

# **Kuali Rice 2.5.6 KIM Guide**

---

---

---

# Table of Contents

1. KIM Overview .....	1
KIM Features .....	1
2. Person .....	3
Person Lookup .....	3
Finding the Person Lookup Screen .....	3
Person Lookup Options .....	3
Person Maintenance .....	5
Business Rules .....	5
Routing .....	5
Displaying the Person Lookup Screen .....	5
Document Layout .....	6
Document Overview Tab .....	6
Overview Tab .....	6
Contact Tab .....	8
Privacy Preferences Tab .....	10
Membership Tab .....	11
Ad Hoc Recipients Tab .....	12
Route Log Tab .....	12
3. Group .....	13
Group Lookup Screen .....	13
Group Inquiry Screen .....	14
Overview Tab .....	14
Assignees Tab .....	14
Group Maintenance Document .....	15
Creating New Groups .....	15
Group Maintenance Document: Layout .....	15
4. Role .....	18
Role Lookup Screen .....	18
Document Layout .....	18
Role Maintenance Document .....	19
Role Maintenance Document .....	19
5. KIM Type .....	25
KIM Type Lookup .....	25
KIM Type Inquiry .....	26
6. Responsibility .....	27
Responsibility Lookup .....	27
Responsibility Inquiry .....	29
7. Permission .....	31
Permission Lookup .....	31
Permission Inquiry .....	32
Permission Template Inquiry .....	33
Delivered Permission Templates .....	33
8. Terminology .....	37
Principal .....	37
Entity .....	37
Group .....	37
Permission .....	37
Responsibility .....	38
Role .....	38
Reference Information .....	38
Configuration Parameters .....	39

9. Services .....	40
Using the Services .....	40
IdentityService .....	40
Retrieving Principal Information .....	40
Retrieving Entity Default Information .....	41
Retrieving Reference Information .....	41
GroupService .....	42
Retrieving Group Membership Information .....	42
Retrieving Group Information .....	42
PermissionService .....	42
Checking Permission .....	42
Retrieving Permission Information .....	42
ResponsibilityService .....	43
Checking Responsibility .....	43
Retrieving Responsibility Information .....	43
AuthenticationService .....	43
Checking Authentication .....	43
RoleService .....	44
Checking Role Assignment .....	44
Retrieving Role Information .....	44
Updating Role Membership .....	44
Person Service .....	44
Retrieving Personal Information .....	45
10. KimTypeService Callbacks .....	46
Implementing Custom KIM Types .....	46
Configuring Custom KIM Types .....	46
Publishing Custom KIM Types to the Kuali Service Bus .....	48
11. KIM Database Tables .....	49
Table Name Prefixes .....	49
Unmapped LAST_UPDT_DT Columns .....	49
Glossary .....	50

---

# List of Figures

- 1.1. KIM Architecture ..... 1
- 1.2. Detailed KIM Architecture ..... 2
- 2.1. Person Lookup ..... 3
- 2.2. Person Lookup: Results ..... 4
- 2.3. Person Document ..... 4
- 2.4. Person Lookup: Create New Button ..... 4
- 2.5. Person Document ..... 6
- 2.6. Person Document: Overview Section ..... 6
- 2.7. Person Document: Overview Tab, Affiliations Section ..... 7
- 2.8. Person Document: Contact Tab ..... 8
- 2.9. Person Document: Contact Tab, Names Section ..... 8
- 2.10. Person Document: Contact Tab, Addresses Section ..... 9
- 2.11. Person Document: Contact Tab, Phone Numbers Section ..... 9
- 2.12. Person Document: Contact Tab, Email Addresses Section ..... 10
- 2.13. Person Document: Privacy Preferences Tab ..... 10
- 2.14. Person Document: Memberships Tab ..... 11
- 3.1. Identity Channel: Group Link ..... 13
- 3.2. Group Lookup ..... 13
- 3.3. Group Lookup: Results ..... 14
- 3.4. Group Inquiry ..... 14
- 3.5. KIM Type Lookup ..... 15
- 3.6. Group Maintenance Document ..... 16
- 3.7. Group Maintenance Document: Group Overview ..... 16
- 3.8. Group Maintenance Document: Assignees Tab ..... 17
- 4.1. Role Lookup ..... 18
- 4.2. Role Maintenance Document ..... 19
- 4.3. Role Maintenance Document: Tabs ..... 20
- 4.4. Role Maintenance Document: Overview Tab ..... 20
- 4.5. KIM Type Lookup ..... 21
- 4.6. Role Maintenance Document: Permissions Tab ..... 21
- 4.7. Role Maintenance Document: Permissions Tab, Add Permissions ..... 22
- 4.8. Role Maintenance Document: Responsibilities Tab ..... 23
- 4.9. Role Maintenance Document: Responsibility, Added Responsibility ..... 23
- 4.10. Role Maintenance Document: Responsibility Tab, Action Section ..... 24
- 5.1. KIM Type Lookup ..... 25
- 5.2. KIM Type Lookup: Results Example ..... 25
- 5.3. KIM Type Inquiry ..... 26
- 6.1. Identity Channel: Responsibility Link ..... 27
- 6.2. Responsibility Lookup ..... 28
- 6.3. Responsibility Lookup: Results ..... 29
- 6.4. Responsibility Inquiry ..... 30
- 7.1. Permission Lookup ..... 31
- 7.2. Permission Lookup: Results Example ..... 32
- 7.3. Permission Inquiry ..... 33
- 7.4. Permission Template Inquiry ..... 33

---

# List of Tables

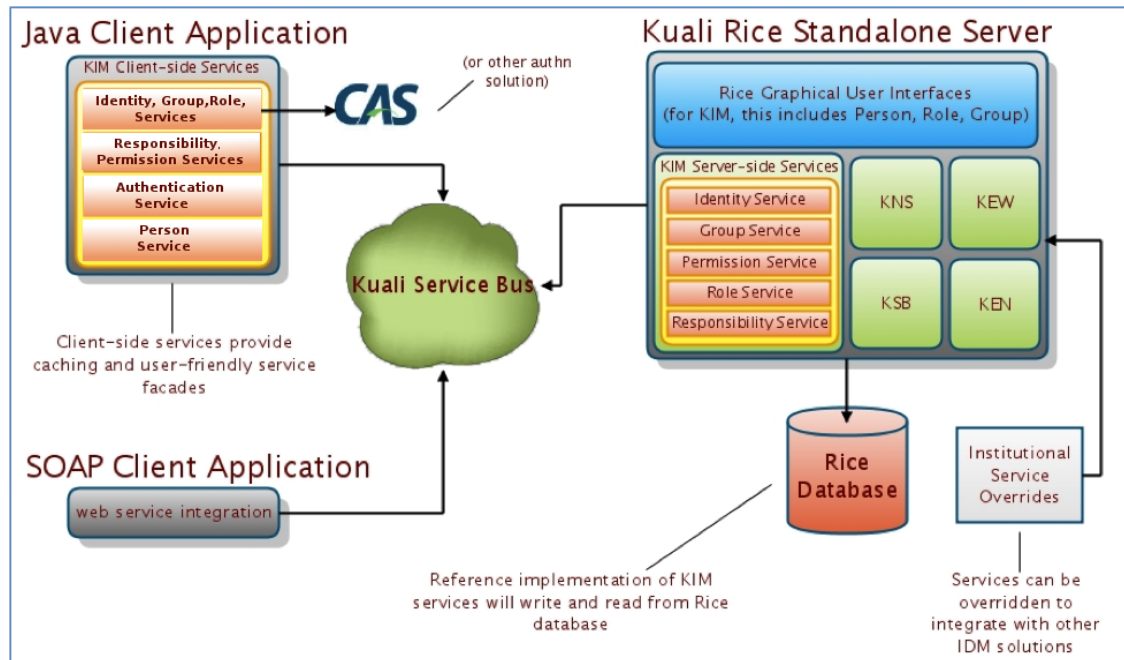
- 2.1. Person Document: Overview Attributes ..... 6
- 2.2. Person Document: Overview Attributes, Affiliations ..... 7
- 2.3. Person Document: Overview Attributes, Affiliations Continued ..... 7
- 2.4. Person Document: Contact Tab, Names Section Attributes ..... 8
- 2.5. Person Document: Contact Tab, Address Section Attributes ..... 9
- 2.6. Person Document: Contact Tab, Phone Numbers Attributes ..... 9
- 2.7. Person Document: Contact Tab, Email Address Attributes ..... 10
- 2.8. Person Document: Privacy Preferences Tab Attributes ..... 10
- 2.9. Person Document: Memberships Tab, Groups Attributes ..... 11
- 2.10. Person Document: Memberships Tab, Roles Attributes ..... 11
- 3.1. Group Inquiry: Assignees Attributes ..... 14
- 3.2. KIM Type Lookup Search Attributes ..... 15
- 3.3. Group Maintenance Document: Group Overview Attributes ..... 16
- 3.4. Group Maintenance Document: Assignees Tab Attributes ..... 17
- 4.1. Role Maintenance Document: Overview Attributes ..... 20
- 4.2. Role Maintenance Document: Permissions Attributes ..... 22
- 4.3. Role Maintenance Document: Permissions Tab, Add Attributes ..... 22
- 4.4. Role Maintenance Document: Responsibility Attributes ..... 23
- 4.5. Role Maintenance Document: Responsibility, Add Attributes ..... 23
- 4.6. Role Maintenance Document: Responsibility Tab, Action Section Attributes ..... 24
- 5.1. KIM Type Lookup Attributes ..... 25
- 5.2. KIM Type Inquiry Attributes ..... 26
- 6.1. Responsibility Lookup Attributes ..... 28
- 6.2. Responsibility Lookup: Results Attributes ..... 29
- 7.1. Permission Lookup Attributes ..... 31
- 7.2. Permission Lookup: Results Attributes ..... 32
- 7.3. Delivered Permission Templates ..... 34
- 8.1. KIM Configuration Parameters ..... 39

# Chapter 1. KIM Overview

Kuali Identity Management (KIM) provides identity and access management services to Rice and other applications. All KIM services are available on the service bus with both SOAP endpoints. KIM provides a service layer and a set of GUIs that you can use to maintain the identity information.

KIM is designed to be used with both Kuali and non-Kuali applications. The permissions and responsibilities it provides are defined by each user's Role or Roles in the system. Roles can be customized to handle permissions and responsibilities in a variety of ways based on your particular needs.

