Kintana Server File Chooser	File Chooser	N
DLV - ODF Mode	Drop Down List	N
DLV - Object Category - Enabled	Drop Down List	N
DLV - Object Category -All	Drop Down List	N
DLV - Object History Order By	Drop Down List	N
DLV - Object Name - All	Auto Complete List	N

Table 0-13. Kintana System Validations

Validation Name	Component Type	Use in Workflow?
DLV - Object Type - All	Auto Complete List	N
DLV - Object Type - Enabled	Auto Complete List	Y
DLV - Object Type ID - Enabled	Auto Complete List	N
DLV - Object Type ID - Restricted	Auto Complete List	N
DLV - Object Type Id - All	Auto Complete List	N
DLV - Object Type Name (Migrator Source)	Auto Complete List	N
DLV - Package Activity Portlet Group By	Drop Down List	N
DLV - Package Group - All	Auto Complete List	N
DLV - Package Group - Enabled	Auto Complete List	N
DLV - Package ID	Auto Complete List	N
DLV - Package List	Auto Complete List	N
DLV - Package List Portlet Narrow Sort By	Drop Down List	N
DLV - Package List Portlet Wide Sort By	Drop Down List	N
DLV - Package Number	Auto Complete List	N
DLV - Package Pending Filter	Drop Down List	N
DLV - Package Pending Order By	Drop Down List	N
DLV - Package Priority - All	Drop Down List	N
DLV - Package Priority - Enabled	Drop Down List	Y
DLV - Package Status - All	Auto Complete List	N
DLV - Package Type - All	Drop Down List	N
DLV - Package Type - Enabled	Drop Down List	N
DLV - Patch Env Order By	Drop Down List	N
DLV - Patch Object Type Names	Drop Down List	N
DLV - Priority AutoComplete	Auto Complete List	N
DLV - Regular Report Types	Auto Complete List	N
DLV - Release Number	Auto Complete List	N
DLV - SQL Paths	Drop Down List	N
DLV - Sort By	Drop Down List	N
DLV - Workflow Id - All	Auto Complete List	N
DLV - Workflow Id - Enabled	Auto Complete List	N
DLV - Workflow Id - Restricted	Auto Complete List	N
DLV - Workflow Id - Restricted Enabled	Auto Complete List	N
DLV Package - Queryable Fields	Auto Complete List	N
DRV - Activity Priority - Enabled	Drop Down List	N
DRV - Activity Status - Enabled	Drop Down List	N
DRV - Add User To Task	Auto Complete List	N
DRV - All DSS Report Types	Auto Complete List	N
DRV - All Project List	Auto Complete List	N
DRV - All Regular Report Types	Auto Complete List	N
DRV - Baseline for Comparision	Auto Complete List	N
DRV - Booked Skill	Auto Complete List	N



Table 0-13. Kintana System Validations

Validation Name	Component Type	Use in Workflow?
DRV - Calendar Reason	Drop Down List	N
DRV - Confidence	Drop Down List	N
DRV - Custom Fields base on Project Template	Auto Complete List	N
DRV - DSS Report Types	Auto Complete List	N
DRV - Exception Rule Types	Auto Complete List	N
DRV - Filter Reference Type	Drop Down List	N
DRV - Filter Relationship	Drop Down List	N
DRV - Interface Resources Tables	Auto Complete List	N
DRV - Master Projects - Enabled	Auto Complete List	N
DRV - Master Projects - Enabled and not New	Auto Complete List	N
DRV - Master Projects with Baselines - Enabled	Auto Complete List	N
DRV - My Tasks Sort By	Auto Complete List	N
DRV - Proj Manager - Restricted	Auto Complete List	N
DRV - Project Detail Report Order By	Auto Complete List	N
DRV - Project Fields	Auto Complete List	N
DRV - Project Grouping Type	Drop Down List	N
DRV - Project Header Fields	Auto Complete List	N
DRV - Project List Portlet Narrow Sort By	Drop Down List	N
DRV - Project List Portlet Wide Sort By	Drop Down List	N
DRV - Project Names - All	Auto Complete List	N
DRV - Project Names - All - Depend on [P_SHOW_MASTER_ONLY]	Auto Complete List	N
DRV - Project Names - In Templates	Auto Complete List	N
DRV - Project Names by Template	Auto Complete List	N
DRV - Project State Search	Auto Complete List	N
DRV - Project States	Auto Complete List	N
DRV - Project Status - All	Auto Complete List	N
DRV - Project Task Aging Report Order By	Auto Complete List	N
DRV - Project Team Resource	Auto Complete List	N
DRV - Project Template Names	Auto Complete List	N
DRV - Project Template Names - Enabled	Auto Complete List	N
DRV - Project Template with TemplateID and ParameterSetContextID	Auto Complete List	N
DRV - Project Type Names - All	Auto Complete List	N
DRV - Project Type Prompt - All	Auto Complete List	N
DRV - Project Type Status - All	Auto Complete List	N
DRV - Project Types - All	Auto Complete List	N
DRV - Project Types - Enabled	Auto Complete List	N
DRV - Project User Data Roll-Up Fields	Auto Complete List	N
DRV - Projects (Only) - Enabled	Auto Complete List	N
DRV - Projects - All	Auto Complete List	N
DRV - Projects - Enabled	Auto Complete List	N

Table 0-13. Kintana System Validations

Validation Name	Component Type	Use in Workflow?
DRV - Regular Report Types	Auto Complete List	N
DRV - Request Priority - All	Auto Complete List	N
DRV - Request Reference Relationships	Drop Down List	N
DRV - Resource Group - Restricted Enabled	Auto Complete List	N
DRV - Resource Names - Proj Teams	Auto Complete List	N
DRV - Resource or Project Manager	Auto Complete List	N
DRV - Scheduling Constraints	Drop Down List	N
DRV - Security Restrictions	Drop Down List	N
DRV - Sort By for Projects	Drop Down List	N
DRV - Sort By for Tasks	Drop Down List	N
DRV - Sub Project Ids - Enabled	Auto Complete List	N
DRV - Sub Projects - Enabled	Auto Complete List	N
DRV - Subproject Name for My Tasks	Auto Complete List	N
DRV - Summary Condition	Auto Complete List	N
DRV - Task Categories	Drop Down List	N
DRV - Task Categories - AutoComp	Auto Complete List	N
DRV - Task Exception Type Names	Auto Complete List	N
DRV - Task Ids - Enabled	Auto Complete List	N
DRV - Task Name - Enabled	Auto Complete List	N
DRV - Task Name - In Templates	Auto Complete List	N
DRV - Task Notification Dates	Auto Complete List	N
DRV - Task State Search	Auto Complete List	N
DRV - Task States	Auto Complete List	N
DRV - Task States for My Tasks	Auto Complete List	N
DRV - Task States for Notifications	Drop Down List	N
DRV - Task User Data Roll-Up Fields	Auto Complete List	N
DRV - Task and Project #s - Enabled	Auto Complete List	N
DRV - Task and Project Names - Enabled	Auto Complete List	N
DRV - Tasks - All	Auto Complete List	N
DRV - Unlinked Packages for Drive	Auto Complete List	N
DRV - Unlinked Project References for Drive	Auto Complete List	N
DRV - Unlinked Requests for Drive	Auto Complete List	N
DRV - Workflow - All	Auto Complete List	N
DRV - Workflow - Enabled	Auto Complete List	N
DSH - Column Type	Drop Down List	N
DSH - Hyperlink Type	Drop Down List	N
DSH - Portlet Category	Drop Down List	N
DSH - Portlet Category with ALL	Drop Down List	N
DSH - Portlet Names - All	Auto Complete List	N
DSH - Portlet Types in Category	Auto Complete List	N
DSH - Portlet Width	Drop Down List	N

Table 0-13. Kintana System Validations

Validation Name	Component Type	Use in Workflow?
DSH - Project References Portlet Reference Type	Auto Complete List	N
DSH - Project References Portlet Sort By	Drop Down List	N
DSH - Time-Out	Drop Down List	N
DSH - Validations - Enabled	Auto Complete List	N
DSS - Chart Type	Auto Complete List	N
DSS - Migration Status	Drop Down List	N
DSS - Object Migration Groupings	Drop Down List	N
DSS - Package Line Grouping Types	Drop Down List	N
DSS - Period End Dates	Auto Complete List	N
DSS - Period Start Dates	Auto Complete List	N
DSS - Period Types	Drop Down List	N
DSS - Request Grouping Types	Drop Down List	N
DSS - Workflows - All	Auto Complete List	N
Dashboard - All Users - Fullname	Auto Complete List	N
Data Mask	Drop Down List	Y
Date	Date Field	Y
Date (Short Format)	Date Field	N
Date Format	Drop Down List	Y
Debug Level	Drop Down List	N
Default Type	Drop Down List	N
Directory Chooser	Directory Chooser	N
ENV - All DB Environments	Auto Complete List	N
ENV - App Code - Enabled	Drop Down List	N
ENV - App Code - Restricted	Drop Down List	N
ENV - Compared App Codes	Auto Complete List	N
ENV - Comparison Types	Drop Down List	N
ENV - Custom Objects	Auto Complete List	N
ENV - Env Group Id - Access Specific	Auto Complete List	N
ENV - Env Group Id - Enabled	Auto Complete List	N
ENV - Environment Id - Access Specific	Auto Complete List	N
ENV - Environment Id - All	Auto Complete List	N
ENV - Environment Id - Enabled	Auto Complete List	N
ENV - Environment Id - OM installed	Auto Complete List	N
ENV - Environment Name - All	Auto Complete List	N
ENV - Environment Server/Client Type	Drop Down List	N
ENV - Filesystem Environments	Auto Complete List	N
ENV - Filesystem Exclusion Choices	Drop Down List	N
ENV - MS SQLServer Environments	Auto Complete List	N
ENV - Oracle Environments	Auto Complete List	N
ENV - Reference App Codes	Auto Complete List	N
ENV - Tier	Drop Down List	N

Table 0-13. Kintana System Validations

Validation Name	Component Type	Use in Workflow?
File Chooser - Base File Name	File Chooser	N
File Chooser - Full File Name	File Chooser	N
Gantt Overview Portlet Display and Tooltip label fields	Auto Complete List	N
Gantt Portlet Display and Tooltip label fields	Auto Complete List	N
Gantt Portlet Sort By	Drop Down List	N
KCST - Budget Names	Budget	N
KCST - Budget Names - All	Auto Complete List	N
KCST - Existing Org Unit Budgets	Auto Complete List	N
KCST - Existing Program Budgets	Auto Complete List	N
KCST - Existing Projects Budgets	Auto Complete List	N
KCST - New Org Unit Budgets	Auto Complete List	N
KCST - New Program Budgets	Auto Complete List	N
KCST - New Projects Budgets	Auto Complete List	N
KDRV - Project Predecessor Types	Drop Down List	N
KNTA - Authentication Mode All	Drop Down List	N
KNTA - Access Grant - All	Auto Complete List	N
KNTA - Application - Enabled	Auto Complete List	N
KNTA - Applications - All	Auto Complete List	N
KNTA - Autocomp Validation Type	Drop Down List	N
KNTA - Budgets	Auto Complete List	N
KNTA - Budgets - All	Auto Complete List	N
KNTA - Department - All	Drop Down List	N
KNTA - Department - Enabled	Drop Down List	N
KNTA - Departments - AutoComp	Auto Complete List	N
KNTA - Dependency Rule Usage	Drop Down List	N
KNTA - Dept - All	Auto Complete List	N
KNTA - Dropdown by List	Auto Complete List	N
KNTA - Enabled Combo	Drop Down List	N
KNTA - Entities for Notification History Report	Drop Down List	N
KNTA - FLS Denormalization Entity State	Drop Down List	N
KNTA - FLS Denormalization User Data Type	Drop Down List	N
KNTA - Field Entities	Auto Complete List	N
KNTA - Field Prompts	Auto Complete List	N
KNTA - Field Security Standard Tokens	Auto Complete List	N
KNTA - Field Security Types	Drop Down List	N
KNTA - Field Tokens	Auto Complete List	N
KNTA - Field Validations	Auto Complete List	N
KNTA - Finish Periods of type Fiscal Month	Auto Complete List	N
KNTA - Kintana Migrator Action	Drop Down List	N
KNTA - Kintana Server Names	Auto Complete List	N
KNTA - Kintana objects from migration source instance	Auto Complete List	N

Table 0-13. Kintana System Validations

Validation Name	Component Type	Use in Workflow?
KNTA - LDAP Import Authentication Modes	Drop Down List	N
KNTA - List Validations - All	Auto Complete List	N
KNTA - Long Date Formats	Drop Down List	N
KNTA - Lookup Types - All	Auto Complete List	N
KNTA - Medium Date Formats	Drop Down List	N
KNTA - Meta Layer Action	Drop Down List	N
KNTA - Meta Layer Scope	Drop Down List	N
KNTA - Meta Layer Views Autocomp	Auto Complete List	N
KNTA - Meta Layer view definition template file chooser	Auto Complete List	N
KNTA - Migrator Internationalization Modes	Drop Down List	N
KNTA - Migrator Source Types	Drop Down List	N
KNTA - Notification Formats	Drop Down List	N
KNTA - Notification Recipient Type Choices	Drop Down List	N
KNTA - Notification Recipient Type Code	Drop Down List	N
KNTA - Notification Types	Drop Down List	N
KNTA - Org Unit Linked Security Groups	Auto Complete List	N
KNTA - Org Unit Linked Security Groups - Editable	Auto Complete List	N
KNTA - Organization Unit Names - Enabled	Auto Complete List	N
KNTA - Package and Request Groups	Drop Down List	N
KNTA - Period Types	Drop Down List	N
KNTA - Period Types - All	Drop Down List	N
KNTA - Periods of type Fiscal Month	Auto Complete List	N
KNTA - Product Scope	Drop Down List	N
KNTA - Products	Auto Complete List	N
KNTA - Query Condition	Drop Down List	N
KNTA - Refresh Group Status	Drop Down List	N
KNTA - Relationships for Projects/Tasks	Auto Complete List	N
KNTA - Report Recurrence Pattern	Drop Down List	N
KNTA - Report Submission Status	Drop Down List	N
KNTA - Report Type - All	Auto Complete List	N
KNTA - Report View Access	Drop Down List	N
KNTA - Rule Events	Auto Complete List	N
KNTA - Security Group Id - Access Specific	Auto Complete List	N
KNTA - Security Group Id - All	Auto Complete List	N
KNTA - Security Group Id - Editable	Auto Complete List	N
KNTA - Security Group Id - Enabled	Auto Complete List	N
KNTA - Security Group Name - All	Auto Complete List	N
KNTA - Short Date Formats	Drop Down List	N
KNTA - Shortcut Bar Location	Drop Down List	N
KNTA - Solutions - All	Auto Complete List	N
KNTA - Special Command Names - All	Auto Complete List	N