**Figure 1.1. KIM Architecture**



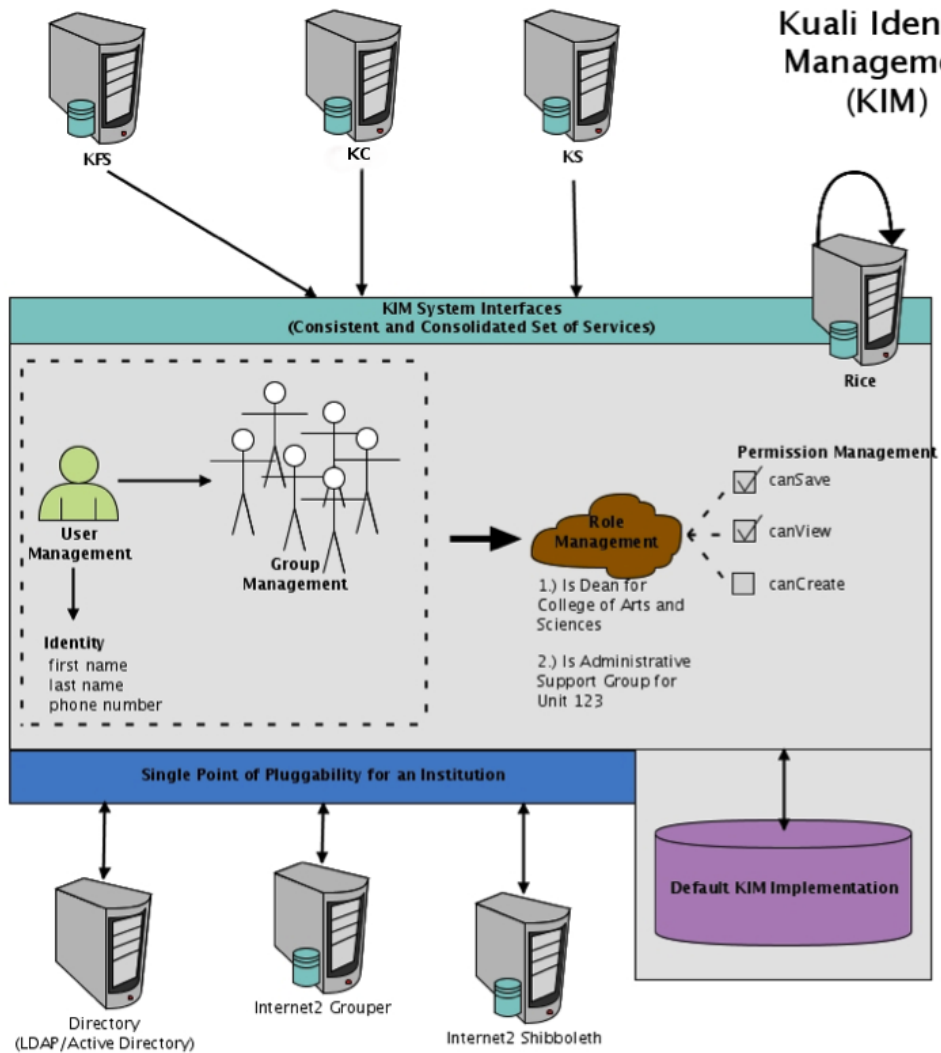
## KIM Features

- KIM provides a reference implementation of the services but allows for customization and/or replacement to facilitate integration with institutional services or other third party identity management solutions.
- KIM allows you to override its core services.
- KIM consists of these services, which encompass its API:
  - AuthenticationService
  - GroupService
  - IdentityService
  - PermissionService
  - PersonService

- ResponsibilityService
- RoleService
- KimTypeInfoService
- KIM evaluates permissions through its permission service. KIM provides plug points for implementing custom logic for permission checking, such as permission checks based on hierarchical data.

A more detailed picture of the KIM architecture:

**Figure 1.2. Detailed KIM Architecture**





---

# Chapter 2. Person

## Person Lookup

Use the **Person Lookup** screen to quickly find basic information about Persons in Quali Rice (Rice). (In Quali, a Person is a set of information about a real person or something that stands for real people, like a job title.) From the Person Lookup screen, you can also link to screens where you can create new Persons and edit a Person's information if you have permission to do so.

### Finding the Person Lookup Screen

You can go to the **Person Lookup** screen from many Quali screens by clicking the field search button next to a Name-related field. How you get to the Person Lookup screen and your **Role** in Quali determine what you see and what you can do there.

If you have permission to use the Administration screens in Rice, you can go directly to the **Person Lookup** screen:

1. Click the **Administration** tab
2. Look under **Identity**
3. Click **Person**

### Person Lookup Options

#### From the Field Search Button

If you clicked the field search button on another Quali page to get to the Person Lookup screen, your screen looks like this:

**Figure 2.1. Person Lookup**

The screenshot shows the 'Person Lookup' interface. At the top left is the title 'Person Lookup' with a help icon. At the top right is a 'create new' button and a note '\* required field'. The main area contains a list of search fields: Principal Name, Principal ID, Entity ID, First Name, Middle Name, Last Name, Email Address, Phone Number, Employee ID, Campus Code, Primary Department Code, Employee Status Code, Employee Type Code, and Active Indicator. The Active Indicator field has radio buttons for Yes, No, and Both. At the bottom are 'search', 'clear', and 'cancel' buttons.

To find a Person, enter some information about that person in the appropriate fields and click the **search** button at the bottom of the screen.

- You can display a list of all Persons in KIM if you leave the search fields empty and set the **Active Indicator** to **Both**, then click the search button.

- Rice displays a list of Persons who match your search information. The list is just below the search fields on the Person Lookup page:

**Figure 2.2. Person Lookup: Results**

Person Lookup create new  
\* required field

Principal Name:   
 Principal ID:   
 Entity ID:   
 First Name:   
 Middle Name:   
 Last Name:   
 Email Address:   
 Phone Number:   
 Employee ID:   
 Campus Code:   
 Primary Department Code:   
 Employee Status Code:    
 Employee Type Code:    
 Active Indicator:  Yes  No  Both

5 items retrieved, displaying all items.

Actions	Principal ID	Principal Name	Name	Entity ID	Campus Code	Primary Department Code	Employee ID
<a href="#">edit</a>	<a href="#">earl</a>	earl	Employee_Earl	1116			
<a href="#">edit</a>	<a href="#">edna</a>	edna	Employee_Edna	1114			
<a href="#">edit</a>	<a href="#">employee</a>	employee	employee_employee	1137			
<a href="#">edit</a>	<a href="#">eric</a>	eric	Employee_Eric	1115			
<a href="#">edit</a>	<a href="#">erin</a>	erin	Employee_Erin	1117			

Export options: [CSV](#) | [spreadsheet](#) | [XML](#)

Click one of the underlined column titles on the search results list to sort the list by that column. Click it again to sort the list in the opposite order.

You can also export the search results list in CSV, spreadsheet, or XML format. Click the appropriate link at the bottom of the search results list to do this.

If you need to see more information about one of the Persons in the search results list, click one of the information fields for that Person. Kuali then displays the **Person Document** screen for that Person.

**Figure 2.3. Person Document**

Person expand all collapse all

Overview hide

Entity ID: 1116 Principal ID: earl  
 Principal Name: earl Active: Yes

Affiliation Type	Campus Code	Default

Contact show

Privacy Preferences show

Membership show

## From the Administration Tab

If you come to the **Person Lookup** screen after clicking **Person** (under **Identity**) on the **Administration** tab, you may see a **create new** button in the upper right corner of your screen if the user logged in has permissions to create a Person, like this:

**Figure 2.4. Person Lookup: Create New Button**

Person Lookup create new

You can click this create new button to go to the **Person Document** screen, where you can add a new Person as a Kualu user. For more information on adding a Person, see the **Person Maintenance** function section (next) in this User Guide.

If you need to edit information for a Person in KIM, you first need to display that Person's information. Search for the Person (see search instructions above) from the **Administration** tab's link for **Person**. When Kualu displays the list of Persons that match your search, you'll notice one difference in the columns because you did your search from the Administration tab – there is an **Actions** column on the left.

The **Actions** column has an **edit** link for each person. Click this edit link to go to the **Person Document** screen where you can edit this Person's information. For more information on working with Person information, see the **Person Maintenance** section (next) in this User Guide.

## Person Maintenance

The Person Document allows you to identify and maintain each user in KIM. Each Person Document includes data about that user's relationship with your institution as well as the roles and groups to which the person belongs.

In KIM, a Person is a unique combination of an *Entity ID* and a *Principal ID*. The Entity ID represents a Person with a unique number, and the Person Document associates the Entity ID with the user's Principal ID number and Principal Name (often referred to as a user name or user ID). When searching for or working with users in KIM, you usually use either the Principal ID or the Principal Name. A single Entity ID can have multiple Principals associated with it, but the base implementation of KIM assumes that each Entity ID has only a single Principal.

### Note

**Person and HR Systems:** Many institutions choose to override parts of the Person Document (especially the affiliation and contact information) with data from an HR system.

## Business Rules

- Each Person must have at least one Affiliation.
- Each faculty or staff affiliation must have at least one Employment Information record associated with it.
- If a Person has any faculty or staff affiliations, then one Employment Information record must be marked as Primary.
- Each Person must have a default Name record in the Contacts section.
- Each affiliation must be associated with a Campus.

## Routing

Only people with the appropriate role can initiate routing for Person documents.

## Displaying the Person Lookup Screen

On the **Administration** tab, click **Person** to display the **Person Lookup** screen. Do a lookup to see a list of Persons. Click any Person on the search results list to display his or her **Person Document** screen.

## Document Layout

The Person Document includes **Document Overview**, **Overview**, **Contact**, **Privacy Preferences**, **Membership**, **Ad Hoc Recipients**, and **Route Log** tabs.

**Figure 2.5. Person Document**

## Document Overview Tab

The **Document Overview** tab combined with the **Overview** tab identifies the **Person** as a unique combination of Entity and Principal ID.

Additional information on the **Document Overview Tab** can be found in the [Document Overview Tab](#) section of the Introduction to Rice Guide.

## Overview Tab

The **Overview** tab identifies the **Person** as a unique combination of **Entity ID** and **Principal ID**. It also contains information about how this Person is affiliated with your institution. Two types of affiliations—**Staff** and **Faculty**—contain additional data fields to further define a person's relationship with your institution.

The instructions below assume that you are manually entering this information. Many institutions either have this data fed from an existing Person database or simply override this information with existing Person data.

**Figure 2.6. Person Document: Overview Section**

## Overview Section

**Table 2.1. Person Document: Overview Attributes**

Field Name	Description
Entity Id	Display-only. The unique ID number identifying this Person in your database. An individual may have multiple Principal IDs but only one Entity ID. The base implementation assumes that each user

## Person

Field Name	Description
	has only one Entity ID and one Principal ID. KIM automatically assigns an Entity Id to a new Person when you save or submit a Person Document.
Principal Id	Display-only. The unique ID number identifying this Principal. Whereas Entity Id represents a unique Person, Principal Id represents a set of login information for that Person. When selecting a Person, you ordinarily reference his or her Principal Id. KIM automatically assigns a Principal Id to a new Person when you save or submit the Person Document.
Principal Name	Required. The user name for this Principal
Active	Check the box to indicate that this Principal ID is Active. Uncheck the box to indicate that this Principal ID is Inactive.

## Affiliations Section

Use the **Affiliations** section of the **Overview** tab to add Affiliations for this Principal ID. Depending on the Affiliation type you select, you may need to complete additional fields.

**Table 2.2. Person Document: Overview Attributes, Affiliations**

Field Name	Description
Affiliation Type	Optional. Select the Type of Affiliation from the list. Options: <ul style="list-style-type: none"> <li>• <b>Affiliate:</b> Users in your system who are neither employees nor students</li> <li>• <b>Faculty:</b> A faculty employee</li> <li>• <b>Staff:</b> A non-faculty employee</li> <li>• <b>Student:</b> A non-employee identified as a student of your institution</li> </ul> Affiliation types of Faculty and Staff require additional information (see below).
Campus Code	Required. Select the Campus Code associated with this Affiliation
Default	Check the box to indicate that this Affiliation is this Principal's default association with your institution. Each Principal must have at least one default Affiliation.
Actions	Click the <b>Add</b> button to save the affiliation information.

If you select an Affiliation of Faculty or Staff, KIM displays additional fields to collect employment information:

**Figure 2.7. Person Document: Overview Tab, Affiliations Section**

Affiliations				
	* Affiliation Type	* Campus Code	Default	Actions
Add:	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	<input type="button" value="add"/>

**Table 2.3. Person Document: Overview Attributes, Affiliations Continued**

Field Name	Description
Employment ID	Optional. Enter the Employment ID number associated with this Faculty or Staff affiliation. Ordinarily this entry is the ID number identifying this Principal in your HR system.
Primary	Check this box to indicate that this Faculty or Staff affiliation represents the Principal's primary job with your institution. Each Principal with a Faculty or Staff affiliation must have exactly one affiliation marked as Primary.
Employee Status	Required. Select the current status of this Faculty or Staff affiliation. Options: <ul style="list-style-type: none"> <li>• Active</li> <li>• Deceased</li> <li>• On Non-Pay Leave</li> <li>• Status Not Yet Processed</li> <li>• Processing</li> <li>• Retired</li> </ul>

## Person

Field Name	Description
	<ul style="list-style-type: none"> <li>Terminated</li> </ul>
Employee Type	Required. Select the type of employment for this Affiliation. Options: <ul style="list-style-type: none"> <li>Non-Professional</li> <li>Other</li> <li>Professional</li> </ul>
Base Salary Amount	Required. Enter the base annual salary for this Faculty or Staff Affiliation.
Primary Department Code	Optional. Enter the code for the Department associated with this Faculty or Staff affiliation.
Add	Click the <b>Add</b> button to save this row of employment information.

## Contact Tab

The **Contact** tab records the names, addresses, phone numbers, and email addresses associated with this Person. Any Person record can store multiple contact information records of each type (name, address, phone number, and email address). You must select one value of each type as the default for that Person record.

**Figure 2.8. Person Document: Contact Tab**

## Names Section

**Figure 2.9. Person Document: Contact Tab, Names Section**

Names									
	Name Code	Name Prefix	* First Name	Middle Name	* Last Name	Name Suffix	Default	Active	Actions
Add:	Other						<input type="checkbox"/>	<input checked="" type="checkbox"/>	add

**Table 2.4. Person Document: Contact Tab, Names Section Attributes**

Field Name	Description
Name Code	Optional. Select the type of name to be added in this row. Options: <ul style="list-style-type: none"> <li>Other</li> <li>Preferred</li> <li>Primary</li> </ul>
Name Prefix	Optional. Select the appropriate title or name prefix for the name being added in this row. Options: <ul style="list-style-type: none"> <li>Ms</li> <li>Mrs</li> <li>Mr</li> <li>Dr</li> </ul>
First Name	Required. Enter the first name for this record.
Middle Name	Optional. Enter the first name for this record.

## Person

Field Name	Description
Last Name	Required. Enter the last name for this record.
Name Suffix	Optional. Select a suffix for this name record. Options: <ul style="list-style-type: none"> <li>• Jr</li> <li>• Sr</li> <li>• Mr</li> <li>• MD</li> </ul>
Default	Check this box to indicate that this Name record is the default for this person. Each Person record must have exactly one Name record identified as the default.
Active	Check the box to indicate that this Name record is Active. Uncheck the box to indicate that this record should be considered Inactive.
Actions	Click the <b>Add</b> button to save this Name record.

## Addresses Section

**Figure 2.10. Person Document: Contact Tab, Addresses Section**

Addresses											
Add:	Address Type	Line 1	Line 2	Line 3	City	State/Province	Postal Code	Country	Default	Active	Actions
	Home								<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="button" value="add"/>

**Table 2.5. Person Document: Contact Tab, Address Section Attributes**

Field Name	Description
Address Type	Optional. Select the type of address being added on this row. Options: <ul style="list-style-type: none"> <li>• Home</li> <li>• Other</li> <li>• Work</li> </ul>
Line 1 to 3	Optional. Use lines 1, 2, and 3 to enter the street address for this row.
City	Optional. Enter the city associated with this address.
State/Province	Optional. Select the state associated with this address from the list.
Postal Code	Optional. Enter the postal code (zip code in the U.S.) associated with this address.
Country	Optional. Select the country associated with this address.
Default	Check this box to indicate this address record should be used as the Default. A Person record can have only one Default Address record.
Active	Check this box to indicate that this Address record is Active. Uncheck the box to indicate that this record is Inactive.
Actions	Click the <b>Add</b> button to save this Address record.

## Phone Numbers Section

**Figure 2.11. Person Document: Contact Tab, Phone Numbers Section**

Phone Numbers							
Add:	Phone Type	Phone Number	Extension	Country	Default	Active	Actions
	Home				<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="button" value="add"/>

**Table 2.6. Person Document: Contact Tab, Phone Numbers Attributes**

Field Name	Description
Phone Type	Optional. Select the type of phone number being added on this row. Options: <ul style="list-style-type: none"> <li>• Home</li> <li>• Mobile</li> </ul>

## Person

Field Name	Description
	<ul style="list-style-type: none"> <li>• Other</li> <li>• Work</li> </ul>
Phone Number	Optional. Enter the area code and phone number.
Extension	Optional. Enter the appropriate extension.
Country	Optional. Select the country associated with this Phone Number record.
Default	Check this box to indicate that this Phone Number record should be used as the Default. A Person record can have only one default Phone Number record.
Active	Check this box to indicate that this Phone Number record is Active. Uncheck the box to indicate that this record is inactive.
Actions	Click the <b>Add</b> button to save this Phone Number record.

## Email Addresses Section

**Figure 2.12. Person Document: Contact Tab, Email Addresses Section**

**Table 2.7. Person Document: Contact Tab, Email Address Attributes**

Field Name	Description
Email	Optional. Enter the email address for this record.
Type	Optional. Select the type of email address on this row. Options: <ul style="list-style-type: none"> <li>• Home</li> <li>• Other</li> <li>• Work</li> </ul>
Default	Check this box to indicate that this Email Address record should be used as the Default. A Person record can have only one default Email Address record.
Active	Check this box to indicate that this Email Address record is Active. Uncheck the box to indicate that this record is Inactive.
Actions	Click the <b>Add</b> button to save this Email Address record.

## Privacy Preferences Tab

The **Privacy Preferences** tab allows you to suppress the display of (hide) fields on the **Contact** Tab.

**Figure 2.13. Person Document: Privacy Preferences Tab**

**Table 2.8. Person Document: Privacy Preferences Tab Attributes**

Field Name	Description
Suppress Name	Optional. Check this box to specify NOT to display this person's names.
Suppress Personal	Optional. Check this box to specify NOT to display any of this person's personal data.
Suppress Phone	Optional. Check this box to specify NOT to display this person's phone numbers.
Suppress Address	Optional. Check this box to specify NOT to display this person's addresses.
Suppress Email	Optional. Check this box to specify NOT to display this person's email addresses.



## Membership Tab

The **Membership Tab** allows you to associate a person with groups and roles and, by extension, with KIM permissions and responsibilities. Assigning a person to a role is the most direct way to give him or her KIM permissions and responsibilities.

**Figure 2.14. Person Document: Memberships Tab**

The screenshot shows the Membership Tab interface. It is divided into three main sections: Groups, Roles, and Delegations. Each section has a 'hide' button and an 'Add:' button. The Groups section has columns for Group, Namespace Code, Name, Type, Active From Date, Active To Date, and Actions. The Roles section has columns for Role, Namespace Code, Name, Type, Active From Date, Active To Date, and Actions. The Delegations section has columns for Role Member, Active From Date, Active To Date, Delegation Type Code, and Actions.

The Membership tab is divided into two sections, one for managing assignments to **Groups** and another for **Roles**.

## Groups Section

**Table 2.9. Person Document: Memberships Tab, Groups Attributes**

Field Name	Description
Group	Optional. Enter the name of the KIM group you want to assign this person to. You can also use the Group lookup to search for and select a valid Group.
Namespace Code	Display-only. After you select a group to add this person to, KIM displays the namespace code associated with the group you selected.
Name	Display-only. After you select a group to add this person to, KIM displays the name of that group.
Type	Display-only. After you select a group to add this person to, KIM displays the type of that group.
Active From Date	Optional. If this user's assignment to this group is to be effective as of a certain date, enter that date here.
Active To Date	Optional. If this user's assignment to this group is to terminate as of a certain date, enter that date here.
Actions	Click the <b>Add</b> button to add this group assignment.

### Note

There is no way to delete a person's assignment to a group. To remove a person from a group, use the Active To Date field to specify a date in the past.

## Roles Section

**Table 2.10. Person Document: Memberships Tab, Roles Attributes**

Field Name	Description
Role	Optional. Use the Name lookup to search for and select the role you want to assign this person to.
Namespace Code	Display-only. After you select a role to assign to this Person record, KIM displays the namespace code associated with that role.
Name	Display-only. After you select a role to assign to this Person record, KIM displays the name associated with that role.
Type	Display-only. After you select a role to assign to this Person record, KIM displays the role type for the selected role.
Active From Date	Optional. If this user's assignment to this role is to be effective as of a certain date, enter that date here.
Active To Date	Optional. If this user's assignment to this role is to terminate as of a certain date, enter that date here.

Field Name	Description
Actions	Click the <b>Add</b> button to save this role data.

## Note

There is no way to delete a person's assignment to a role. To remove a person from a role, use the Active To Date field to specify a date in the past.

When assigning some roles, you may need to supply additional qualifying information that further defines this person's assignment. For more information on Roles see the Role Maintenance section in this User Guide.