Table 0-13. Kintana System Validations

Validation Name	Component Type	Use in Workflow?
KNTA - Special Command Type Name (Migrator Source)	Auto Complete List	N
KNTA - Special Commands - Enabled	Auto Complete List	N
KNTA - Specific Entity	Auto Complete List	N
KNTA - Table Comp Rule Dependencies Fields	Auto Complete List	N
KNTA - Table Comp Rule Results Fields	Auto Complete List	N
KNTA - True or False	Drop Down List	Y
KNTA - User Accelerator - Enabled	Auto Complete List	N
KNTA - User Data - All	Auto Complete List	N
KNTA - User Data Context Field	Auto Complete List	N
KNTA - User Data Type - All	Drop Down List	N
KNTA - User Data Type - Enabled	Auto Complete List	N
KNTA - User Data Validations - Enabled	Auto Complete List	N
KNTA - User Id - All	Auto Complete List	N
KNTA - User Id - Enabled	Auto Complete List	N
KNTA - User Names - All	Auto Complete List	N
KNTA - User Names - Enabled	Auto Complete List	N
KNTA - User Security Action	Drop Down List	N
KNTA - Validation Name (Migrator Source)	Auto Complete List	N
KNTA - Validation Names	Auto Complete List	N
KNTA - Validation Type	Drop Down List	N
KNTA - Validations (Exclude Table, Budget, Staffing Profile and Resource Pool)	Auto Complete List	N
KNTA - Validations (For Table Column Headers)	Auto Complete List	N
KNTA - Validations - All	Auto Complete List	N
KNTA - Validations - Lookups	Auto Complete List	N
KNTA - Workflow Name (Migrator Source)	Auto Complete List	N
KNTA - Workflow Steps	Auto Complete List	N
KNTA - Workflows - Enabled	Auto Complete List	N
KRSC - Organization Unit Member Action	Drop Down List	N
KRSC - Resource	Auto Complete List	N
KRSC - Resource Pool Names	Resource Pool	N
KRSC - Resource Pool Names - All	Auto Complete List	N
KRSC - Skill	Auto Complete List	N
KRSC - Staffing Profile Names	Staffing Profile	N
KRSC - Staffing Profile Names - All	Auto Complete List	N
Master Projects - Cost Enabled and Not New	Auto Complete List	N
Numeric Text Field	Text Field	Y
Numeric Text Field (length = 4)	Text Field	Y
Numeric Text Field - 10 decimals	Text Field	N
Numeric Text Field - 2 decimals	Text Field	N
PMO - Business Objective States	Drop Down List	Y

Table 0-13. Kintana System Validations

Validation Name	Component Type	Use in Workflow?
PMO - Business Objectives	Auto Complete List	N
PMO - CR Level	Drop Down List	Y
PMO - Issue Escalation Level	Drop Down List	N
PMO - Master Projects	Auto Complete List	N
PMO - Period	Drop Down List	N
PMO - Program Names	Auto Complete List	N
PMO - Program Projects	Auto Complete List	N
PMO - Program State	Drop Down List	N
PMO - Program State AC	Auto Complete List	N
PMO - Request Entities	Drop Down List	N
PMO - Risk Impact	Drop Down List	Y
PMO - Risk Probability	Drop Down List	Y
PMO - Scope Change Severity	Drop Down List	Y
PMO - Sort By for Programs	Drop Down List	N
PMO - Summary Condition Impact	Auto Complete List	N
PMO - Summary Condition Priority	Auto Complete List	N
PMO - Summary Condition Probability	Auto Complete List	N
PMO - Summary Condition Severity	Auto Complete List	N
Parameter Column	Drop Down List	N
Password Field	Password Field	N
RM - All Distributions By Name	Auto Complete List	N
RM - All Releases By ID	Auto Complete List	N
RM - Distribution Detail Order By	Drop Down List	N
RM - Object Types in Distribution	Auto Complete List	N
RM - Ready for Release	Drop Down List	Y
RM - Release Status	Drop Down List	N
RM - Releases - All	Auto Complete List	N
RM - Releases - Open	Auto Complete List	N
RM - Workflow Step Statuses For Step	Auto Complete List	N
RM Package - Filterable Fields	Auto Complete List	N
RSC - Location	Drop Down List	N
RSC - Location - Autocomp	Auto Complete List	N
RSC - Org Unit Category	Drop Down List	N
RSC - Org Unit Category - Autocomp	Auto Complete List	N
RSC - Org Unit ID - Enabled	Auto Complete List	N
RSC - Org Unit Name - All	Auto Complete List	N
RSC - Org Unit Sort By	Drop Down List	N
RSC - Org Units - Enabled (non-seeded)	Auto Complete List	N
RSC - Period Numbers	Drop Down List	N
RSC - Projects w Staffing Profiles	Auto Complete List	N
RSC - Resource Category	Drop Down List	N

Table 0-13. Kintana System Validations

Validation Name	Component Type	Use in Workflow?
RSC - Resource Category - Autocomp	Auto Complete List	N
RSC - Resource ID - Enabled	Auto Complete List	N
RSC - Resource Managers	Auto Complete List	N
RSC - Resource Pool	Auto Complete List	N
RSC - Resource Pool Sort By	Drop Down List	N
RSC - Resource Pool Status	Drop Down List	N
RSC - Resource Pool Status w/o Approved	Drop Down List	N
RSC - Resource Pools (restricted)	Auto Complete List	N
RSC - Resource Sort By	Drop Down List	N
RSC - Resource Title	Drop Down List	N
RSC - Resource Title - Autocomp	Auto Complete List	N
RSC - Resources - Enabled (non-seeded)	Auto Complete List	N
RSC - Skill Category	Drop Down List	N
RSC - Skill Name - All	Auto Complete List	N
RSC - Skill Proficiency	Drop Down List	N
RSC - Skills - Enabled	Auto Complete List	N
RSC - Staffing Profile Id - All	Auto Complete List	N
RSC - Staffing Profile Sort By	Drop Down List	N
RSC - Staffing Profile Status w/o Approved	Drop Down List	N
RSC - Status	Drop Down List	N
RSC - Vis - Assignment Load Group By	Drop Down List	N
RSC - Vis Resource Pool Group By	Drop Down List	N
RSC - Workload Category	Drop Down List	N
RTRULES_EXCLUDED_FIELDS	Drop Down List	N
Radio Buttons (Y/N)	Radio Button (Yes/No)	N
References	Drop Down List	N
Rule Types	Drop Down List	N
Sql Command	Drop Down List	N
Sub Paths	Drop Down List	N
TMG - Approval Types	Drop Down List	N
TMG - Approvals Search Order	Drop Down List	N
TMG - Charge Code Categories - All	Drop Down List	N
TMG - Charge Code Categories - Enabled	Drop Down List	N
TMG - Charge Code Filter Types	Drop Down List	N
TMG - Charge Code Id - All	Auto Complete List	N
TMG - Charge Code Id - Enabled	Auto Complete List	N
TMG - Clients - All	Auto Complete List	N
TMG - Clients - Enabled	Auto Complete List	N
TMG - Days of Week	Drop Down List	N
TMG - Managers	Auto Complete List	N
TMG - Master Projects - Enabled	Auto Complete List	N



Table 0-13. Kintana System Validations

Validation Name	Component Type	Use in Workflow?
TMG - Misc. Work Items	Drop Down List	N
TMG - Period Type Id - All	Auto Complete List	N
TMG - Period Type Id - Enabled	Auto Complete List	N
TMG - Resource Group Id - All	Auto Complete List	N
TMG - Resource Group Id - Enabled	Auto Complete List	N
TMG - Resource Id - All	Auto Complete List	N
TMG - Resource Id - Enabled	Auto Complete List	N
TMG - Resource Id - Restricted	Auto Complete List	N
TMG - Resource Pool Resource Group	Auto Complete List	N
TMG - Resource Types	Drop Down List	N
TMG - Time Entry Durations	Drop Down List	N
TMG - Time Entry Units	Drop Down List	N
TMG - Time Period Calculation Types	Drop Down List	N
TMG - Time Periods	Auto Complete List	N
TMG - Time Sheet Details - Work Item	Auto Complete List	N
TMG - Time Sheet Search Order	Drop Down List	N
TMG - Time Sheet Statuses	Drop Down List	N
TMG - Work Allocation Search - Work Items	Auto Complete List	N
TMG - Work Allocation Search Order	Drop Down List	N
TMG - Work Allocation Sets - Allocation Search	Auto Complete List	N
TMG - Work Allocation Statuses	Drop Down List	N
TMG - Work Allocation Wide Sort By	Drop Down List	N
TMG - Work Allocation Work Bench - Work Item	Auto Complete List	N
TMG - Work Item Sets	Auto Complete List	N
TMG - Work Item Types	Drop Down List	N
TMG - Work Items	Auto Complete List	N
TRANSFER_PROTOCOL	Drop Down List	N
Text Area	Text Area	N
Text Area - 1800	Text Area	N
Text Field - 200	Text Field	Y
Text Field - 40	Text Field	Y
Time Format	Drop Down List	Y
URL	Web Address (URL)	N
User Data Column	Drop Down List	N
User Sign Off	Drop Down List	Y
VC - Applications Existing	Auto Complete List	N
VC - Branching	Drop Down List	N
VC - Check In Source	Auto Complete List	Y
VC - Check Out Destination	Auto Complete List	Y
VC - Dev Environment	Drop Down List	N
VC - Dev Pkg Number	Auto Complete List	N

Table 0-13. Kintana System Validations

Validation Name	Component Type	Use in Workflow?
VC - Directory Chooser	Directory Chooser	N
VC - Employees	Auto Complete List	N
VC - Existing Sub Paths	Auto Complete List	Y
VC - File Chooser	File Chooser	N
VC - File Revisions	Auto Complete List	N
VC - File Type	Drop Down List	N
VC - Files Existing	Auto Complete List	Y
VC - Report Type	Drop Down List	N
Version Control Application codes	Auto Complete List	N
WF - Allowed Package Workflows	Auto Complete List	N
WF - Approval Step	Drop Down List	Y
WF - Approval Step w/ Cancel	Drop Down List	Y
WF - Approval Step w/Cancel	Drop Down List	Y
WF - Child Package Results	Drop Down List	Y
WF - Close Statuses	Drop Down List	Y
WF - Decisions - All	Auto Complete List	N
WF - Decisions Required	Drop Down List	N
WF - Default Package Workflow	Auto Complete List	N
WF - Default Request Types	Auto Complete List	N
WF - Evaluation Step	Drop Down List	Y
WF - Events	Drop Down List	N
WF - Execution Step w/ Retry and Abort	Drop Down List	Y
WF - Execution Type	Drop Down List	N
WF - Executions - All	Auto Complete List	N
WF - Jump/Receive Step Labels	Drop Down List	N
WF - Migrate Step	Drop Down List	Y
WF - Notification Intervals - Enabled	Drop Down List	N
WF - Package Status	Drop Down List	N
WF - Parent Status Code - All	Drop Down List	N
WF - Processing Type	Drop Down List	N
WF - Product Scope (with ALL)	Drop Down List	N
WF - Product Scopes	Drop Down List	N
WF - QA Test Step	Drop Down List	Y
WF - Recipient Tokens - Enabled	Auto Complete List	N
WF - Recipient Tokens w/ no Security Groups - Enabled	Auto Complete List	N
WF - Request to Child Package Reference Relationships	Drop Down List	N
WF - Request to Child Request Reference Relationships	Drop Down List	N
WF - Rework Code	Drop Down List	Y
WF - Show All Workflow Steps	Drop Down List	N
WF - Standard Condition Results	Drop Down List	Y
WF - Standard Execution Results	Drop Down List	Y

Table 0-13. Kintana System Validations

Validation Name	Component Type	Use in Workflow?
WF - Standard Execution Results w/ Reset, Abort	Drop Down List	Y
WF - Standard Execution Results w/ Reset, Rejected	Drop Down List	Y
WF - Standard Execution Results w/ Reset, Reset Failures, Abort	Drop Down List	Y
WF - Step Authentication Type	Drop Down List	N
WF - Step Security Type	Drop Down List	N
WF - Step Security Type Choices	Drop Down List	N
WF - Step Timeout Type	Drop Down List	Y
WF - Timeout Unit	Drop Down List	Y
WF - Tokens - Enabled	Auto Complete List	N
WF - Txn Operators - NonNumeric	Drop Down List	N
WF - Txn Operators - Numeric	Drop Down List	N
WF - Validations	Auto Complete List	N
WF - Workflow Command - Enabled	Drop Down List	N
WF - Workflow Errors	Drop Down List	N
WF - Workflow ID - All	Auto Complete List	N
WF - Workflow ID w/o Subworkflow	Auto Complete List	N
WF - Workflow Name - All	Auto Complete List	N
WF - Workflow Step Display Types	Drop Down List	N
WF - Workflow Steps - All	Auto Complete List	N
Web Address	Web Address (URL)	N
Yes No Radio Buttons	Radio Button (Yes/No)	N
Yes or No Drop Down list	Drop Down List	Y
Yes or No Drop Down list with ALL	Drop Down List	Y



# Appendix

# C

# Tokens

While configuring certain features in Kintana, it is often necessary to reference information that is undefined until the Kintana product is actually used a particular context. Instead of generating objects that are valid only in specific contexts, Kintana uses variables can be used to facilitate the creation of general objects that can be applied to a variety of contexts. These variables are called tokens.

There are two types of tokens found within Kintana: custom tokens and standard tokens. Standard tokens are provided with the product. Custom tokens are generated to suit specific needs. Each field of the following Kintana entities can be referenced as a custom token:

- Object Types
- Request Types and Request Header Types
- Report Types
- User Data
- Workflow Parameters

In addition, numerous standard Tokens are available that provide other useful pieces of information related to the Kintana system. For example, Kintana has a Token that represents the users currently logged onto the system.