## Ad Hoc Recipients Tab

Ad Hoc Recipients tab information can be found in the Common Features and Functions section of this User Guide.

## Route Log Tab

Route Log tab information can be found in the Common Features and Functions section of this User Guide.

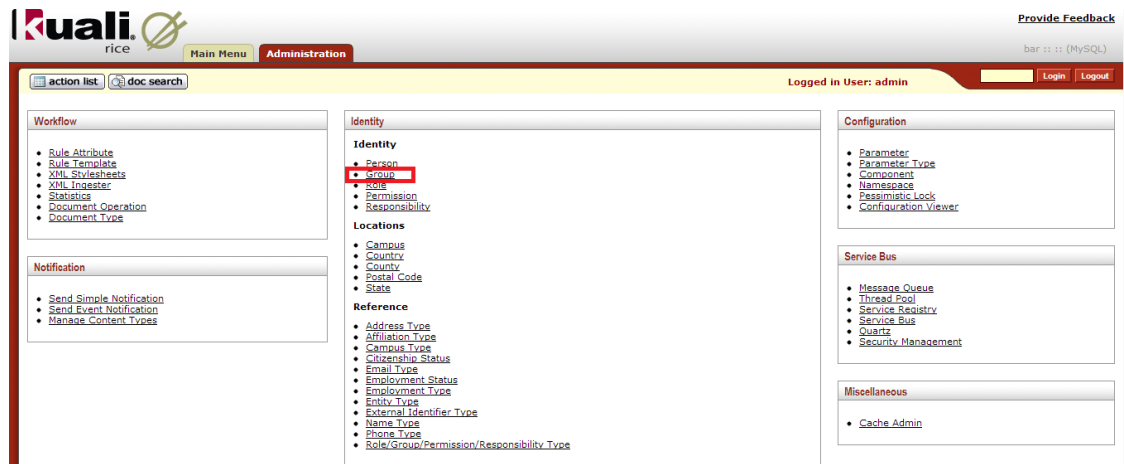
# Chapter 3. Group

## Group Lookup Screen

The Group Lookup function provides a quick reference to key Group information.

You get to this screen by clicking the **Group** option in the **Identity** section of the **Administration** screen.

**Figure 3.1. Identity Channel: Group Link**



**Figure 3.2. Group Lookup**

**Group Lookup** [create new](#)

\* required field

<b>Group Id:</b>	<input style="width: 90%;" type="text"/>
<b>Group Namespace:</b>	<input style="width: 90%;" type="text"/> ▼
<b>Group Name:</b>	<input style="width: 90%;" type="text"/>
<b>Group Description:</b>	<input style="width: 90%;" type="text"/>
<b>Type:</b>	<input style="width: 90%;" type="text"/> ▼
<b>Active?:</b>	<input checked="" type="radio"/> Yes <input type="radio"/> No <input type="radio"/> Both
<b>Principal Name:</b>	<input style="width: 90%;" type="text"/> 🔍
<input type="button" value="search"/> <input type="button" value="clear"/> <input type="button" value="cancel"/>	

On the **Group Lookup** screen, click the **create new** button to go to the **Group Maintenance** screen, where you can create a new Group in Rice.

Conducting a search from the Group Lookup screen returns a list similar to this:

### Figure 3.3. Group Lookup: Results

5 items retrieved, displaying all items.

Actions	Group Type Name	Group Namespace	Group Name	Group Description
<a href="#">edit</a>	Default	KR-WKFLW	WorkflowAdmin	WorkflowAdmin
<a href="#">edit</a>	Default	KR-WKFLW	MagazineManagers	The managers of the magazine system.
<a href="#">edit</a>	Default	KR-WKFLW	ChickenRecipeMasters	The masters of chicken recipes. These people know it all!
<a href="#">edit</a>	Default	KR-WKFLW	RecipeMasters	The masters of recipe making. These people know it all!
<a href="#">edit</a>	Default	KR-WKFLW	NotificationAdmin	Notification system admin group for automation.

Export options: [CSV](#) | [spreadsheet](#) | [XML](#)

For information on any of the fields above, please refer to the **Group Maintenance** section below.

## Group Inquiry Screen

This screen provides high-level information about a Group and shows who is assigned to it.

Click a **Group Request** value on the **Action List** screen to display the **Group Inquiry** page for that Group:

### Figure 3.4. Group Inquiry

The screenshot shows the 'Group Inquiry' screen for a group named 'WorkflowAdmin'. It features two main tabs: 'Overview' and 'Assignees'. The 'Overview' tab contains a 'hide' button and a table with the following data:

<b>Group Id:</b>	1	<b>Type Name:</b>	Default
<b>Group Namespace:</b>	KR-WKFLW - Workflow	<b>Group Name:</b>	WorkflowAdmin
<b>Active?:</b>	Yes	<b>Group Description:</b>	WorkflowAdmin

The 'Assignees' tab also has a 'hide' button and shows 'Viewing rows 1 to 2'. Below this is a table titled 'Members:' with the following data:

	* Type Code	* Member Identifier	Namespace	Name	Full Name	Active From Dt	Active To Dt
1	Principal	admin		admin	admin_admin		
2	Principal	notsys		notsys	Notification System		

Buttons for 'expand all', 'collapse all', and 'close' are also visible on the screen.

## Overview Tab

For information on these fields please refer to the **Group Maintenance** section that follows.

## Assignees Tab

Table 3.1. Group Inquiry: Assignees Attributes

Field	Description
Type Code	The code for the type of this Group member
Member Identifier	The user ID for the Member
Name	The user name for the Member
Active From Dt	The effective date of membership in this Group for this member; if it is blank, membership is valid as soon as the member is assigned This field is useful when you want to add a Person, Role, or Group

Field	Description
	to a Group before he/she/they should be active in the Group. This might happen, for example, when you add a new employee before their start date.
Active To Dt	The termination date of membership for this member; if blank, the membership does not terminate

## Group Maintenance Document

The Group document allows you to associate Persons, Roles, or other Groups with each other so you can assign the same role to all Group members.

Groups have no inherent permissions or responsibilities of their own. Only by associating a Group with a role do the members of that Group become associated with permissions and responsibilities.

## Creating New Groups

If you are creating a new Group and have clicked the **Create New** button on the Group Lookup screen, you will be redirected to the Group creation page. One of the first things to do, if wanting to create a group with a non-default Group Type, is to use the kim type lookup to select a different group type. Once you click on the Type Name in the Overview tab, you will be redirected to a screen like the following:

**Figure 3.5. KIM Type Lookup**

**Table 3.2. KIM Type Lookup Search Attributes**

Field	Description
Namespace Code	Optional. Select the code for the application and module to which this KIM Type pertains.
Type Name	Optional. Enter the name for this KIM Type.
Type Identifier	Optional. Enter the unique system-assigned Identifier for this KIM Type.
Active Indicator	Required (defaults to Yes). Change the default selection to view KIM types that are inactive or to see all Types (active and inactive).

**The search results list includes the same fields as the Lookup screen.** The search results are displayed below the search fields on the Lookup screen.

To select the Type you want to use for your new Group, click the return value link for that Type. KIM will copy the Type information to use in creating your new Group.

## Group Maintenance Document: Layout

The Group document includes these tabs:

- Document Overview

- Overview
- Assignees
- Ad Hoc Recipients
- Route Log

The Overview and Assignees tabs are described below. Information on the Ad Hoc Recipients and Route Log tabs can be found in the [Frequently Used Tabs](#) section of the Introduction to Rice Guide.

**Figure 3.6. Group Maintenance Document**

The screenshot shows the 'Group Maintenance Document' interface. At the top right, there is a header with 'Doc Nbr: 3063', 'Status: INITIATED', 'Initiator: admin', and 'Created: 01:30 PM 05/17/2013'. Below this is a 'Document Overview' tab with a 'hide' button. The 'Document Overview' section contains fields for '\* Description:', 'Organization Document Number:', and 'Explanation:'. Below that is the 'Overview' tab with a 'hide' button, containing fields for 'Group Id: 10046', '\* Group Namespace:', 'Active?: ', 'Type Name: Default', '\* Group Name:', and 'Group Description:'. The 'Assignees' tab is also visible, with a 'hide' button and a message: 'Select a Group Namespace to determine if you have the appropriate permission to add members to this group'. Below this is the 'Add Member:' section with a table for adding members, including columns for 'Type Code', 'Member Identifier', 'Name', 'Active From Dt', 'Active To Dt', and 'Actions'. At the bottom, there are 'Ad Hoc Recipients' and 'Route Log' tabs with 'show' buttons. At the very bottom, there are buttons for 'submit', 'save', 'blanket approve', 'close', and 'cancel'.

## Overview Tab

This tab identifies the Group with a unique system-assigned ID number, a namespace, and a name. Each Group also has a Type that specifies any qualifiers that this Group might require.

**Figure 3.7. Group Maintenance Document: Group Overview**

The screenshot shows the 'Overview' tab of the Group Maintenance Document. It contains the following fields: 'Group Id: 10046', '\* Group Namespace:' (with a dropdown menu), 'Active?: ', 'Type Name: Default', '\* Group Name:', and 'Group Description:'. There is a 'hide' button at the top right of the tab.

**Table 3.3. Group Maintenance Document: Group Overview Attributes**

Field	Description
Group ID	Display-only. The unique system-assigned ID number that identifies this Group; Rice completes this field when you submit the document.
Type Name	Required. The type of attributes that are associated with this Group. Some Group types, such as the Default Type, do not require attributes. <b>Note:</b> When creating a new Group, you must select the Type before Rice can generate the document. See the Creating New Groups section of this document
Group Namespace	Required. An indicator that associates the Group with a particular application and module
Group Name	Required. The common descriptive name by which this Group is known

## Group

Field	Description
Active	Check this box to indicate that this Group is active and is a valid choice for assigning to roles. Uncheck the box to indicate that this Group is inactive (it is no longer valid when making role assignments).
Group Description	Optional. Some text to describe the group.

## Assignees Tab

This tab displays the members who belong to this Group. You can also use this tab to add new members or to edit the fields for existing members.

Note that for members not added to the group on this maintenance document, the "delete" button to remove that member is inactive. If a member's association with a group is past, users should set the "Active To Date" field to the end of the membership; the member will not be an active member of the group after that date.

**Figure 3.8. Group Maintenance Document: Assignees Tab**

**Table 3.4. Group Maintenance Document: Assignees Tab Attributes**

Field Name	Description
Type Code	Required. Select the Type of the member you are adding to this Group. Group members can be Principals (as defined on the Person document), Roles, or other Groups.
Member Identifier	Required. Enter the ID that identifies the member you are adding, or use the lookup to search for and select a valid Member ID. The lookup directs you to the <b>Principal</b> , <b>Group</b> , or <b>Role</b> lookup, based on your <b>Member Type Code</b> selection.
Name	Optional. Enter the person name. Clicking on the magnifying glass icon brings up the Person Lookup screen, where a person may be looked up and, once located, inserted.
Active From Date	Optional. To specify the earliest date on which this member is to be considered a valid member of this Group, enter a date in this field.
Active To Date	Optional. To specify a date on which this member is no longer to be considered a valid member of this Group, enter a date in this field. As of this date, the Member will no longer be considered a member of this Group.  <b>Note:</b> You cannot delete or inactivate Group members. To remove a member from a Group, enter an Active To Date.
Actions	Click the <b>Add</b> button to add this member to the Group.

---

# Chapter 4. Role

## Role Lookup Screen

This screen provides detailed information relating to the Permissions assigned to a **Role** in KIM.

The **Role Lookup** screen is accessible directly off the Rice **Administration** menu by clicking **Role** in the Identity section.

Click the **Granted to Roles** field on the **Document Configuration** screen to display the **Role Lookup** screen. You can also access the Role Lookup screen by clicking a **Granted to Roles** value in any list where it appears.

**Figure 4.1. Role Lookup**

The screenshot shows the 'Role Lookup' interface. At the top left is the title 'Role Lookup' with a help icon. At the top right is a 'create new' button and a note '\* required field'. The main area contains a table of search criteria:

Role:	<input type="text"/>
Role Name:	<input type="text"/>
Type:	<input type="text"/>
Namespace:	<input type="text"/>
Principal Name:	<input type="text"/>
Group Namespace:	<input type="text"/>
Group Name:	<input type="text"/>
Permission Namespace:	<input type="text"/>
Permission Name:	<input type="text"/>
Permission Template Namespace:	<input type="text"/>
Permission Template Name:	<input type="text"/>
Responsibility Namespace:	<input type="text"/>
Responsibility Name:	<input type="text"/>
Responsibility Template Namespace:	<input type="text"/>
Responsibility Template Name:	<input type="text"/>
Active?:	<input checked="" type="radio"/> Yes <input type="radio"/> No <input type="radio"/> Both

At the bottom of the form are three buttons: 'search', 'clear', and 'cancel'.

## Document Layout

This screen may have three or four tabs of information, depending on the Role. The first three tabs are **Overview**, **Permissions**, and **Responsibilities**. KIM only displays the fourth tab, **Assignees**, when Assignee information is associated with this **Role**.



**Figure 4.2. Role Maintenance Document**

The screenshot shows a web-based interface for role maintenance. At the top, there are fields for 'Doc Nbr: 3064', 'Status: INITIATED', 'Initiator: admin', and 'Created: 01:58 PM 05/17/2013'. Below this is a 'Role' field with a question mark icon. The main interface is divided into several tabs: 'Document Overview', 'Overview', 'Permissions', 'Responsibilities', 'Ad Hoc Recipients', and 'Route Log'. The 'Permissions' tab is currently active, displaying a table of permissions with columns for Permission Namespace, Permission Identifier, Permission Name, Permission Detail Values, Active Indicator, and Actions. The table contains 11 rows of data. At the bottom of the interface, there are buttons for 'submit', 'save', 'blanket approve', 'close', and 'cancel'.

Information on the fields in all of these tabs can be found in the **Role Maintenance** section of this user guide.

## Role Maintenance Document

The Role Maintenance Document allows you to create a new KIM Role and edit existing Roles. Each Role collects a specific set of Permissions and Responsibilities and allows you to assign members to it.

The purpose of each Role is defined by its associated Permissions and Responsibilities. Roles are classified by **Types** that generally indicate with which Permissions and Responsibilities they are associated.

### Note

The process of creating a new Type for Roles requires technical assistance.

To access the **Role** screen:

1. Do a **Role Lookup**.
2. Find the Role you want to change in the list of search results.
3. Click **Edit** for the row that has the Role that you need to change.

## Role Maintenance Document

The Role document includes six tabs:

- Document Overview
- Overview
- Permissions
- Responsibilities
- Ad Hoc Recipients
- Route Log

Those that contain unique information are addressed below. For more information about the Document Overview, Ad Hoc Recipients and Route Log tabs, please see the [Frequently Used Tabs](#) section of the Introduction to Rice Guide.

**Figure 4.3. Role Maintenance Document: Tabs**

The screenshot shows a web interface for Role Maintenance. At the top, there is a header bar with the following information: **Doc Nbr:** 3064, **Status:** INITIATED, **Initiator:** admin, and **Created:** 01:58 PM 05/17/2013. Below this, there are two buttons: "expand all" and "collapse all", with a note "\* required field". The main area contains six tabs: "Document Overview", "Overview", "Permissions", "Responsibilities", "Ad Hoc Recipients", and "Route Log". Each tab has a "show" button next to it. At the bottom of the interface, there are five buttons: "submit", "save", "blanket approve", "close", and "cancel".

## Overview Tab

This tab gives the Role a unique system-assigned ID number, a Namespace, and a Name. Each Role also has a Type that usually matches the kinds of Permissions and Responsibilities associated with it.

**Figure 4.4. Role Maintenance Document: Overview Tab**

The screenshot shows the "Overview" tab of the Role Maintenance Document. It contains the following fields: **Role:** 1, **Type Name:** Derived Role: Principal, **\* Namespace:** KUALI - Kual Systems, **\* Role Name:** User, and **Active?:** . The **Description:** field contains the text: "This role derives its members from the users in the Principal table. This role gives users high-level permissions to". There is a "hide" button in the top right corner of the tab.

**Table 4.1. Role Maintenance Document: Overview Attributes**

Field	Description
Role	Display-only. The unique, system-assigned ID number that identifies this Role
Type Name	Display-only. The Type Name usually identifies the general kinds of Permissions and Responsibilities associated with this Role. <b>Note:</b> When creating a new Role, you must select its Type before you can display a new Role document. See Creating New Roles below.

## Role

Field	Description
Namespace	Required. An indicator that associates the Role with a particular application and module
Role Name	Required. The common descriptive name by which this Role is known
Active	Check this box to indicate that this Role is Active and is, therefore, to be included by KIM when it evaluates Permissions and Responsibilities. Uncheck the Active box to indicate that this Role is inactive.
Description	Optional. Text description of the role.

## Creating New Roles

You must search for and select an existing Type for KIM to generate a new Role document. When you click the Create New button, KIM displays the **KIM Type Lookup** screen:

**Figure 4.5. KIM Type Lookup**

### Note

While you use the KIM Type Lookup screen both for creating new Groups and new Roles, not all KIM Types are valid for both Groups and Roles. When using this Lookup, you may see different results depending on the KIM Types that are appropriate for your task.

## Permissions Tab

This tab identifies the Permissions associated with this Role. Permissions authorize specific actions in the system with which they are associated. A Role can have any number of Permissions (including no Permissions) associated with it.

**Figure 4.6. Role Maintenance Document: Permissions Tab**

* Permission Namespace	Permission Identifier	Permission Name	Permission Detail Values	Active Indicator	Actions
1 KR-SYS - Enterprise Infrastructure	146	Ad Hoc Review RICE Document	documentTypeName=RiceDocument	<input checked="" type="checkbox"/>	<a href="#">delete</a>
2 KR-SYS - Enterprise Infrastructure	149	Initiate RICE Document	documentTypeName=RiceDocument	<input checked="" type="checkbox"/>	<a href="#">delete</a>
3 KR-SYS - Enterprise Infrastructure	156	Copy RICE Document	documentTypeName=RiceDocument	<input checked="" type="checkbox"/>	<a href="#">delete</a>
4 KR-SYS - Enterprise Infrastructure	161	Inquire Into RICE Records	namespaceCode=KR*	<input checked="" type="checkbox"/>	<a href="#">delete</a>
5 KR-SYS - Enterprise Infrastructure	162	Look Up RICE Records	namespaceCode=KR*	<input checked="" type="checkbox"/>	<a href="#">delete</a>
6 KR-SYS - Enterprise Infrastructure	165	Open RICE Document	documentTypeName=RiceDocument	<input checked="" type="checkbox"/>	<a href="#">delete</a>
7 KUALI - KualI Systems	174	Log In KualI Portal		<input checked="" type="checkbox"/>	<a href="#">delete</a>
8 KR-SYS - Enterprise Infrastructure	161	Inquire Into RICE Records	namespaceCode=KR*	<input checked="" type="checkbox"/>	<a href="#">delete</a>
9 KR-SYS - Enterprise Infrastructure	162	Look Up RICE Records	namespaceCode=KR*	<input checked="" type="checkbox"/>	<a href="#">delete</a>
10 KR-SYS - Enterprise Infrastructure	163	Maintain System Parameter	namespaceCode=KR*	<input checked="" type="checkbox"/>	<a href="#">delete</a>
11 KR-SYS - Enterprise Infrastructure	166	Use all RICE Screen	namespaceCode=KR*	<input checked="" type="checkbox"/>	<a href="#">delete</a>

**Table 4.2. Role Maintenance Document: Permissions Attributes**

Field Name	Description
Add Permission ID	To add a Permission to this Role, enter the appropriate Permission ID, or search for and select a value using the <b>Permission</b> lookup
Add	Click the <b>Add</b> button to add the selected Permission to this Role document.

After you add a Permission to the document, KIM displays additional information about the Permission:

**Note**

You *cannot* edit Permissions from the **Role Maintenance Document**.