For instructions on using Tokens and for a list of available system Tokens, see *"Using Commands and Tokens"*.



# Appendix D

## User Data Creation and Processing

Every entity in Kintana (such as Packages, Workflows, Requests and Projects) has a set of standard fields that provide information about the entity. While these fields are normally sufficient for day to day processing, it is possible to capture additional information specific to each organization. Kintana's User Data provides the ability to capture this additional information.

For every major entity in Kintana, up to 20 User Data fields can be defined. These fields are displayed in the **USER DATA** tab for the specific 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.

User Data fields are available for each entity instance generated; the fields are available globally. For example, you can configure a **MANAGER** field to appear on the **USER DATA** tab in the **USER** window. You could then specify each user's manager when setting up their Kintana account.

For some entities, context-sensitive custom fields can be set up. For example, User Data fields could be defined for the Request entities that are only available when the priority of a Request is **CRITICAL**.

The following entities support User Data functionality:

- Budgets
- Organizations
- Resource Pools
- Staffing Profiles
- Packages
- Package Lines
- Environments

- Environment Application
- Environment Refresh
- Requests
- Request Types
- Projects
- Tasks
- Security Groups
- Users
- Validation Values
- Workflows
- Workflow Steps
- Workflow Executions
- Workflow Decisions

The following topics are discussed:

- [“Creating and Editing Kintana User Data”](#) on page 308
- [“Creating and Editing Context Sensitive User Data”](#) on page 318
- [“Project/Task User Data Roll-Up”](#) on page 331
- [“Referring to User Data”](#) on page 344
- [“Migrating User Data”](#) on page 344



Note

For information on screens and fields in the User Data Workbench, refer to the [“Configuration Workbench Reference”](#) document.

## Creating and Editing Kintana User Data

The following sections provide detailed instructions for creating and editing Kintana User Data:



- [Adding User Data Fields](#)
- [Copying a Field's Definition](#)
- [Editing User Data Fields](#)
- [Configuring User Data Field Dependencies](#)
- [Removing Fields](#)
- [Modifying the User Data Layout](#)



Note

To configure User Data, you must have the **CONFIG: EDIT USER DATA** Access Grant.

## Adding User Data Fields

To generate a new User Data field:

1. Click the **CONFIGURATION** screen group and click the **USER DATA** screen. The **USER DATA WORKBENCH** opens.
2. Search for the desired User Data Type from the **QUERY** tab and select it from the **RESULTS** tab of the **USER DATA WORKBENCH**.
3. Click **OPEN**. The **USER DATA** window opens.
4. Click **NEW** in the **FIELDS** Tab. The **FIELD: NEW** window opens.

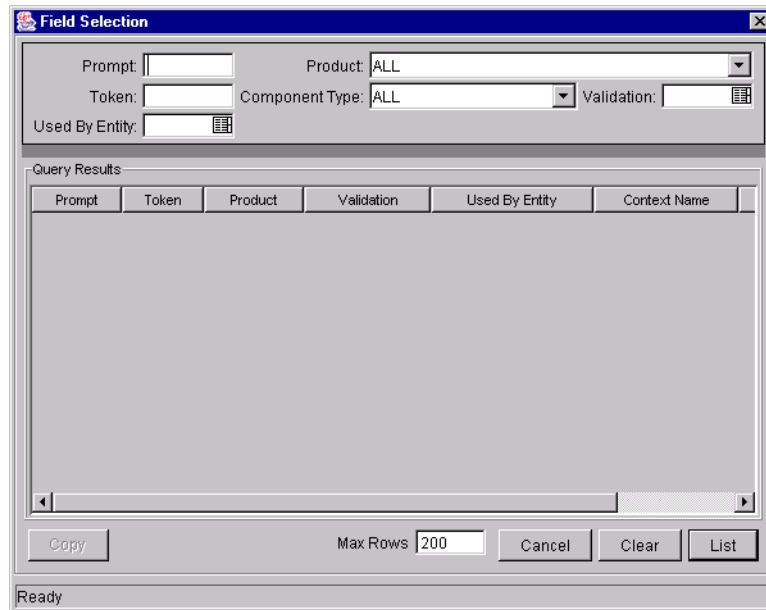
5. Enter the general information region fields for the User Data field. Fields appearing in the general information region are defined in "*Configuration Workbench Reference*".
6. Click the **ATTRIBUTES** tab to define the field's basic properties (DISPLAY, DISPLAY ONLY, REQUIRED). Fields appearing in the **ATTRIBUTES** tab are defined in "*Configuration Workbench Reference*".
7. Click the **DEFAULTS** tab to define the default value for that field. Fields appearing in the **DEFAULTS** tab are defined in "*Configuration Workbench Reference*".
8. Click the **DEPENDENCIES** tab to define the field dependent properties of the field (CLEAR WHEN, DISPLAY ONLY WHEN, REQUIRED WHEN). Fields appearing in the **DEPENDENCIES** tab are defined in "*Configuration Workbench Reference*".
9. Click **OK** to add the User Data field to the entity.

## 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. Click the **CONFIGURATION** screen group and click the **USER DATA** screen. The **USER DATA WORKBENCH** opens.
2. Search for the desired User Data Type from the **QUERY** tab and select it from the **RESULTS** tab of the **USER DATA WORKBENCH**.
3. Click **OPEN**. The **USER DATA** window opens.
4. Click **NEW** in the **FIELDS** Tab. The **FIELD: NEW** window opens.
5. Click **COPY FROM**. The **FIELD SELECTION** window opens.



6. Search for 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.



Note

Because of the large number of fields in the Kintana system, you should limit the list of fields by one or more of the query criteria.

7. Select the desired field.
8. Click **COPY**. This closes the window and copies the definition of the selected field into the **FIELD: NEW** window.
9. Make any necessary modifications and click **OK**.

## Editing User Data Fields

To edit an existing field:

1. Click the **CONFIGURATION** screen group and click the **USER DATA** screen. The **USER DATA WORKBENCH** opens.
2. Search for the desired User Data Type from the **QUERY** tab and select it from the **RESULTS** tab of the **USER DATA WORKBENCH**.

3. Click **OPEN**. The USER DATA window opens.
4. Select the field you wish to edit.
5. Either double-click on the Field in the **FIELDS** tab or select the field and click **EDIT**. This opens up a FIELD window.

6. Make the desired changes in the header region, **ATTRIBUTES** tab, **DEFAULT** tab, and **DEPENDENCIES** tab.
7. Click **APPLY** to save the changes to the **FIELDS** tab without closing the FIELD window, or click **OK** to save the changes and close the FIELD window.

The field has been updated with the changes.

## 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 0-14](#).

- Become required when another field meets a logical condition, defined in [Table 0-14](#).

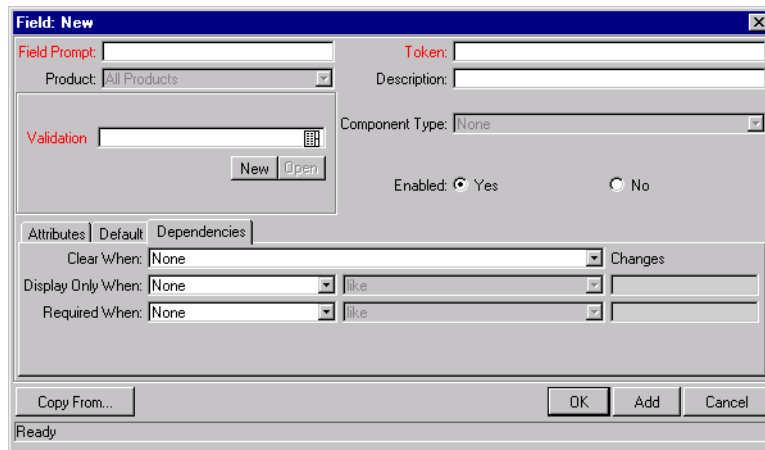
Table 0-14. Field Dependency Logical Qualifiers

Logical qualifier	Definition
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. Click the **CONFIGURATION** screen group and click the **USER DATA** screen. The **USER DATA WORKBENCH** opens.
2. Search for the desired User Data Type from the **QUERY** tab and select it from the **RESULTS** tab of the **USER DATA WORKBENCH**.
3. Click **OPEN**. The **USER DATA** window opens.
4. Select the field you wish to edit.

5. Either double-click on the Field in the **FIELDS** tab or select the field and click **EDIT**. The **FIELD** window opens.
6. Click the **DEPENDENCIES** tab.



The screenshot shows the 'Field: New' dialog box with the 'Dependencies' tab selected. The dialog is divided into several sections:

- Field Prompt:** A text input field.
- Token:** A text input field.
- Product:** A dropdown menu currently set to 'All Products'.
- Description:** A text input field.
- Validation:** A text input field with a small icon to its right.
- Component Type:** A dropdown menu currently set to 'None'.
- Enabled:** Radio buttons for 'Yes' (selected) and 'No'.
- Attributes | Default | Dependencies:** A tabbed section with 'Dependencies' selected.
- Clear When:** A dropdown menu currently set to 'None'.
- Changes:** A dropdown menu.
- Display Only When:** A dropdown menu currently set to 'None', followed by a logical qualifier dropdown (set to 'like') and a field name dropdown.
- Required When:** A dropdown menu currently set to 'None', followed by a logical qualifier dropdown (set to 'like') and a field name dropdown.
- Buttons:** 'Copy From...', 'OK', 'Add', and 'Cancel'.
- Status:** 'Ready'.

7. 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 (i.e. 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.
8. Click **OK**.

## Removing Fields

To remove a field permanently from a User Data Type:

1. Click the **CONFIGURATION** screen group and click the **USER DATA** screen. The USER DATA WORKBENCH opens.
2. Search for the desired User Data Type from the **QUERY** tab and select it from the **RESULTS** tab of the USER DATA WORKBENCH.
3. Click **OPEN**. The USER DATA window opens.
4. Select the field in the **FIELDS** tab.
5. Click **REMOVE**.
6. Click **OK** to save the change to the database and close the window.

## Modifying the User Data Layout

The layout of User Data fields can be changed in the **LAYOUT** tab of the USER DATA window.

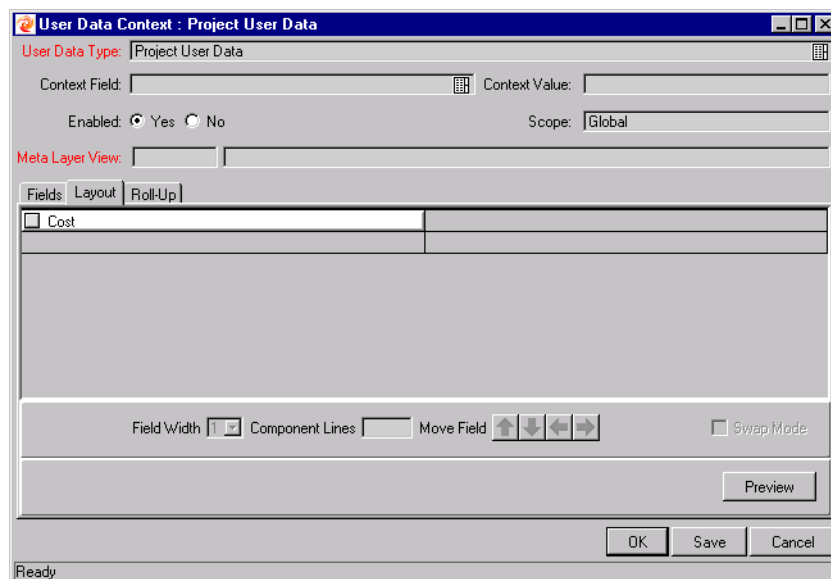


Figure 0-10 User Data Window - Layout Tab

The following sections discuss modifying User Data field layout in more detail:

- [Changing Column Width](#)
- [Moving a Field](#)

- *Swapping Positions of Two Fields*

### Changing Column Width

To change the column width of a Field:

1. Click the **CONFIGURATION** screen group and click the **USER DATA** screen. The **USER DATA WORKBENCH** opens.
2. Search for the desired User Data Type from the **QUERY** tab and select it from the **RESULTS** tab of the **USER DATA WORKBENCH**.
3. Click **OPEN**. The **USER DATA** window opens.
4. Click the **LAYOUT** tab.
5. Select the Field.
6. Select either **1** or **2** in the **FIELD WIDTH** radio button.



Note

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.

### Moving a Field

To move a field or a set of fields:

1. Click the **CONFIGURATION** screen group and click the **USER DATA** screen. The **USER DATA WORKBENCH** opens.
2. Search for the desired User Data Type from the **QUERY** tab and select it from the **RESULTS** tab of the **USER DATA WORKBENCH**.
3. Click **OPEN**. The **USER DATA** window opens.
4. Click the **LAYOUT** tab.



5. Select the field(s). 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.
6. Use the directional arrow buttons to move the fields to the desired location in the layout builder.



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.

### *Swapping Positions of Two Fields*

To swap the positions of two fields:

1. Click the **CONFIGURATION** screen group and click the **USER DATA** screen. The USER DATA WORKBENCH opens.
2. Search for the desired User Data Type from the **QUERY** tab and select it from the **RESULTS** tab of the USER DATA WORKBENCH.
3. Click **OPEN**. The USER DATA window opens.
4. Click the **LAYOUT** tab.
5. Select the first field.
6. Select the SWAP MODE check box. This causes an **S** to appear in the check box area of the selected field.
7. Once the **S** appears, double-click on the field to be swapped with. This causes the two fields to change positions.
8. Following the swap, the Swap Mode is turned off.

The fields have now been swapped. To swap another set of fields, repeat the above procedure.

### *Previewing the Layout*

To check what the layout will look like in actual use, click **PREVIEW**. This opens a small window that shows the fields as they will appear in the window, shown in [Figure 0-11](#).

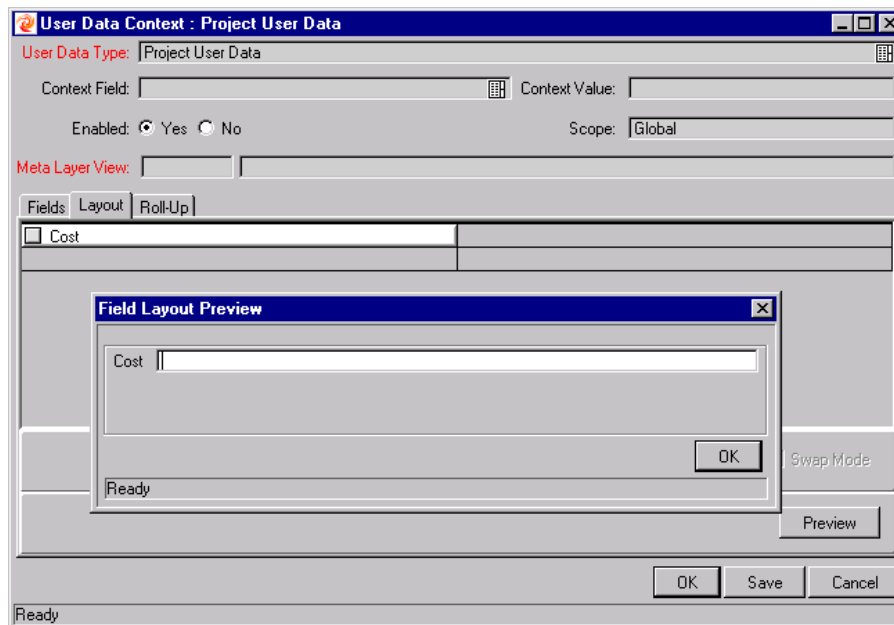


Figure 0-11 Layout Tab in Preview Mode

User Data fields are also visible in the Kintana interface.



Note

If all the 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.

## Creating and Editing Context Sensitive User Data

The following sections provide detailed instructions for creating and editing Context Sensitive User Data:

- [Creating Context Sensitive User Data](#)
- [Editing Context Sensitive User Data](#)
- [Deleting Context Sensitive User Data](#)
- [Copying Context Sensitive User Data](#)

- [Example - Using Context Sensitive User Data for a Field in a Request Header Type](#)

## Creating Context Sensitive User Data

Context Sensitive User Data can be defined for the Request, Package, and Validations (Validation value region) windows in the USER DATA WORKBENCH. To define Context Sensitive User Data:

1. Define a Context Field in the Global User Data scope. See [“Defining the Context Field”](#) on page 319.
2. Define a Context Value. See [“Defining a Context Value”](#) on page 320.
3. Define and configure the fields which appear under certain contexts. See [“Defining the Context Sensitive Fields”](#) on page 321.



Note

When defining or editing Context Sensitive User Data fields for the same User Data Type, make sure to save any changes to the Global User Data fields before working on the Context User Data fields.

### *Defining the Context Field*

Only one field can be defined as the Context Field at any given time. To specify the Context Field:

1. Click the **CONFIGURATION** screen group and click the **USER DATA** screen. The USER DATA WORKBENCH opens.
2. Click **LIST** to display all of the existing User Data Types.

The screenshot shows the 'User Data Workbench' window with a table of records. The table has the following columns: User Data Type, Scope, Context Field, Context Value, and Enabled. The records are as follows:

User Data Type	Scope	Context Field	Context Value	Enabled
Request Type User Data	Global			Y
Request User Data	Global	Application		Y
Request User Data	Context	Application	CSM App	Y
Request User Data	Context	Application	ERP Application	Y
Request User Data	Context	Application	HR Application	Y
Request User Data	Context	Application	HR Application	Y
Security Group User Data	Global			Y
Task User Data	Global			Y
User User Data	Global			Y
Validation Value User Data	Global	Validation Name		Y
Validation Value User Data	Context	Validation Name	CONNECTION_PR...	Y
Validation Value User Data	Context	Validation Name	TRANSFER_PROT...	Y

At the bottom of the window, there are buttons for 'New', 'Open', 'Copy', 'Delete', and 'Refresh'. A status bar at the bottom indicates '27 User Data Context Records are loaded.'

3. Select the desired User Data Type (Package, Request or Validation) with a **GLOBAL** scope.
4. Click **OPEN**. The USER DATA CONTEXT window opens.
5. Select the CONTEXT FIELD from the auto-complete list.

Note

The CONTEXT FIELD is disabled if any specific contexts for the User Data has been defined. In order to change the CONTEXT FIELD, all specific contexts for the User Data Type must first be deleted. For more information on editing Context Sensitive User Data, see *“Editing Context Sensitive Fields”* on page 323.

6. Verify that the global context is enabled (ENABLED=YES).
7. Click **OK** to save and close the window.

The CONTEXT FIELD has now been specified.

Note

The CONTEXT FIELD for the Validations Value User Data Type is always **VALIDATION NAME**.

### Defining a Context Value

Context Values are the predefined possible values for the selected CONTEXT FIELD. Different User Data fields can be defined to be displayed for each of the possible Context Values. To define a Context Value:

1. Click the **CONFIGURATION** screen group and click the **USER DATA** screen. The USER DATA WORKBENCH opens.

2. Click **NEW** in the **RESULTS** tab or **NEW USER DATA CONTEXT** in the **QUERY** tab. The **USER DATA CONTEXT** window opens.
3. Select a **USER DATA TYPE** from the auto-complete list. The **CONTEXT FIELD** is displayed as read-only.



Note

Only User Data Types with a defined **CONTEXT FIELD** appear in the list. To define a **CONTEXT FIELD** for Request, Package, or Validation Values, see [“Defining the Context Field”](#) on page 319.

4. Select a **CONTEXT VALUE** from the drop down list or auto-complete list.

### Defining the Context Sensitive Fields

User Data fields to be used with the specified **CONTEXT VALUE** are defined and configured just as in other areas of the Kintana Product Suite. For details on defining the field content and layout, see one of the following sections:

- [“Adding User Data Fields”](#) on page 309
- [“Editing User Data Fields”](#) on page 311
- [“Removing Fields”](#) on page 314

To define different fields based on a different **CONTEXT VALUE**, see [“Defining a Context Value”](#) on page 320.

### Editing Context Sensitive User Data

Context Sensitive User Data can be edited for the Request, Package, Environment, and Validations (Validation value region) windows in the **USER DATA WORKBENCH**. For details on editing Context Sensitive User Data, see one of the following sections:

- [Changing the Context Field](#)
- [Changing the Context Value](#)
- [Editing Context Sensitive Fields](#)

### Changing the Context Field

In order to change the **CONTEXT FIELD**, all specific contexts for the User Data Type must first be deleted. To change the **CONTEXT FIELD** for a particular User Data Type:

1. Click the **CONFIGURATION** screen group and click the **USER DATA** screen. The **USER DATA WORKBENCH** opens.
2. Click **LIST** to display all of the existing User Data Types.
3. Locate the desired User Data Type.
4. Select the rows where the desired User Data Type's **SCOPE=CONTEXT**.
5. Click **DELETE**.
6. Select the desired User Data Type. It will have a **GLOBAL** Scope.
7. Click **OPEN**. The **USER DATA CONTEXT** window opens.
8. Select the **CONTEXT FIELD** from the auto-complete list.
9. Verify that the Global Context is enabled (**ENABLED=YES**).
10. Click **OK** to save and close the window.

The User Data Type's **CONTEXT FIELD** has now been changed.



Note

The **CONTEXT FIELD** for the **Validations Value** User Data Type is always **VALIDATION NAME** and cannot be changed.

### *Changing the Context Value*

To change an existing User Data Type's **CONTEXT VALUE**:

1. Click the **CONFIGURATION** screen group and click the **USER DATA** screen. The **USER DATA WORKBENCH** opens.
2. Click **LIST** to display all of the existing User Data Types.
3. Select the desired User Data Type that has the **CONTEXT VALUE** to be changed.
4. Click **OPEN**. The **USER DATA CONTEXT** window opens.
5. Select a new **CONTEXT VALUE** from the drop down list or auto-complete list.
6. Click **OK** to save the changes and close the window.

The User Data's **CONTEXT VALUE** has been changed.

## Editing Context Sensitive Fields

User Data fields to be used with the specified CONTEXT VALUE are edited just as in other areas of the Kintana Product Suite. For details on editing the field content and layout, see one of the following sections:

- [“Adding User Data Fields”](#) on page 309
- [“Editing User Data Fields”](#) on page 311
- [“Removing Fields”](#) on page 314

## Deleting Context Sensitive User Data

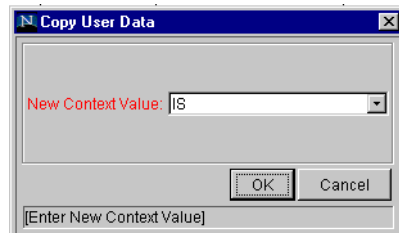
To delete a Context Sensitive User Data Type:

1. Click the **CONFIGURATION** screen group and click the **USER DATA** screen. The **USER DATA WORKBENCH** opens.
2. Click **LIST** to display all of the existing User Data Types.
3. Select the User Data Type to be deleted.
4. Click **DELETE**. A **QUESTION** dialog opens with the message “Delete 1 User Data Context[s]?”
5. Click **YES** to confirm deletion.

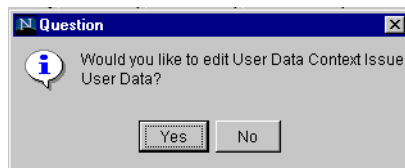
## Copying Context Sensitive User Data

To copy a Context Sensitive User Data Type:

1. Click the **CONFIGURATION** screen group and click the **USER DATA** screen. The **USER DATA WORKBENCH** opens.
2. Click **LIST** to display all of the existing User Data Types.
3. Select the User Data Type with **SCOPE=CONTEXT** that is to be copied.
4. Click **COPY**. The **COPY USER DATA** window opens.



5. Select a New CONTEXT VALUE and click **OK**. A QUESTION dialog opens.



6. Click **YES** to edit the context sensitive fields or **No** to accept.

### Example - Using Context Sensitive User Data for a Field in a Request Header Type

Different values can appear in the Request's APPLICATIONS field depending on which Request Header Type is used. For the APPLICATIONS field in an ERP Request, the following distinct set of fields are available:

- Accounts Receivable
- Accounts Payable
- General Ledger
- Inventory

For an eCommerce Request, the following fields are available:

- Registration
- User Preferences
- Order Entry
- Order Tracking





Note

Changing the Validation of a field in a Request Header Type can affect how information is returned from queries and reports. Kintana, therefore, recommends a context sensitive User Data approach when setting up such a system.

See "[Create Workbench Reference](#)" for more information on Request Header Types.

The following procedure provides an example for setting up the APPLICATIONS Validation for the ERP Request introduced above:

### *Setting Up the Context Sensitive User Data*

First, the Context Sensitive User Data must be configured.

1. Click the **CONFIGURATION** screen group and click the **USER DATA** screen. The USER DATA WORKBENCH opens.
2. Click **NEW USER DATA CONTEXT**. The USER DATA CONTEXT window opens.
3. Select **VALIDATION VALUE USER DATA** from the USER DATA TYPE auto-complete list.
4. Select **KNTA - APPLICATION - ENABLED** from the CONTEXT VALUE auto-complete list.
5. Click **NEW**. The FIELD: NEW window opens.
6. Create a new field with the following specifications:
  - FIELD PROMPT = **USED BY REQUEST HEADER TYPE**
  - TOKEN = **REQUEST\_HEADER\_TYPE\_ID**
  - VALIDATION = **CRT - REQUEST HEADER TYPES - ALL**

**Field: Used By Request Header Type**

Field Prompt: Used By Request Header Type      Token: REQUEST\_HEADER\_TYPE\_ID

Product: All Products      Description:

Validation: >RT - Request Header Typ      Component Type: Auto Complete List

Enabled:  Yes       No

Attributes | Default | Dependencies

User Data Col.: USER\_DATA1      Display Only: Never

Display:  Yes       No      Required: Always

OK    Apply    Cancel

Ready

Signed by: Kintana, Inc.

7. Click **OK**.

### Example: Configuring the Validations

The Validation can now be configured:

1. Click the **CONFIGURATION** screen group and click the **VALIDATIONS** screen. The **VALIDATION WORKBENCH** opens.
2. Search for and open the global 'KNTA - Application - Enabled' Validation. The **VALIDATION** window opens.

**Validation : KNTA - Application - Enabled**

Name: KNTA - Application - Enabled

Description: KNTA - Application - Enabled

Enabled:       Use in Workflow?

Component Type: Auto Complete List

Validated By: List

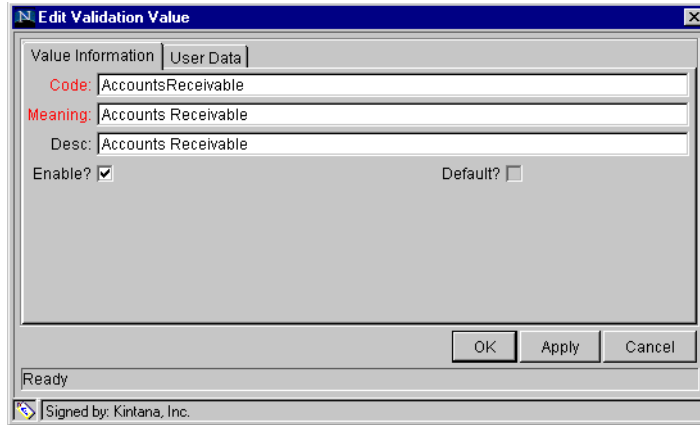
Seq	Code	Meaning	Description	Enabled	Default
1	Apps*Integrity_1.5	Apps*Integrity 1.5	Apps*Integrity 1.5	N	N
2	Apps*Integrity_1.6	Apps*Integrity 1.6	Apps*Integrity 1.6	N	N
3	Apps*Integrity_1.7	Apps*Integrity 1.7	Apps*Integrity 1.7	N	N
4	Apps*Integrity_2.0	Kintana Deliver	Kintana Deliver	Y	N
5	GL*Migrator_1.0	GL*Migrator	GL*Migrator	Y	N
6	Calendar	Calendar		N	N
7	Env*Integrity_1.0	Env*Integrity 1.0	Env*Integrity 1.0	N	N
8	Accelerators	Accelerators	Accelerators	Y	N
9	Support*Express	Support*Express		N	N

New    Edit    Delete    Copy From    ↑ ↓

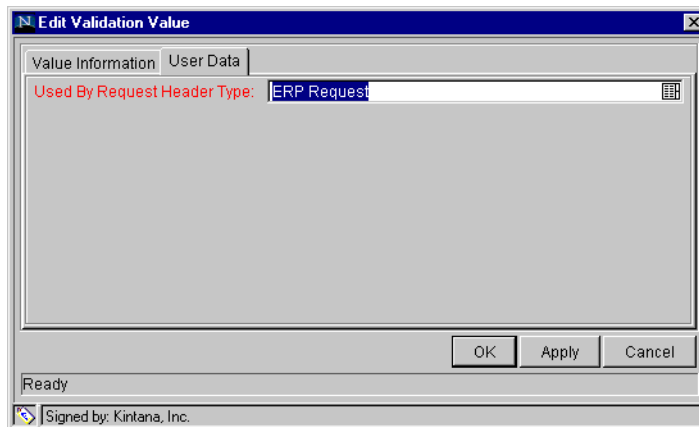
Used By    Ownership      OK    Save    Cancel

Ready (Read-Only, Seed Data)

3. Associate each Validation Value with the appropriate Request Header Type shown in *Table 0-15*. To associate a Validation Value with a Request Header Type:
  - a. Select a Validation Value from the VALIDATION VALUES list.
  - b. Click **EDIT**. The EDIT VALIDATION VALUE window opens.