**Figure 4.7. Role Maintenance Document: Permissions Tab, Add Permissions**

	* Permission Namespace	Permission Identifier	Permission Name	Permission Detail Values	Active Indicator	Actions
1	KR-SYS - Enterprise Infrastructure	146	Ad Hoc Review RICE Document	documentTypeName=RiceDocument	<input checked="" type="checkbox"/>	<input type="button" value="delete"/>
2	KR-SYS - Enterprise Infrastructure	149	Initiate RICE Document	documentTypeName=RiceDocument	<input checked="" type="checkbox"/>	<input type="button" value="delete"/>
3	KR-SYS - Enterprise Infrastructure	156	Copy RICE Document	documentTypeName=RiceDocument	<input checked="" type="checkbox"/>	<input type="button" value="delete"/>
4	KR-SYS - Enterprise Infrastructure	161	Inquire Into RICE Records	namespaceCode=KR*	<input checked="" type="checkbox"/>	<input type="button" value="delete"/>
5	KR-SYS - Enterprise Infrastructure	162	Look Up RICE Records	namespaceCode=KR*	<input checked="" type="checkbox"/>	<input type="button" value="delete"/>
6	KR-SYS - Enterprise Infrastructure	165	Open RICE Document	documentTypeName=RiceDocument	<input checked="" type="checkbox"/>	<input type="button" value="delete"/>
7	KUALI - KualI Systems	174	Log In KualI Portal		<input checked="" type="checkbox"/>	<input type="button" value="delete"/>
8	KR-SYS - Enterprise Infrastructure	161	Inquire Into RICE Records	namespaceCode=KR*	<input checked="" type="checkbox"/>	<input type="button" value="delete"/>
9	KR-SYS - Enterprise Infrastructure	162	Look Up RICE Records	namespaceCode=KR*	<input checked="" type="checkbox"/>	<input type="button" value="delete"/>
10	KR-SYS - Enterprise Infrastructure	163	Maintain System Parameter	namespaceCode=KR*	<input checked="" type="checkbox"/>	<input type="button" value="delete"/>
11	KR-SYS - Enterprise Infrastructure	166	Use all RICE Screen	namespaceCode=KR*	<input checked="" type="checkbox"/>	<input type="button" value="delete"/>
12	KR-WKFLW - Workflow	140	Use Document Operation Screen	actionClass=org.kuali.rice.kew.documentoperation.web.DocumentOperationAction	Yes	<input type="button" value="delete"/>

**Table 4.3. Role Maintenance Document: Permissions Tab, Add Attributes**

Field	Description
Permission Namespace	Display-only. The Namespace identifies the application and module associated with this Permission.
Permission Identifier	Display-only. The unique system-assigned ID number for this Permission
Permission Name	Display-only. The descriptive name of this Permission. This often tells you, in general terms, what the Permission authorizes.
Permission Detail Values	Display-only. The document Types, tabs, and/or fields that this Permission authorizes the holder to see or use. Not all Permissions have Detail Values.
Active Indicator	Display-only. Indicator showing whether this Permission is active in KIM
Actions	Click the <b>Delete</b> button to remove this Permission from this Role.  <b>Warning:</b> You may delete a Permission from a Role only if it has not yet been saved. (In other words, you can delete a Permission from a Role only if you added it to this Role, but have not yet submitted the document.)

**Responsibilities Tab**

This tab identifies the Responsibilities associated with this Role. Responsibilities define the workflow actions that will be requested of the Role. A Role can have any number of Responsibilities (including none) associated with it.

**Figure 4.8. Role Maintenance Document: Responsibilities Tab**



**Table 4.4. Role Maintenance Document: Responsibility Attributes**

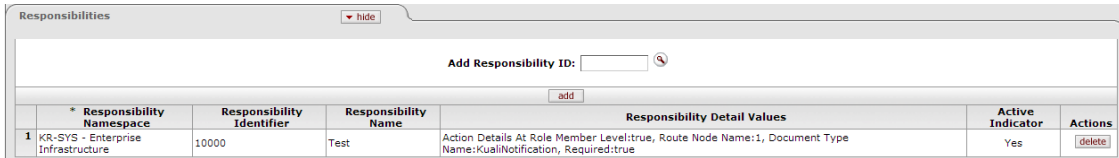
Field Name	Description
Add Responsibility ID	To add a Responsibility to this Role, enter the Responsibility ID, or search for and select a value using the Responsibility lookup .
Add button	Click the <b>Add</b> button to add the selected Responsibility to this Role document.

After you add a Responsibility to the document, KIM displays additional information about this Responsibility.

**Note**

In general, you cannot edit Responsibilities using the Role document, but some Responsibilities have associated attributes that you must define at the Role level.

**Figure 4.9. Role Maintenance Document: Responsibility, Added Responsibility**



**Table 4.5. Role Maintenance Document: Responsibility, Add Attributes**

Field Name	Description
Responsibility Namespace	Display-only. The Namespace tells you the application and module associated with this Responsibility.
Responsibility Identifier	Display-only. The unique system-assigned ID number for this Responsibility
Responsibility Name	Display-only. The descriptive name of this Responsibility. For most Responsibilities, the name is <b>Review</b> .
Responsibility Detail Values	<p>Display-only. This gives you more specific information about the Responsibility. Responsibility Detail Values are formatted in a standard way with these attributes, separated by commas:</p> <ul style="list-style-type: none"> <li>• <b>Route Node:</b> The location where this Responsibility is invoked in Workflow.</li> <li>• <b>Document Type:</b> The Document Type for which this Responsibility generates workflow requests.</li> <li>• <b>Action Details at Role Member Level:</b> When this field is True, action details are kept for each Member of this Role. When this field is False, action details are only kept at the Responsibility level for this Role. (see Assigning Action Detail Values, below).</li> <li>• <b>Required:</b> Indicates if the routing represented by this Responsibility should be required. If this is set to True and the Responsibility fails to generate an Action Request (perhaps because no one is assigned to the associated Role), then the document will go into Exception status. If this routing is optional, this will be False and the document will simply skip this Responsibility if no requests are generated.</li> </ul>
Active Indicator	Display-only. Indicator showing whether this Responsibility is active within the system
Actions	<p>Click the Delete button to remove this Responsibility from this Role.</p> <p><b>Warning:</b> You can delete a Responsibility only if it has not yet been saved to the database (in other words, you can only delete it if you have added it to this Role but have not yet submitted the document).</p>

## Assigning Action Detail Values

When you add a Responsibility with an **Action Detail Values at Role Member Level** value of **False**, you must complete additional fields in a Responsibility Action sub-section. KIM automatically displays this sub-section immediately beneath the Responsibility you've just added.

The fields in this sub-section define the type of Action Requests generated for, and the general workflow behavior associated with, this Responsibility. Entries in these fields cause KIM to generate the same Type of Action Requests for all members of this Role and handle actions by all members of this Role in the same way.

**Figure 4.10. Role Maintenance Document: Responsibility Tab, Action Section**

Responsibility Action					
Name	* Action Type Code	Priority Number	* Action Policy Code	Force Action	
All	<input type="text" value=""/>	<input type="text" value=""/>	<input type="text" value=""/>	<input type="checkbox"/>	

**Table 4.6. Role Maintenance Document: Responsibility Tab, Action Section Attributes**

Field Name	Description
Name	Display-only. The namespace and name of the Responsibility associated with these action details
Action Type Code	Required. The Type of Action Request that KIM generates for this Responsibility. Action Type Codes are: Approve, FYI, and Acknowledge.
Priority Number	Optional. If multiple requests are generated at the route node specified on this Responsibility, this determines the order in which KIM generates these requests. KIM processes requests with lower priority numbers before processing requests with higher numbers. Requests with no number are treated as a priority 1.
Action Policy Code	Required. This determines what happens if multiple members of this Role receive the same Action Request and one of them takes the action. For example, if a Role has a Group with three members assigned, all of these members receive the Action Request defined here; this code then determines what KIM does when one of them takes action on the document. A value of <i>FIRST</i> indicates that the first Group member to take action on the document causes KIM to clear all the requests for this Responsibility that may be in other Group member's action lists. A value of <i>ALL</i> indicates that each Group member must take individual action to clear his or her own requests.
Force Action	Check this box to indicate that each user must take action for this request, even if the user has previously taken action on this document. Leaving the box unchecked allows a request to be immediately fulfilled if the Role member has previously taken action on this specific document.

# Chapter 5. KIM Type

## KIM Type Lookup

Use this lookup when you create a new group in KIM to find the KIM Type for the new group.

**Figure 5.1. KIM Type Lookup**

You may enter information in one or more of the search fields to find the correct Type for your new group:

**Table 5.1. KIM Type Lookup Attributes**

Field Name	Description
Namespace Code	Optional. Select the code of the application and module for this KIM Type.
Type Name	Optional. Enter the name for this KIM Type.
Type Identifier	Optional. Enter the unique system-assigned Identifier for this KIM Type.
Active Indicator	Required. Defaults to Yes (Yes means display only Active Types). Change the default selection to view KIM Types that are Inactive or Types that are both Active and Inactive.

**Figure 5.2. KIM Type Lookup: Results Example**

One item retrieved.

Return Value	Namespace Code	Type Name	Type Identifier	Active Indicator
<a href="#">return_value</a>	KUALI	Default	<u>1</u>	Yes

Export options: [CSV](#) | [spreadsheet](#) | [XML](#)

The search results list for this screen has the same fields as the **Lookup** screen. To select the Type you want to use for your new group, click the **return value** link for it in the search results list.

# KIM Type Inquiry

The KIM Type Inquiry screen provides details about a KIM Type data element.

To access this screen, from the **Kim Type Lookup** screen, click a **Type Name** in your search results list.

**Figure 5.3. KIM Type Inquiry**

Kim Type	
Type Identifier:	1
Type Name:	Default
Service Name:	
Active Indicator:	Yes
Namespace Code:	KUALI - KualI Systems

All fields are defined above in *KIM Type Lookup* except these:

**Table 5.2. KIM Type Inquiry Attributes**

Field Name	Description
Service Name	Display-only. The name of the service associated with this KIM Type
Namespace Code	Display-only. The namespace code associated with the selected KIM Type

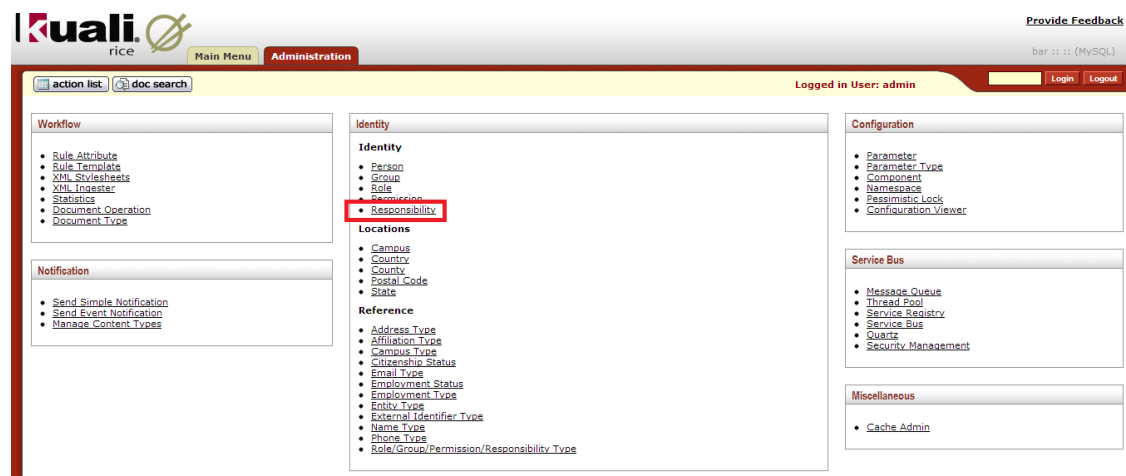


# Chapter 6. Responsibility

## Responsibility Lookup

Similar to the Permission Lookup, you can use the Responsibility Lookup to search for and view existing responsibilities in KIM. You can view summarized information about the responsibility detail values as well as the roles with which the responsibility is currently associated.

Figure 6.1. Identity Channel: Responsibility Link



To display this screen, from the **Administration** menu, click **Responsibility** in the Identify section of the menu.