- c. Click the **USER DATA** tab.
  - d. Select the Request Header Type (**ERP REQUEST** in this example) from the USED BY REQUEST HEADER TYPE auto-complete list.



- e. Click **OK**.
  - f. Repeat these steps for each row in *Table 0-15*.

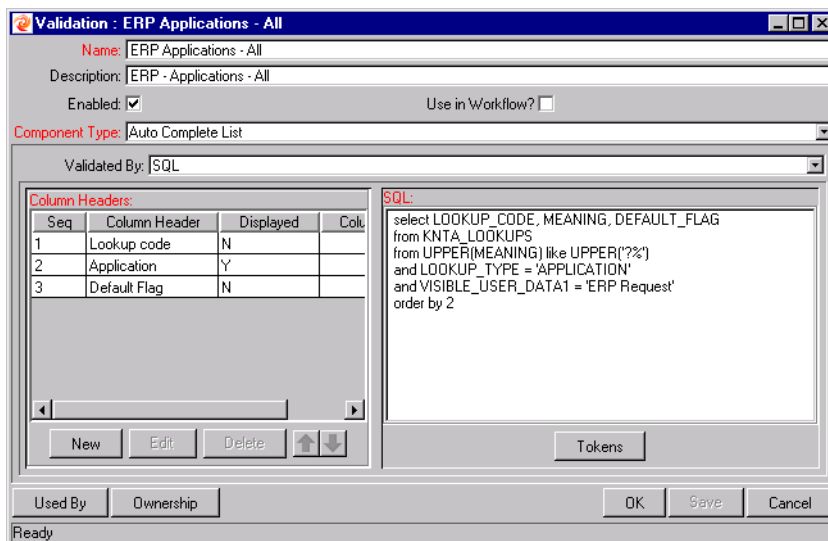
*Table 0-15. Validation values associated with Request Header Types*

<b>Validation Value</b>	<b>Used by Request Header Type</b>
Accounts Receivable	ERP Request
Accounts Payable	ERP Request
General Ledger	ERP Request
Inventory	ERP Request
Registration	eCommerce Request
User Preferences	eCommerce Request
Order Entry	eCommerce Request
Order Tracking	eCommerce Request

### *Example: Modifying the SQL*

You can now create variants of the standard Application Validation which vary depending on which Request Header Type is being used.

1. From the VALIDATIONS WORKBENCH, copy the 'KNTA - Application - All' Validation.
2. Rename the Validation to 'ERP Applications - All.'
3. Edit the Validation's SQL as shown below.



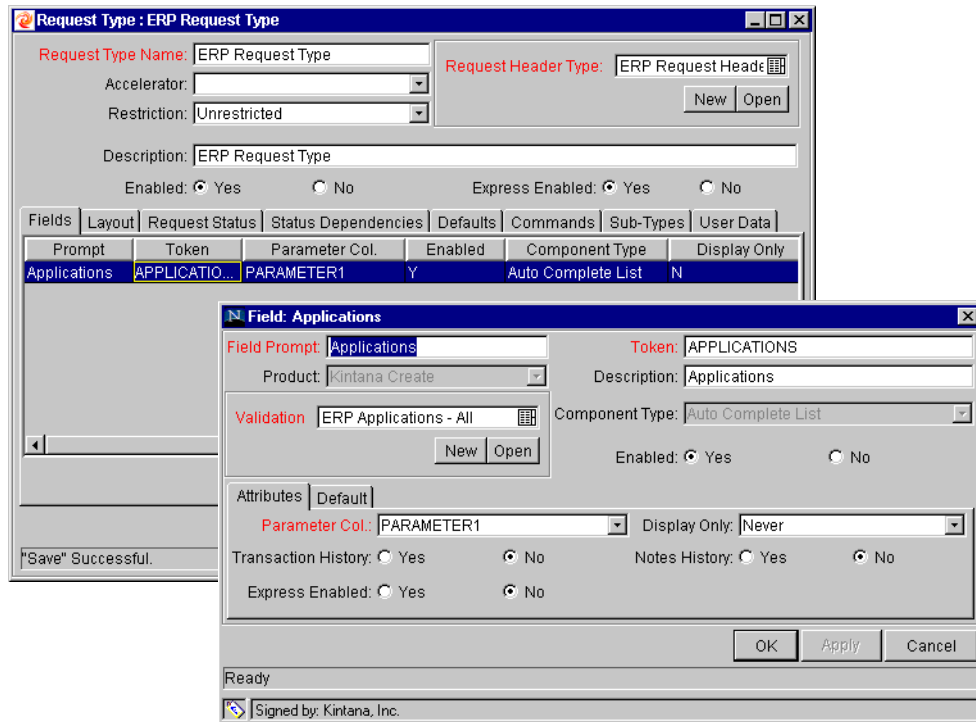
The specific variant for the ERP Request Header Type would be (assuming that the context-specific user data field was captured in the USER\_DATA1 column):

```
select LOOKUP_CODE, MEANING, DEFAULT_FLAG
from KNTA_LOOKUPS
where UPPER(MEANING) like UPPER('??')
and LOOKUP_TYPE = 'APPLICATION'
and VISIBLE_USER_DATA1 = 'ERP Request'
order by 2
```

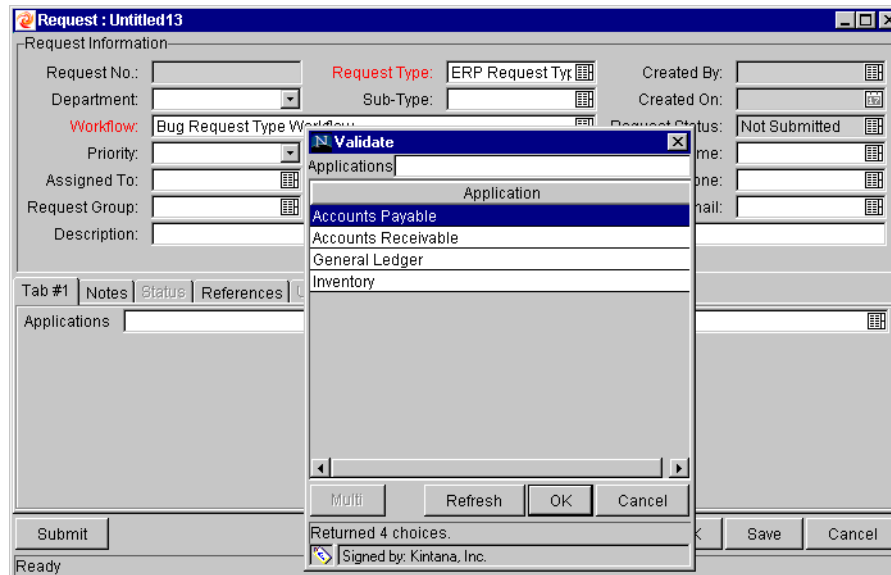
### Example: Resulting Behavior

You can now create a context-sensitive APPLICATIONS field for any Request Type. In this example, simply create a new field with the Validation 'ERP - Applications - All.'

## Configuring a Request Resolution System



When a user creates a Request with this Request Type (which references the ERP Request Header Type), the following Validations are associated with the **APPLICATIONS** field.



---

## Project/Task User Data Roll-Up

Values from Task User Data fields can be configured to “roll up” (combine/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/Task User Data Roll-Up can be used to capture various important aspects of a Project.



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.

## Creating Project/Task User Data Roll-Up

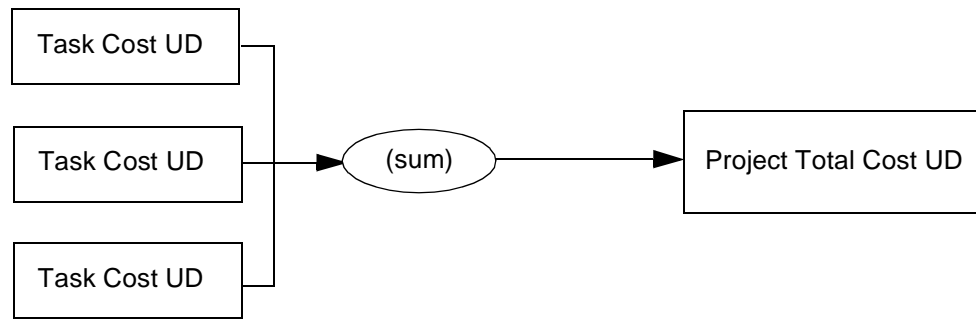
User Data must be configured for Projects and Tasks before specifying Roll-Up Methods.

1. Create and configure Project and Task User Data fields.
2. Link Project and Task User Data fields with Roll-Up Method.

For more detailed information on configuring User Data, see [“Creating and Editing Kintana User Data”](#) on page 308.

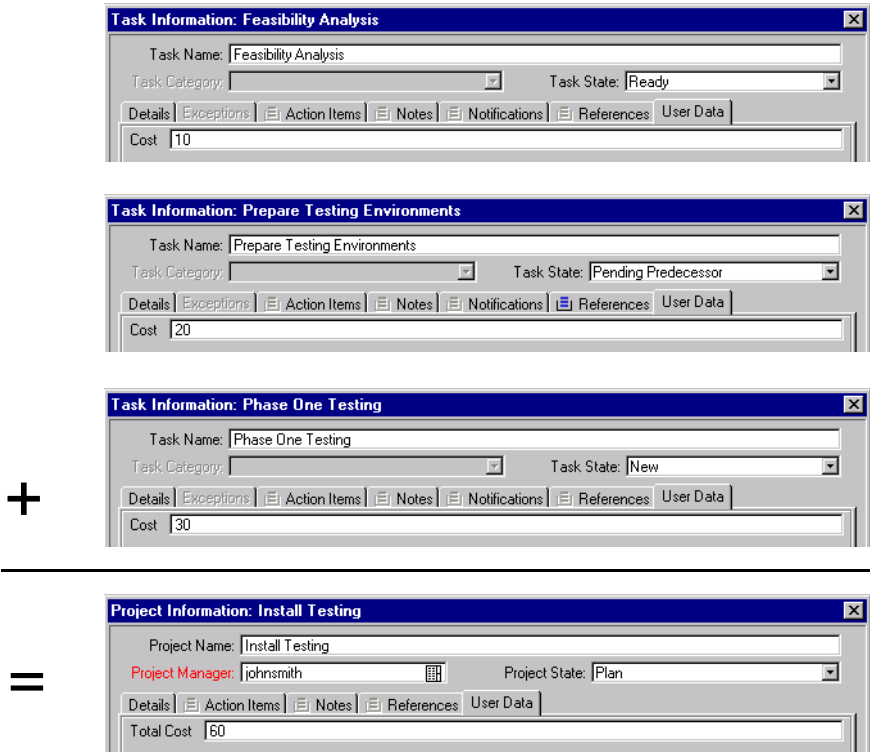
### *Example: Using Project/Task User Data Roll-Up*

Company X would like to capture total cost for its Projects. 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 below:



Each Task has its own Cost User Data field. 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.

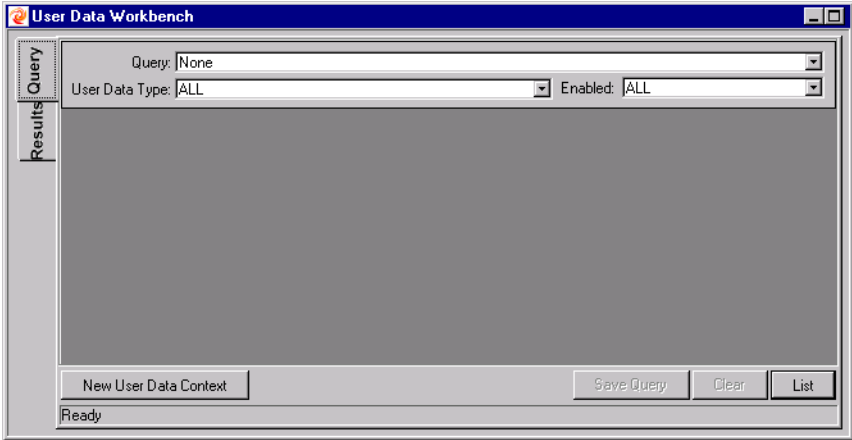




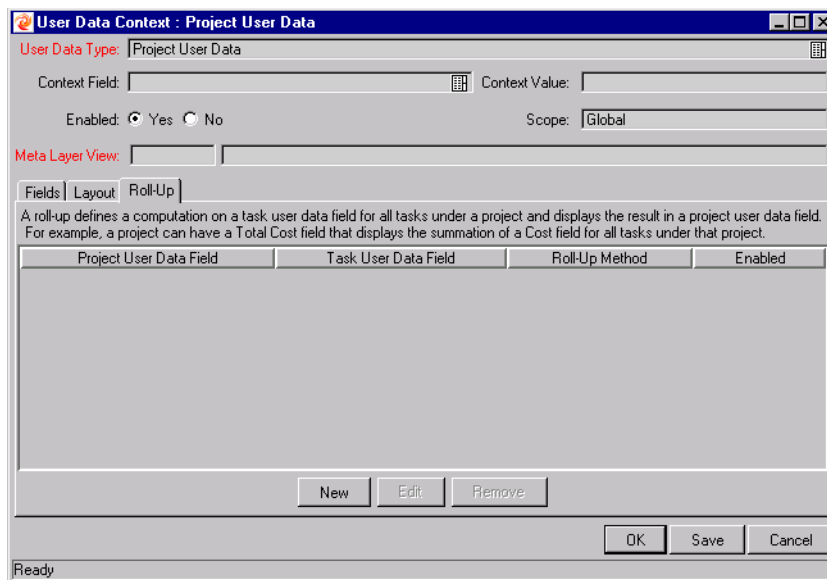
Once Project and Task User Data fields have been configured and saved, the Roll-Up relationship can be specified.

To specify the Roll-Up Method:

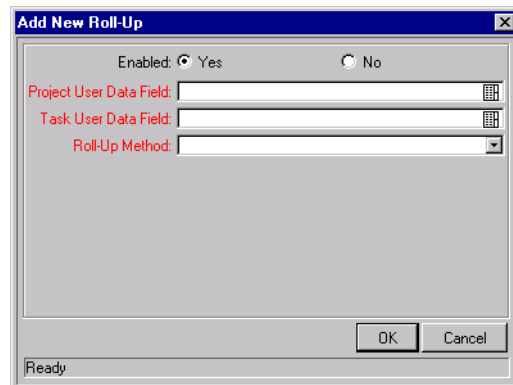
1. Click the **CONFIGURATION** screen group and click the **USER DATA** screen. The **USER DATA WORKBENCH** opens.



2. Select **PROJECT USER DATA** from the **USER DATA TYPE** drop down list and click **LIST**. The **RESULTS** tab opens with the Project User Data Type loaded.
3. Select the Project User Data and click **OPEN**.
4. Click the **ROLL-UP** tab.



5. Click **NEW**. The **ADD NEW ROLL-UP** window opens.



6. Select the **PROJECT USER DATA FIELD** that will contain rolled-up Task User Data values.
7. Select the **TASK USER DATA FIELD** whose values will roll up into the chosen **PROJECT USER DATA FIELD**.
8. Select the **ROLL-UP METHOD** from the drop down list.

The drop down list will only display valid options for the data types of the Project and Task User Data fields.

9. Select **YES** to enable the Roll-Up.
10. Click **OK**.

The Roll-Up relationship is added to the **ROLL-UP** tab.

11. Click **SAVE**.



Note

Only two User Data fields of the same type can be selected for Roll-Up (for example, a Numeric field cannot roll up into a Date field, nor vice versa).

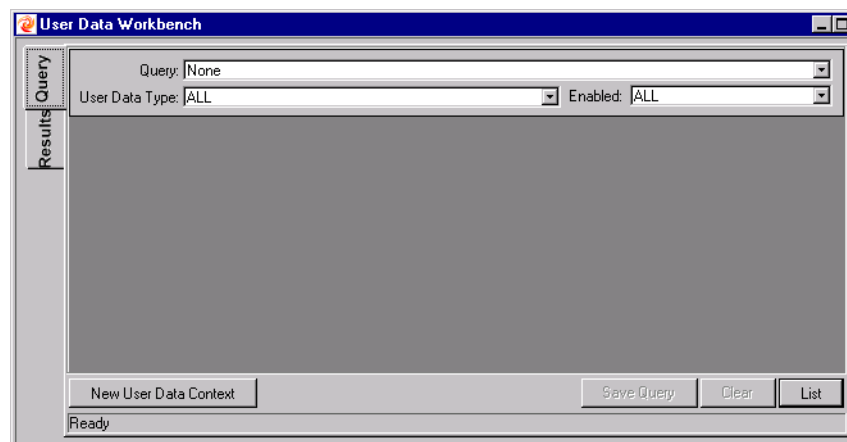
While a Task User Data field can have multiple Roll-Up relationships associated with it, a Project User Data field cannot have more than one Roll-Up relationship defined.

## Editing Project/Task User Data Roll-Up

Project/Task User Data Roll-Up can be edited from the Kintana Workbench once it has been created.

To edit a Project/Task User Data Roll-Up relationship:

1. Click the **CONFIGURATION** screen group and click the **USER DATA** screen. The **USER DATA WORKBENCH** opens.



2. Select **PROJECT USER DATA** from the **USER DATA TYPE** drop down list and click **LIST**. The **RESULTS** tab opens with the Project User Data Type loaded.
3. Select the Project User Data and click **OPEN**.

4. Click the **ROLL-UP** tab.

**User Data Context : Project User Data**

User Data Type: Project User Data

Context Field: Context Value:

Enabled:  Yes  No Scope: Global

Meta Layer View:

Fields | Layout | **Roll-Up**

A roll-up defines a computation on a task user data field for all tasks under a project and displays the result in a project user data field. For example, a project can have a Total Cost field that displays the summation of a Cost field for all tasks under that project.

Project User Data Field	Task User Data Field	Roll-Up Method	Enabled
Total Cost	Cost	Sum	Y

New Edit Remove

OK Save Cancel

Ready

5. Select the Roll-Up relationship you wish to edit.
6. Click **EDIT**. The **EDIT ROLL-UP** window opens.

**Add New Roll-Up**

Enabled:  Yes  No

Project User Data Field: Project Cost

Task User Data Field: Cost

Roll-Up Method: Sum

OK Cancel

Ready

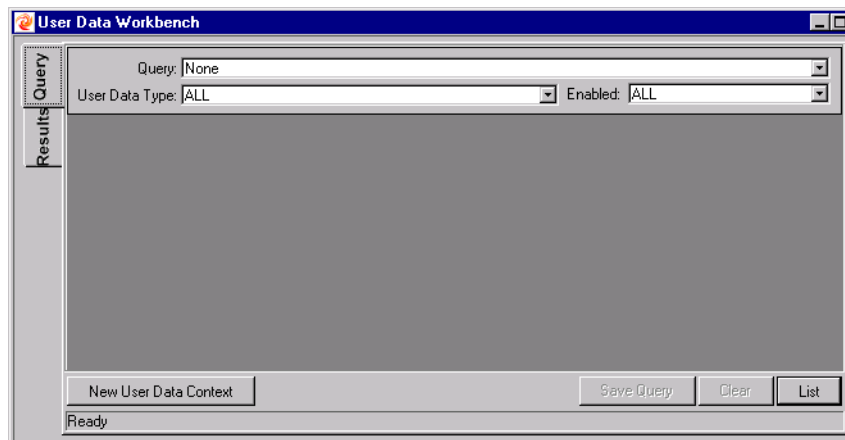
7. Make any desired changes to the Project/Task User Data field or Roll-Up Method.
  8. Click **OK**.
- The Roll-Up definition is updated in the **ROLL-UP** tab.
9. Click **SAVE**.

## Deleting Project/Task User Data Roll-Up

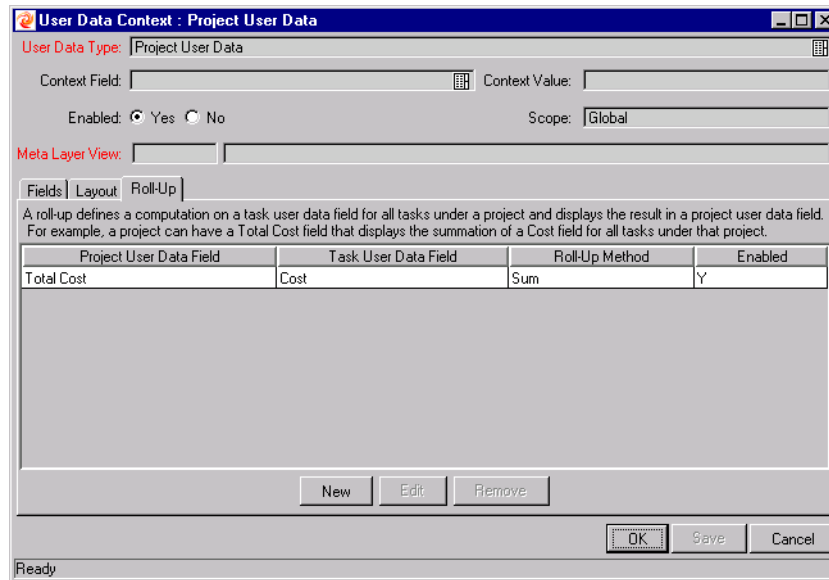
Project/Task User Data Roll-Up can be deleted. This deletion only removes the Roll-Up relationship; it does not delete the referenced User Data fields.

To delete a Project/Task User Data Roll-Up relationship:

1. Click the **CONFIGURATION** screen group and click the **USER DATA** screen. The **USER DATA WORKBENCH** opens.



2. Select **PROJECT USER DATA** from the **USER DATA TYPE** drop down list and click **LIST**. The **RESULTS** tab opens with the Project User Data Type loaded.
3. Select the Project User Data and click **OPEN**.
4. Click the **ROLL-UP** tab.



**User Data Context : Project User Data**

User Data Type: Project User Data

Context Field:  Context Value:

Enabled:  Yes  No Scope: Global

Meta Layer View:

Fields | Layout | Roll-Up

A roll-up defines a computation on a task user data field for all tasks under a project and displays the result in a project user data field. For example, a project can have a Total Cost field that displays the summation of a Cost field for all tasks under that project.

Project User Data Field	Task User Data Field	Roll-Up Method	Enabled
Total Cost	Cost	Sum	Y

New Edit Remove

OK Save Cancel

Ready

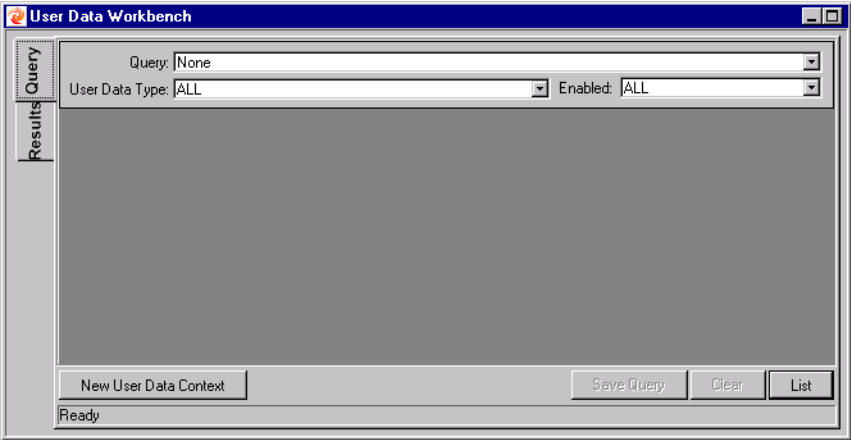
5. Select the Roll-Up relationship you wish to remove.
6. Click **REMOVE**.
7. Click **SAVE**.

### Example: Creating and Using Project/Task User Data Roll-Up

Company X wants to capture the total cost of any Project. This value will be calculated as the sum of all Task costs. They also want the calculated cost to be updated every time a Task cost is changed. They will accomplish this using Project/Task User Data Roll-Up.

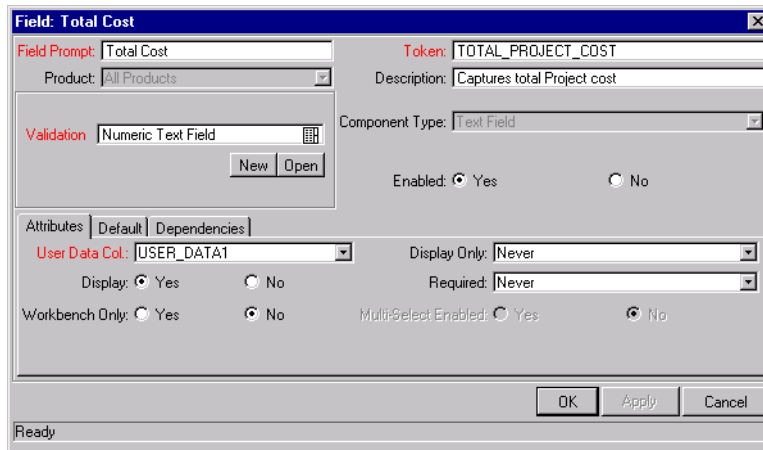
To create Project/Task User Data Roll-Up that will calculate total Project cost:

1. Click the **CONFIGURATION** screen group and click the **USER DATA** screen. The **USER DATA WORKBENCH** opens.



2. Create the Project User Data field.
  - a. Select **PROJECT USER DATA** from the USER DATA TYPE drop down list.
  - b. Click **LIST**. The **RESULTS** tab opens with the Project User Data Type loaded.
  - c. Open the Project User Data Type.
  - d. Click **NEW** in the **FIELDS** tab. The **FIELD: NEW** window opens.
  - e. Fill in the following information:

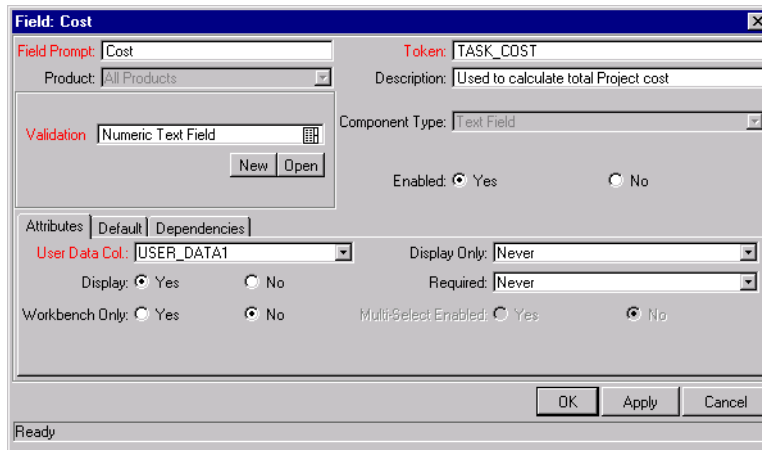
Field	Value
FIELD PROMPT	Total Cost
TOKEN	(any useful token)
DESCRIPTION	(any useful description)
VALIDATION	Numeric Text Field
ENABLED	Yes
USER DATA COL	(any available User Data column)
DISPLAY ONLY	Never
DISPLAY	Yes
REQUIRED	Never
WORKBENCH ONLY	No



- f. Click **OK** in the FIELD: NEW window. Click **OK** in the PROJECT USER DATA window to save the new field.
3. Create the Task User Data field.
    - a. In the USER DATA WORKBENCH **QUERY** tab, select **TASK USER DATA** from the USER DATA TYPE drop down list.
    - b. Click **LIST**. The **RESULTS** tab opens with the Task User Data Type loaded.
    - c. Open the Task User Data Type.
    - d. Click **NEW** in the **FIELDS** tab. The FIELD: NEW window opens.
    - e. Fill in the following information:

Field	Value
FIELD PROMPT	Cost
TOKEN	(any useful token)
DESCRIPTION	(any useful description)
VALIDATION	Numeric Text Field
ENABLED	Yes
USER DATA COL	(any available User Data column)
DISPLAY ONLY	Never
DISPLAY	Yes
REQUIRED	Never
WORKBENCH ONLY	No





- f. Click **OK** in the FIELD: NEW window. Click **OK** in the TASK USER DATA window to save the new field.
4. Create the Roll-Up relationship between the Task and Project User Data fields.
    - a. In the USER DATA WORKBENCH **QUERY** tab, select **PROJECT USER DATA** from the USER DATA TYPE drop down list.
    - b. Click **LIST**. The **RESULTS** tab opens with the Project User Data Type loaded.
    - c. Open the Project User Data Type.
    - d. Click the **ROLL-UP** tab.