### Note

This table is display-only. Technical assistance is required to modify responsibilities.

**Figure 6.2. Responsibility Lookup**

**Responsibility Lookup** ?
create new

\* required field

<b>Template Namespace:</b>	<input type="text"/>
<b>Template Name:</b>	<input style="border: 2px solid orange;" type="text"/>
<b>Responsibility Namespace:</b>	<input type="text"/>
<b>Responsibility Name:</b>	<input type="text"/>
<b>Role Namespace:</b>	<input type="text"/>
<b>Role Name:</b>	<input type="text"/> <span style="font-size: 0.8em;">🔍 📖</span>
<b>Principal Name:</b>	<input type="text"/> <span style="font-size: 0.8em;">🔍 📖</span>
<b>Group Namespace:</b>	<input type="text"/>
<b>Group Name:</b>	<input type="text"/> <span style="font-size: 0.8em;">🔍 📖</span>
<b>Attribute Value:</b>	<input type="text"/>
<b>Active Indicator:</b>	<input checked="" type="radio"/> Yes <input type="radio"/> No <input type="radio"/> Both
<span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 10px; margin: 0 5px;">search</span> <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 10px; margin: 0 5px;">clear</span> <span style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 10px; margin: 0 5px;">cancel</span>	

To find information about a Responsibility, enter information in one or more of the fields on the Lookup page and then click the Search button.

**Table 6.1. Responsibility Lookup Attributes**

Field Name	Description
Template Namespace	Optional. The code identifying the application and module the template pertains to. Because responsibilities pertain to workflow, most responsibility templates are associated with the KR-WKFLW (Kuali Rice-Workflow) namespace.
Template Name	Optional. The template the responsibility is based on. A template usually defines, in a broad sense, what the responsibility is. Since responsibilities normally are associated with action requests for user review, most responsibilities have a template name of "Review."
Responsibility Namespace	Optional. The code designating the application and module this responsibility is associated with. This code usually corresponds to the namespace of the document type for which the responsibility generates action requests.
Responsibility Name	Optional. The name of this responsibility. In most cases, the responsibility name is the same as the associated template name ("Review"). Like permission names, responsibility names are not unique.
Role Namespace	An indicator that associates the Role with a particular application and module. To search for a responsibility based on the namespace of the role to which it is assigned, enter the name of that namespace.
Role Name	Optional. The name by which a Role is known in the system. To search for a responsibility based on the Role to which it is assigned, enter that Role name.
Principal Name	Optional. One of the principals that currently have this responsibility through their association with a role
Group Namespace	Optional. The namespace of groups that have this responsibility through the group's association with a role
Group Name	Optional. The name of a group that has this responsibility through its association with a role
Attribute Value	Optional. A specific responsibility detail value associated with a responsibility
Active Indicator	Required (defaults to Yes). Change the default selection to view responsibilities that are inactive or to see all (active and inactive).

### Figure 6.3. Responsibility Lookup: Results

3 items retrieved, displaying all items.

Actions	Template Namespace	Template Name	Responsibility Namespace	Responsibility Name	Responsibility Detail Values	Granted to Roles
<a href="#">edit</a> <a href="#">copy</a>	KR-WKFLW	Review	KR-SYS	Test	Action Details At Role Member Level:true, Route Node Name:1, Document ...	
<a href="#">edit</a> <a href="#">copy</a>	KR-WKFLW	Review	KR-SYS	Test1	Action Details At Role Member Level:true, Route Node Name:2, Document ...	
<a href="#">edit</a> <a href="#">copy</a>	KR-WKFLW	Review	KR-SYS	Test3	Action Details At Role Member Level:false, Route Node Name:3, Document...	

Table 6.2. Responsibility Lookup: Results Attributes

Field Name	Description
Responsibility Detail Values	<p>Display-only. Detailed information that defines:</p> <ul style="list-style-type: none"> <li>• What document this responsibility generates action requests for</li> <li>• When the requests are generated</li> <li>• How the requests are handled by workflow</li> </ul> <p>Unlike permissions, which sometimes have different detail values, responsibility detail values generally contain these elements:</p> <ul style="list-style-type: none"> <li>• <b>routeNodeName:</b> The point in a document's workflow routing at which this responsibility generates requests.</li> <li>• <b>documentTypeName:</b> The name of the document type for which this responsibility generates action requests. This value may also be a parent document type, which indicates that this responsibility applies to all child documents.</li> <li>• <b>actionDetailsAtRoleMemberLevel:</b> A True or False indicator that defines where the system collects details of this workflow action request. If the value is True, the system collects action details when members are assigned to the role. If the value is False, the system collects action details when this responsibility is assigned to a role.</li> <li>• <b>Required:</b> A True or False value that indicates whether the system is required to generate an action request for this document type. If the value is True and the document generates no requests associated with this responsibility, then the document will go into exception status. If the value is False and the responsibility generates no action requests, then the document continues to route as normal.</li> </ul>
Granted to Roles	<p><b>Display-only.</b> Lists the namespace and name of roles that have this responsibility. Click a linked name to view the Role Inquiry for that role name.</p>

## Responsibility Inquiry

To view the **Responsibility Inquiry** screen for a responsibility, click its **Responsibility Name** in the search results list displayed when you do a Responsibility Lookup. The Responsibility Inquiry screen contains the same information displayed in the search results, but in a slightly different format:

Figure 6.4. Responsibility Inquiry

The screenshot shows a web application window titled "Responsibility Inquiry". At the top right, there are two buttons: "expand all" and "collapse all". The main content is organized into several sections, each with a "hide" button:

- Responsibility Attributes**:
  - Template Namespace: KR-WKFLW - Workflow
  - Template Name: Review
  - Responsibility Namespace: KR-SYS - Enterprise Infrastructure
  - Responsibility Name: Test
- Responsibility Detail Values**:
  - Detail Values ( actionDetailsAtRoleMemberLevel - true )**: Attribute Name: actionDetailsAtRoleMemberLevel, Attribute Value: true
  - Detail Values ( routeNodeName - 1 )**: Attribute Name: routeNodeName, Attribute Value: 1
  - Detail Values ( documentTypeName - KualNotification )**: Attribute Name: documentTypeName, Attribute Value: KualNotification
  - Detail Values ( required - true )**: Attribute Name: required, Attribute Value: true
- Responsibility Assigned Roles**: (This section is currently empty)

At the bottom center of the window, there is a "close" button.

The fields on this screen are documented in the Responsibility Lookup section above.

# Chapter 7. Permission

## Permission Lookup

The Permission Lookup screen allows you to search for and view existing permissions. It displays summarized information about the permission detail values as well as the roles that are currently associated with this permission.

### Note

This table is display-only. Technical assistance is required to modify permissions.

You get to this screen by clicking **Permissions** in the **Identity** section of the **Administration** menu.

**Figure 7.1. Permission Lookup**

Permission Lookup create new

\* required field

Template Namespace:	<input type="text"/>
Template Name:	<input type="text"/>
Permission Namespace:	<input type="text"/>
Permission Name:	<input type="text"/>
Role Namespace:	<input type="text"/>
Role Name:	<input type="text"/>
Principal Name:	<input type="text"/>
Group Namespace:	<input type="text"/>
Group Name:	<input type="text"/>
Attribute Value:	<input type="text"/>
Template Id:	<input type="text"/>
Active Indicator:	<input checked="" type="radio"/> Yes <input type="radio"/> No <input type="radio"/> Both
<input type="button" value="search"/> <input type="button" value="clear"/> <input type="button" value="cancel"/>	

Enter information in one or more fields on the **Permission Lookup** screen and then click the **search** button to display permissions that match your information.

**Table 7.1. Permission Lookup Attributes**

Field Name	Description
Template Namespace	Optional. The code identifying the application and module the template pertains to. Because templates tend to be general categories, they are often associated with system-level namespaces.
Template Name	Optional. The template the permission is based on. A template usually defines, in a broad sense, what the permission controls. Similar types of permissions use the same template.
Permission Namespace	Optional. The code designating the application and module this permission is associated with.

## Permission

Field Name	Description
Permission Name	Optional. The name of this permission. In most cases, the permission name is the same as its associated template name.
Role Namespace	Optional. An indicator that associates the role with a particular application and module.
Role Name	Optional. The common descriptive name by which this role is known.
Principal Name	Optional. The principals that currently have this permission through their association with a role
Group Namespace	Optional. The namespace of groups that have this permission through the groups' association with a role
Group Name	Optional. The name of a group that has this permission through its association with a role
Permission Detail Values	Optional. Detailed information that, in combination with the permission name, defines the permission's function. For example, if the permission name is <b>Initiate Document</b> , the <b>Permission Detail Values</b> field indicates the specific type of document the initiate permission pertains to. Permission detail values can include many different types of data. Some common permission details: <ul style="list-style-type: none"> <li>• <b>documentTypeName</b>: The name of the document Type associated with this permission.</li> <li>• <b>routeNodeName</b>: The point in a document's workflow routing at which this permission becomes relevant.</li> <li>• <b>routeStatusCode</b>: The routing status that a document must be in for this permission to apply.</li> <li>• <b>propertyName</b>: A field or document element that the permission pertains to.</li> </ul>

When you click the **search** button for a Permission Lookup, KIM displays your search results in a table like this:

**Figure 7.2. Permission Lookup: Results Example**

Actions	Template Namespace	Template Name	Permission Namespace	Permission Name	Permission Detail Values	Granted to Roles
<a href="#">edit</a> <a href="#">copy</a>	KR-NS	Use_Screen	KR-WKFLW	Use Document Operation Screen	Action Class:org.kuali.rice.kew.documentoperation.web.DocumentOperatio...	<a href="#">KR-SYS Technical Administrator</a> , <a href="#">KR-SYS Technical Administrator</a>
<a href="#">edit</a> <a href="#">copy</a>	KR-NS	Use_Screen	KR-BUS	Use Java Security Management Screen	Action Class:org.kuali.rice.ksb.security.admin.web.JavaSecurityManagem...	<a href="#">KR-SYS Technical Administrator</a> , <a href="#">KR-SYS Technical Administrator</a>
<a href="#">edit</a> <a href="#">copy</a>	KR-NS	Use_Screen	KR-BUS	Use Message Queue Screen	Action Class:org.kuali.rice.ksb.messaging.web.MessageQueueAction	<a href="#">KR-SYS Technical Administrator</a> , <a href="#">KR-SYS Technical Administrator</a>
<a href="#">edit</a> <a href="#">copy</a>	KR-NS	Use_Screen	KR-BUS	Use Service Registry Screen	Action Class:org.kuali.rice.ksb.messaging.web.ServiceRegistryAction	<a href="#">KR-SYS Technical Administrator</a> , <a href="#">KR-SYS Technical Administrator</a>

The information in the search results table is display-only and is defined above. New field:

**Table 7.2. Permission Lookup: Results Attributes**

Field Name	Description
Granted to Roles	Lists the namespace and name of roles that have this permission. Click on a linked name to view its Role Inquiry screen.