**User Data Context : Project User Data**

User Data Type: Project User Data

Context Field: Context Value:

Enabled:  Yes  No Scope: Global

Meta Layer View:

Fields | Layout | Roll-Up

A roll-up defines a computation on a task user data field for all tasks under a project and displays the result in a project user data field. For example, a project can have a Total Cost field that displays the summation of a Cost field for all tasks under that project.

Project User Data Field	Task User Data Field	Roll-Up Method	Enabled
-------------------------	----------------------	----------------	---------

New Edit Remove

OK Save Cancel

Ready

e. Click **NEW**. The ADD NEW ROLL-UP window opens.

**Add New Roll-Up**

Enabled:  Yes  No

Project User Data Field:

Task User Data Field:

Roll-Up Method: SUM

OK Cancel

Ready

f. Select **TOTAL COST** for the PROJECT USER DATA FIELD.

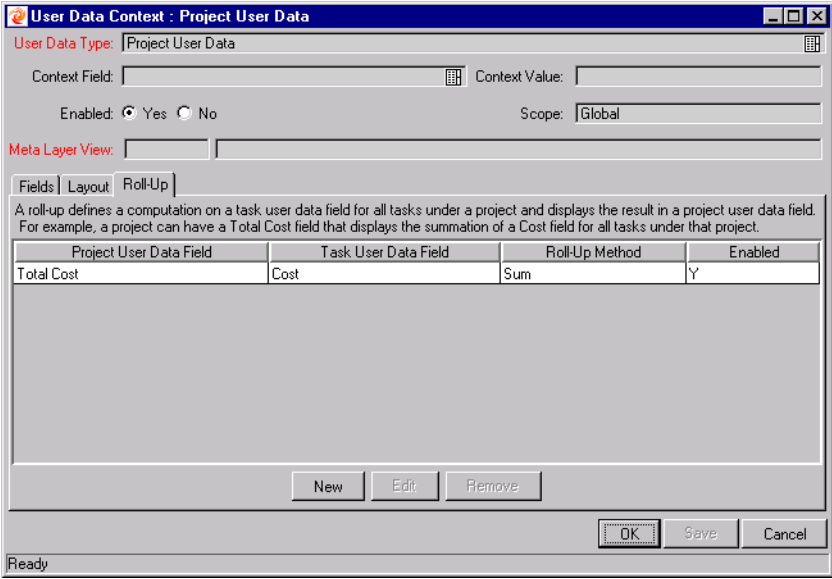
g. Select **COST** for the TASK USER DATA FIELD.

h. Select **SUM** from the ROLL-UP METHOD drop down list.

i. Click **OK**.

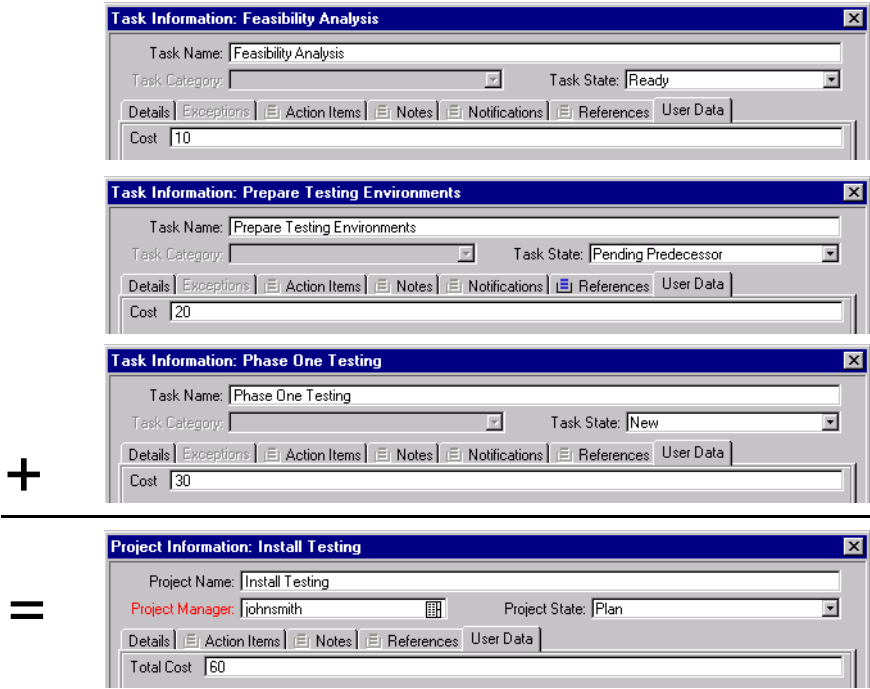
The Roll-Up relationship is added to the **ROLL-UP** tab.

5. Click **SAVE**.



The rolled-up fields can now be accessed from the **USER DATA** tab of the respective Project or Task.

Each Task has its own Cost User Data field. 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.



## Referring to User Data

Once a User Data field has been created, it is possible to refer to it from other parts of the Kintana Product Suite by its Token name, preceded by the entity abbreviation and the 'UD' qualifier.



It is possible to define a custom field for the Users entity to store the Department each user is in. This custom field would be defined using the user's User Data and would generate a field with a token of 'DEPARTMENT.' Then, in an Object Type command, a Workflow Step, or in a Report, refer to this new field as the Token [USR.UD.DEPARTMENT]. The Kintana Product Suite would then look into the custom field for the value of this Token. For more information on Tokens and their use, see the Tokens chapter in "[Using Commands and Tokens](#)".

## Migrating User Data

Kintana configuration data such as Workflows, Validations, and Request Types can be migrated between instances (installations) of Kintana. User Data values and configurations can also be migrated between instances. The following sections contain more detailed information:

- [Migrating User Data Values](#)
- [Migrating User Data Contexts](#)

### Migrating User Data Values

For any particular Kintana configuration entity with User Data fields (Request Type, Object Type, Workflow, etc.) the data in the User Data fields is migrated along with the entity.

- If the two instances have identical User Data configurations, then the User Data will be migrated correctly.
- If the 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.



Note

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 (example: 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.

## Migrating User Data Contexts

User Data field contexts can also be migrated between Kintana instances using the Kintana User Data Context Migrator Object Type. This Migrator Object Type can migrate Global as well as Context Sensitive User Data Contexts.

The screenshot shows a dialog box titled "Add Line" with a close button (X) in the top right corner. The dialog is divided into several sections:

- Object Type Information:** Contains a text field for "Object Type" with the value "Kintana User Data Context Migrator", a "Sequence" field with the value "1", and an "Application Code" dropdown menu with the value "None".
- Parameters:** A tabbed section with "User Data" selected. It contains:
  - "Migrator action:" dropdown menu with "Migrate (extract and import)" selected.
  - "Preview import?" radio buttons: "Yes" is unselected, "No" is selected.
  - "Kintana source password:" text field with a clear button (C).
  - "Kintana dest password:" text field with a clear button (C).
  - "User data context:" text field with a list icon.
  - "Content bundle directory:" text field with a folder icon.
  - "Content bundle filename:" text field with a file icon.
  - Three sets of radio buttons for "Replace existing user data context?", "Replace existing validations?", and "Replace existing special cmds?". In each set, "Yes" is selected.
- Buttons:** "Clear", "OK", "Add", and "Cancel" buttons are located at the bottom.
- Status Bar:** At the very bottom, it says "'Kintana User Data Context Migrator' parameters loaded."

For more detailed information on the Kintana User Data Context Migrator, see ["Kintana Migrators"](#).



# Appendix **E**

## Configuration Worksheets

This appendix provides worksheets that can be printed out and used to capture data required for configuring a Request resolution system in Kintana. Worksheets are provided for the following entities:

- Workflows
- Workflow Steps
- Request Header Types
- Request Types
- Request Type Fields and Commands
- Security Groups

For more information on any of the settings of entity parameters, refer to the appropriate Workbench Reference guide.

<b>Information on:</b>	<b>Kintana manual</b>
Workflows and Workflow Steps (screen and field info)	<i>"Configuration Workbench Reference"</i> (Workflow and Validation chapters)
Request Header Types	<i>"Create Workbench Reference"</i>
Request Types and Request Type Fields	<i>"Configuration Workbench Reference"</i> <i>"Configuration Workbench Reference"</i> (Validation chapter)
Request Type Commands	<i>"Using Commands and Tokens"</i>
Participant and Security	<i>"Kintana Security Model"</i>

Table E-1. Workflow Skeleton

Step No.	Step Name	Description	Type (Execution, Decision, Condition, or Subworkflow)	Transition Values	Validation
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					
18					
19					



Table E-2. Workflow Step [Execution] -- Step Number \_\_\_\_.

	Value
Step Name	
Goal / Result of Step	
<b>Validation*</b>	
<b>Execution Type**</b>	
Processing Type	
Timeout (Days)	
Source Environment (Group)	
Dest Environment (Group)	
Security (who can act on step):	
<ul style="list-style-type: none"> <li>• User Name</li> <li>• Standard Token</li> <li>• User Defined Token</li> </ul>	
Include Notification (Yes/No)	
Notification Event	
Notification Recipient:	
<ul style="list-style-type: none"> <li>• Username</li> <li>• Email Address</li> <li>• Security Group</li> <li>• Standard Token</li> <li>• User Defined Token</li> </ul>	
Notification Message	
Request Status at Step	
Request % Complete at Step	
Authentication Required (Y/N)	
Authentication Type (if Y)	

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

Execution Type**	Value
Built-in Workflow Event:	
<ul style="list-style-type: none"> <li>• Execute Commands</li> <li>• Close</li> <li>• Jump / Receive</li> <li>• Ready for Release</li> <li>• Return from Subworkflow</li> </ul>	
PL/SQL Function	
Token	
SQL Statement	
Workflow step commands	

Table E-3. Workflow Step [Decision] -- Step Number \_\_\_\_.

	Value
Step Name	
Goal / Result of Step	
<b>Validation*</b>	
Decisions Required (Vote on Step's outcome?)	<ul style="list-style-type: none"> <li>• One</li> <li>• At Least One</li> <li>• All</li> </ul>
Timeout (Days)	
Security (who can act on step):	
<ul style="list-style-type: none"> <li>• Security Group</li> <li>• User Name</li> <li>• Standard Token</li> <li>• User Defined Token</li> </ul>	
Include Notification (Yes/No)	
Notification Event	
Notification Recipient:	
<ul style="list-style-type: none"> <li>• Username</li> <li>• Email Address</li> <li>• Security Group</li> <li>• Standard Token</li> <li>• User Defined Token</li> </ul>	
Notification Message	
Request Status at Step	
Request % Complete at Step	
Authentication Required (Y/N)	
Authentication Type (if Y)	

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

Table E-4. Workflow Step [Sub-Workflow] -- Step Number \_\_\_\_\_.

	Value
Step Name	
Goal / Result of Step	
<b>Validation*</b>	
Vote on Step's outcome?	
Timeout (Days)	
Source Environment (Group)	
Dest Environment (Group)	
Security (who can act on step):	
<ul style="list-style-type: none"> <li>• Security Group</li> <li>• User Name</li> <li>• Standard Token</li> <li>• User Defined Token</li> </ul>	
Include Notification (Yes/No)	
Notification Event	
Notification Recipient:	
<ul style="list-style-type: none"> <li>• Username</li> <li>• Email Address</li> <li>• Security Group</li> <li>• Standard Token</li> <li>• User Defined Token</li> </ul>	
Notification Message	
Request Status at Step	
Request % Complete at Step	
Authentication Required (Y/N)	
Authentication Type (if Y)	

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

See “Using Subworkflows” on page 213 for notes on Validations for transitions into and out of Subworkflows.

Table E-5. Request Type Information.

	Value
Request Type Name	
Associated Request Header Type	
Description	

Table E-6. Request Type Commands

Goal of Commands	
Command Steps	
Conditions (When to execute)	

#	Field Names	Description
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		
16		

*Table E-7. Request Type Statuses*

<b>Status</b>	<b>Corresponds to Workflow Step</b>

Table E-8. Request Type Field Information

<b>Field Name</b>	
<b>Validation *</b>	
<b>Field Behavior:</b>	
<b>Attributes (select one):</b>	<ul style="list-style-type: none"> <li>• Display</li> <li>• Editable</li> <li>• Display Only</li> <li>• Required</li> </ul>
<b>Default Value</b>	
<b>Users/Security Groups allowed to View Field</b>	
<b>Users/Security Groups allowed to Edit Field</b>	
<b>Status Dependencies:</b>	
<b>Clear field when Status = ?</b>	
<b>Display only when Status = ?</b>	
<b>Reconfirm only when Status = ?</b>	
<b>Required when Status = ?</b>	
<b>Auto-Population Behavior:</b>	
<b>Auto-Population triggered by (Depends on) Field:</b>	
<b>Value to populate Field with:</b>	

Table E-9. Field Validation Information

<b>Validation Information*</b>	<b>Value</b>
Existing Validation?	
New Validation?	
Validation Type: (text field, auto-complete, drop down list, etc.)	
Validation Definition (list of values or SQL)	

Table E-10. Request Header Type Information

	Value
<b>Request Header Type Name</b>	
<b>Associated Request Type(s)</b>	
<b>Description</b>	
<b>Associated Field Group(s)</b>	

For creating new Request Header fields, use the worksheets for “*Request Type Field Information*” on page 354.

Table E-11. 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					

# Participant and Security

Table E-12. Security Groups.

Security Group Name	Members	Act on Workflow Steps	View/Edit Request Fields	Description



# Index

## A

Access Grants  
 removing 180

Adding Decision Steps 84  
 configuring notifications 88  
 general information 85  
 specifying security 87

Adding Execution Steps 89  
 configuring notifications 93  
 general information 89  
 specifying security 92

Adding Notifications to Workflow Steps 184  
 configuring message 198  
 configuring recipients 196  
 multiple notifications for single step 191  
 overview 184  
 sending reminders 195  
 smart URLs 200  
 smart URLs in HTML format 201  
 specifying intervals 192  
 tokens in message 200  
 when to send 186

Adding Sections  
 Request Types 156

Adding Steps 84

Adding Subworkflows 93

Adding Transitions 84, 94  
 all but one value 98  
 all results 99  
 back to same 102

based on table data 98  
 between Package workflows 106  
 error 100  
 field value 96  
 specific result 95  
 subworkflows 106  
 workflow parameters 103

Additional Resources  
 Kintana documentation 3  
 Kintana education 6  
 Kintana services 6  
 Kintana support 6

Advanced Configuration Guides 3

Advanced Default Rules  
 apply on creation 144  
 apply on field change 142, 144  
 apply on field change and stop processing other rules 143, 144

AND 227

Application  
 request header type 110

Archiving Kintana Configurations 29

Auto Complete Validations  
 example 263  
 limiting returned rows 266  
 user-defined multi-select 262

Auto-Complete Validations

256  
 command with delimited output 257  
 Command With Fixed Width Output 260

## B

Build Request Type  
 copying fields 132  
 modifying layout 154  
 removing fields 133  
 rules 135  
 setting max fields 134  
 status dependencies 146

Build Workflow  
 add steps and transitions 84  
 adding decision steps 84  
 adding execution steps 89  
 adding subworkflows 93  
 adding transitions 94  
 transitions 81  
 validations 81

Building Request Type 107  
 choosing request header type 109  
 configuring field behavior 122  
 creating a field 126  
 field validations 117  
 new request header type 111  
 overview 107  
 selecting validations 118

- Building Workflow 55
  - creating step sources 56
  - decision step source 61
  - execution step source 65
  - overview 55
  - step source creation overview 58
  - step source restrictions 60
- C**
- Choosing Request Header Type 109
- Command Validation 255
- Command with Delimited Output validation 257
- Command With Fixed Width Output validation 260
- Communication and Visibility 52
  - workflow step notifications 52
- Communication Paths 183
  - configuring reports 211
  - dashboard 207
- Component Type
  - file chooser (static environment override) 270
  - file chooser (token-based environment override) 271
- Component Types 244
  - directory chooser 269
  - file chooser 269
  - multi-select auto-complete 262
- Comprehensive
  - request header type 110
- Configuration Worksheets 347
- Configuring Reports 211
- Configuring Request Resolution
  - example 32
  - overview 31
- Context Field
  - changing 321
  - defining 319
- Context Sensitive
  - defining fields 321
  - editing fields 323
- Context Sensitive User Data
  - changing context field 321
  - changing context value 322
  - copying 323
  - defining context field 319
  - defining context value 320
  - defining fields 321
  - deleting 323
  - editing 321
  - editing fields 323
  - example for Request Header Type 324
  - generating 319
- Context Value
  - changing 322
  - defining 320
- Coping Request Header Type 115
- Copying Request Type Fields 132
  - create\_package 67
  - create\_package\_and\_wait 67
  - create\_request 67
- Creating a Field 126
- Creating Step Sources 56
  - decision 61
  - execution 65
  - overview 58
  - restrictions 60
- Custom Portlets 211
- D**
- Dashboard Configuration 207
  - controlling portlet access 207
  - custom portlets 211
  - default dashboard 210
- Decision Step Source
  - creating 61
  - general info 61
  - selecting validation 63
  - specifying default timeout 64
  - specifying voting requirements 63
- Decision Steps
  - adding to workflow 84
  - configuring notifications 88
  - general information 85
  - specifying security 87
- Default
  - request header type 110
- Default Dashboard 210
- Defining Business Flow 38
  - example 38
- Departmental
  - request header type 110
- Developing Kintana Configurations 23
- Directory Chooser 269
- Disabling Portlets 207
- Documentation 3
- Dynamic List Validations 253

- command 255
  - SQL 253
- E**
- Execution Step Source
    - close request failure 72
    - close request success 71
    - creating 65
    - defining executions 68
    - general information 66
    - PL/SQL function 75
    - returning from subwork-  
flow 74
    - select validation 81
    - specify default timeout 81
    - SQL statement 76
    - system-level commands  
78
    - token 77
  - Execution Steps
    - adding to workflow 89
    - configuring notifications  
93
    - general information 89
    - specifying security 92
  - Executions
    - and validations 83
    - defining 68
- F**
- Field Behavior 122
    - defaulting 124
    - editability 124
    - reconfirm 125
    - required 125
    - visibility 122
  - Field Logic 12
  - Field Security 129
- Fields**
- available types 118
  - behavior overview 122
  - building validations 121
  - changing column width  
155
  - creating 126
  - creating new in request  
header type 115
  - defaulting with rules 135
  - existing request header  
type 112
  - grayed out 152
  - modifying width 155
  - non-updateable 152
  - preview layout 157
  - removing from request  
types 133
  - selecting validations 118
  - setting maximum number  
134
  - setting security 129
  - status dependencies 146
  - validations 117
- File Chooser** 269
- static environment over-  
ride 270
  - token-based environment  
override 271
- First Line** 229
- G**
- Gathering Requirements 35
    - needed entities 36
- H**
- Help Desk
    - request header type 111
- I**
- Integrating Participants 159
    - removing access grants  
180
    - security groups 161
  - Integration 18
    - accelerators 20
    - dashboard 19
    - deliver 20
    - drive 20
    - packages 20
    - projects 20
    - solutions 18
    - tasks 20
- J**
- Jump Step
    - generating 222
  - Jump/Receive Step
    - pairing in Workflows 225
  - Jump/Receive Step Label
    - validation 220
  - Jump/Receive Step Labels  
219
- K**
- Key Concepts 9
  - Kintana Configurations
    - archiving 29
    - developing 23
    - migrating 26
  - Kintana Integration 18
- L**
- Last Line 231

Loop counter  
example in Workflow 237

## M

Migrating Kintana Configurations 26

how they work 26  
instance requirements 28  
usage overview 27

Modifying Active Workflows 232

copy and test 233  
disabling step 234  
execution steps 235  
move requests 233  
redirecting 234  
security/performance considerations 235  
verifying workflow logic 236

Modifying Request Type Layout 154

Moving Fields  
Request Types 155

Multiple Instances Overview 24

Multi-Select Auto-Complete  
user-defined 262

## N

Needed Entities 36  
request header type 37  
request type 36  
security groups 37  
workflows 36

Non-Updateable Field 152

Notifications

adding to workflow steps 184

configuring message 198  
configuring recipients 196  
on request field changes 202

smart URLs 200  
smart URLs in HTML messages 201  
tokens in message 200  
when to send 186

## O

OR 228

Ownership  
setting 178

## P

Package Workflow  
integration 218

PACKAGE\_URL 201

Parameters  
copy from 310  
Workflow 236

Participants 18

Participants and Security 45  
configuration security 47  
example 47  
request field security 47  
request type security 46  
workflow security 46

Portlets  
custom 211  
disabling 207  
restricting user access 209

Product Integration 18

Projects 331

## R

Receive Step  
generating 223

Reconfirm 152

Removing Request Type  
Fields 133

Reports 21  
configuring 211

Request 10  
commands 14  
field logic 12  
workflow integration 41  
workflow interaction 13

Request Creation Security 166  
enabling users 166  
request type restrictions 171  
workflow restrictions 170

Request Field Security 17

Request Header Type 11  
application 110  
choosing 109  
comprehensive 110  
copying 115  
creating fields 115  
creating new 111  
default 110  
definition 12, 110  
departmental 110  
help desk 111  
modifying existing fields 112  
simple 110

Request Header Type Field  
Window  
attributes tab 114  
default tab 114  
general information region 113

- security tab 115
  - storage tab 114
  - Request Header Type Fields
    - creating new 115
    - modifying existing 112
  - Request Processing Security 172
    - general access 172
    - participant restriction 176
    - workflow step security 175
  - Request Resolution 9
    - example 32
    - needed entities 36
    - overview 31
  - Request Resolution System
    - building workflow 55
    - communication and visibility 52
    - communication paths 183
    - configuration security 47, 177
    - configuration worksheets 347
    - gathering requirements 35
    - integrating participants 159
    - ownership 177
    - participants and security requirements 45
    - removing access grants 180
    - request creation security 166
    - request field security 47
    - request processing security 172
    - request type requirements 42
    - request type security 46
    - security 159
    - security groups 161
  - security overview 159
  - setting configuration security 178
  - setting ownership 178
  - workflow requirements 37
  - workflow security 46
  - workflow settings 56
  - workflow step notifications 52
  - Request Status
    - adding 148
    - linking 148
  - Request Statuses
    - assigning to workflow steps 153
  - Request Type 11
    - building 107
    - commands 14
  - Request Type Commands 14
    - executing 69
    - special commands 14
  - Request Type Field Window
    - attributes tab 128
    - defaults tab 128
    - general information region 127
    - security tab 129
    - storage tab 129
  - Request Type Fields 42
    - available types 118
    - building validations 121
    - copying 132
    - creating 126
    - example 43
    - notifications 202
    - removing 133
    - selecting validations 118
    - setting maximum number 134
    - setting security 129
    - validations 117
  - Request Type Layout 154
    - adding sections 156
    - modifying 155
    - moving fields 155
    - preview 157
  - Request Type Requirements 42
    - request header type 44
    - request type commands 45
    - request type fields 42
    - request workflow interaction 44
  - Request Type Rules 135
    - advanced default rules 141
    - simple default rules 137
  - Request Types
    - adding sections 156
    - modifying fields 155
    - notifications on field changes 202
  - Request Workflow
    - integration 218
  - Request Workflow Interaction 13
  - REQUEST\_URL 201
  - Requests
    - creation security 166
    - processing security 172
    - setting reopen workflow step 231
  - Rules 135
- S**
- Screen Security 17
  - Security
    - participants 18
    - removing access grants 180

- request creation 166
  - request fields 17
  - request processing 172
  - screen 17
  - workflow step 17
  - Security Groups 16, 161
    - specifying users 161
    - using resource management 164
  - Selecting Field Validations 118
    - available types 118
    - building validations 121
  - Setting Max Request Type Fields 134
  - Simple
    - request header type 110
  - Special Commands 14
  - SQL Validations 253
    - tips 255
  - Static List Validations 251
  - Status Dependencies 146, 152
    - adding statuses 148
    - assigning statuses to workflow steps 153
    - clear 152
    - configuring dependency 150
    - creating statuses 147
    - interactions 153
    - linking statuses 148
    - non-updateable fields 152
    - reconfirm 152
    - required 151
    - updateable 152
    - visible 151
  - Status Dependencies Interactions 153
  - Subworkflows
    - adding to workflow 93
    - example 40
    - overview 213
    - transitioning out of 216
    - transitioning to 214
    - usage 40
  - SYNC 228
- T**
- Table Component Validations 272
    - adding to request type 283
    - column totals 281
    - creating rules 276
    - defining 273
    - rules example 277
    - tokens 281
  - Tasks
    - user data roll-up 331
  - Text Area 316
  - Timeout
    - default for executions 81
    - defaulting for decisions 64
  - Token 15
  - Token Evaluation
    - example 263
  - Tokens
    - types 305
  - Transitions 81
    - adding to workflow 94
    - all but one value 98
    - all results 99
    - back to same 102
    - based on error 100
    - based on table data 98
    - between Package workflows 106
    - field value 96
    - specific result 95
    - subworkflows 106
    - workflow parameters 103
- U**
- URL to Validation 250
  - User Data
    - adding fields 309
    - changing field width 316
    - configuring field dependencies 312
    - context sensitive in validation 247
    - copying fields 310
    - creating project-task roll-up 331
    - deleting project-task roll-up 337
    - editing fields 311
    - editing project-task roll-up 335
    - migrating 344
    - migrating contexts 345
    - migrating values 344
    - moving fields 316
    - overview 307
    - preview layout 317
    - project task roll-up 331
    - referring to 344
    - removing fields 314
    - roll-up example 338
    - supported functionality 307
    - text area 316
    - using in the Kintana product suite 344
    - user data roll-up 331
  - Using Multiple Kintana Instances 24
    - new implementation 25
    - single prod instance 25

**V**

Validation 15

Validations 81

and executions 83

auto-complete 256

Command 255

Command With Delimited Output 257

Command With Fixed Width Output 260

context sensitive user data and 247

creating 247

defined 244

deleting 251

directory chooser 269

dynamic list 253

editing 249

file chooser 269

file chooser (static environment override) 270

file chooser (token-based environment override) 271

overview 244

package and request group 285

quick link 250

request type category 286

seeded 287

special characters and 287

SQL 253

SQL tips 255

static lists 251

system 287

table component 272

text area 1800 271

Voting

All 64

At Least One 63

One 63

**W**

wf\_close\_failure 67

wf\_close\_success 67

wf\_jump 67, 219

wf\_receive 67, 219

wf\_return 67

WORKBENCH\_PACKAGE\_URL 201

Workflow 13

building 55

building overview 55

condition steps 227

jump/receive 218

modifying while in use 232

package request integration 218

request interaction 13

sample 13

setting reopen step for requests 231

settings for request resolution 56

step security 175

worksheet 348

Workflow Integration 218

jump step source 222

jump/receive pair 225

jump/receive step label 220

receive step source 223

Workflow Parameters 236

example 237

generating 236

Workflow Requirements 37

business flow 38

request statuses 41

request workflow interaction 44

step information 39

subworkflows 40

Workflow Step Security 17, 175

Workflow Steps

adding notifications 184

assigning statuses 153

conditions 227

security 175

Worksheets 347

decision workflow step 350

execution workflow step 349

existing request header type fields 355

participants and security 356

request header type 355

request type 352

request type field 354

request type statuses 353

subworkflow step 351

workflow 348