## Permission Inquiry

To view the **Permission Inquiry** screen for a Permission, click the **Permission Name** in the search results from a Permission Lookup. The Permission Inquiry screen contains the same information as the Permission Lookup search results, but in a slightly different format:

**Figure 7.3. Permission Inquiry**

The screenshot shows a web interface titled "Permission Inquiry" with "expand all" and "collapse all" buttons. It contains three main sections:

- Permission Attributes:** A form with fields for "Template Namespace", "Template Name", "Permission Namespace" (value: KR-WKFLW - Workflow), and "Permission Name" (value: Use Document Operation Screen).
- Permission Detail Values:** A section with a "hide" button and a sub-section "Detail Values ( actionClass - org.kuali.rice.kew.documentoperation.web.DocumentOperationAction )". It contains a table:
 

Attribute Name:	actionClass	Attribute Value:	org.kuali.rice.kew.documentoperation.web.DocumentOperationAction
-----------------	-------------	------------------	--
- Permission Assigned Roles:** A section with a "hide" button and a sub-section "Assigned Roles ( KR-SYS - Enterprise Infrastructure - Technical Administrator - Default )". It contains two identical tables:
 

Namespace:	KR-SYS - Enterprise Infrastructure	Role Name:	Technical Administrator	Role Type Name:	Default
------------	------------------------------------	------------	-------------------------	-----------------	---------

A "close" button is located at the bottom center of the interface.

The fields on this screen are documented above in the **Permission Lookup** section.

## Permission Template Inquiry

This screen provides detailed information about a **Template Namespace**. To display it, click a **Template Name** on the **Document Configuration** screen or the **Permission Inquiry** screen.

**Figure 7.4. Permission Template Inquiry**

The screenshot shows a web interface titled "Permission Template Inquiry" with "expand all" and "collapse all" buttons. It contains one main section:

- Permission Template Attributes:** A section with a "hide" button and a form with fields for "Template Namespace" (value: KR-NS - Kuali Nervous System), "Template Name" (value: Use Screen), and "Permission Template Description".

A "close" button is located at the bottom center of the interface.

Information related to the fields on this screen can be found above, in the **Permission Lookup** section of this document.

## Delivered Permission Templates

Rice is delivered with a number of permission templates. Below is a listing of each along with a brief description of their use.

**Table 7.3. Delivered Permission Templates**

Template ID	Template Namespace	Template Name	Permission Description	Template	Permission Details
1	KUALI	Default			
2	KR-NS	Copy Document			documentTypeName=
3	KR-WKFLW	Administer Routing for Document			documentTypeName=
4	KR-WKFLW	Blanket Approve Document	Authorizes user to bypass specified approval route nodes		documentTypeName=
5	KR-WKFLW	Route Document	Authorizes user to route a document from their action list		documentTypeName=
8	KR-NS	Take Requested Action	Authorizes user to take applicable actions (approve, disapprove, acknowledge, etc.) on documents in their action list.		actionRequestCd=
9	KR-WKFLW	Ad Hoc Review Document	Authorizes user to review a document that has been sent to them via an Ad Hoc Request		documentTypeName=
10	KR-SYS	Initiate Document	Authorizes user to create a new document		documentTypeName=
14	KR-WKFLW	Cancel Document	Authorizes users to cancel a document prior to it being submitted for routing.		documentTypeName= routeNodeName=
15	KR-WKFLW	Save Document	Authorizes user to save a document that has been initiated or routed to them for action back to their action list		documentTypeName=
16	KR-NS	Edit Document			documentTypeName= routeStatusCode= routeNodeName=
23	KR-NS	Look Up Records			namespaceCode= componentName=
24	KR-NS	Inquire Into Records			namespaceCode= componentName=
25	KR-NS	View Inquiry or Maintenance Document Field			
26	KR-NS	Modify Maintenance Document Field			
27	KR-NS	Full Unmask Field			propertyName= componentName=
28	KR-NS	Partial Unmask Field			
29	KR-NS	Use Screen			actionClass= namespaceCode=
30	KR-NS	Perform Custom Maintenance Document Function			
31	KR-NS	Use Transactional Document			
32	KR-NS	Modify Batch Job			namespaceCode=
33	KR-NS	Upload Batch Input File(s)			
34	KR-NS	Maintain System Parameter			namespaceCode=



## Permission

Template ID	Template Namespace	Template Name	Permission Description	Template	Permission Details
35	KR-IDM	Assign Role			namespaceCode=
36	KR-IDM	Grant Permission			namespaceCode=
37	KR-IDM	Grant Responsibility			namespaceCode=
38	KR-IDM	Populate Group			namespaceCode=
40	KR-NS	Open Document			documentTypeName=
42	KR-NS	Create / Maintain Record(s)			documentTypeName=
43	KR-NS	View Inquiry or Maintenance Document Section			
44	KR-NS	Modify Maintenance Document Section			
45	KR-NS	Add Note / Attachment	Authorizes a user to add a note / attachment to a document		documentTypeName=
46	KR-NS	View Note / Attachment	Authorizes a user to view notes and download attachments that have been added to a document		documentTypeName=
47	KR-NS	Delete Note / Attachment	Authorizes a user to remove notes and attachments that have been added to a document		documentTypeName= createdBySelfOnly
49	KR-NS	Send Ad Hoc Request	Authorizes a user to create an action request to user(s) outside the normal route chain		actionRequestCd= documentTypeName=
51	KR-WKFLW	Add Message to Route Log	Authorizes user to add annotations to a document that appear in the Route Log of a document		documentTypeName=
52	KR-RULE	KRMS Agenda Permission	Authorizes user to view and edit KRMS agendas		namespaceCode=
53	KR-KRAD	Open View			
54	KR-KRAD	Edit View			
55	KR-KRAD	Use View			
56	KR-KRAD	View Field			
57	KR-KRAD	Edit Field			
58	KR-KRAD	View Group			
59	KR-KRAD	Edit Group			
60	KR-KRAD	View Widget			
61	KR-KRAD	Edit Widget			
62	KR-KRAD	Perform Action			
63	KR-KRAD	View Line	Authorizes a user to view the lines in a collection		
64	KR-KRAD	Edit Line	Authorizes a user to edit the lines in a collection		
65	KR-KRAD	View Line Field	Authorizes a user to view the individual fields in the lines of a collection		
66	KR-KRAD	Edit Line Field	Authorizes a user to edit the individual fields in the lines of a collection		
67	KR-KRAD	Perform Line Action	Authorizes a user to take actions (add, delete, edit, etc.) to the lines of a collection		
68	KR-WKFLW	Recall Document	Authorizes a user to recall a submitted document to		documentTypeName= routeNodeName=

## Permission

---

Template ID	Template Namespace	Template Name	Permission Description	Template	Permission Details
			their action list for edit and resubmission or cancellation		routeStatusCode= appDocStatus=

---

# Chapter 8. Terminology

## Principal

A principal represents an entity that can authenticate. In essence, you can think of a principal as an "account" or as an entity's authentication credentials. A principal has an ID that is used to uniquely identify it. It also has a name that represents the principal's username and is typically what is entered when authenticating. All principals are associated with one and only one entity.

## Entity

An entity represents a person or system. Additionally, other "types" of entities can be defined in KIM. Information like name, phone number, etc. is associated with an entity. While an entity will typically have a single principal associated with it, it is possible for an entity to have more than one principal or even no principals at all (in the case where the entity does not actually authenticate).

Entities have numerous attributes associated with them, including:

- Names
- Addresses
- Phone Numbers
- Email Addresses
- Entity Type
- Affiliations
- Employment Information
- External Identifiers
- Privacy Preferences

## Group

A group is a collection of principals. You can create a group using both direct principal assignment and nested group membership. All groups are uniquely identified by a namespace code plus a name. A principal or group is a "member" of a group if it is either directly assigned to the group or indirectly assigned (through a nested group membership). A principal or group is a "direct" member of another group only if it is directly assigned as a member of the group, and not through a nested group assignment.

## Permission

A permission is the ability to perform an action. All permissions have a permission template. Both permissions and permission templates are uniquely identified by a namespace code plus a name. The permission template defines the coarse-grained permission and specifies what additional permission details need to be collected on permissions that use that template. For example, a permission template might have a name of "Initiate Document," which requires a permission detail specifying the document type that can

be initiated. A permission created from the "Initiate Document" template would define the name of the specific Document Type that can be initiated as a permission detail.

The **isAuthorized** and **isAuthorizedByTemplateName** operations on the **PermissionService** are used to execute authorization checks for a principal against a permission. Permissions are always assigned to roles (never directly to a principal or group). A particular principal will be authorized for a given permission if the principal is assigned to a role that has been granted the permission.

## Responsibility

A responsibility represents an action that a principal is requested to take. This is used for defining workflow actions (such as approve, acknowledge, FYI) for which the principal is responsible. Responsibilities form the basis of the workflow engine routing process.

A responsibility is very similar to a permission in a couple of ways. First, responsibilities are always granted to a role, never assigned directly to a principal or group. Furthermore, similar to permissions, a role has a responsibility template. The responsibility template specifies what additional responsibility details need to be defined when the responsibility is created.

## Role

You grant permissions and responsibilities to roles. Roles have a membership consisting of principals, groups, and/or other roles. As a member of a role, the associated principal has all permissions and responsibilities that have been granted to that role.

You can specify a qualification to any membership assignment on the role, which is extra information about that particular member of the role. For example, a person may have the role of "Dean" but that can be further qualified by the school they are the dean of, such as "Computer Science." You can pass qualifications as part of authorization checks to restrict the subset of roles to check.

## Reference Information

There are several collections of reference information managed within KIM:

- Address type
- Affiliation type
- Citizenship status
- Email type
- Employment status
- Employment type
- Entity name type
- Entity type
- External identifier type
- Phone number type

---

## Configuration Parameters

**Table 8.1. KIM Configuration Parameters**

Configuration Parameter	Description	Default value
kim.mode	The mode that KIM will run in; choices are "LOCAL", "EMBEDDED", or "REMOTE".	LOCAL
kim.soapExposedService.jaxws.security	Determines if KIM services published on the service bus will be secured	true
kim.url	The base URL of KIM services and pages.	\${application.url}/kim

---

# Chapter 9. Services

KIM provides several service APIs with which client applications should interact. These are:

- `org.kuali.rice.kim.api.role.RoleService`
- `org.kuali.rice.kim.api.group.GroupService`
- `org.kuali.rice.kim.api.identity.IdentityService`
- `org.kuali.rice.kim.permission.PermissionService`
- `org.kuali.rice.kim.responsibility.ResponsibilityService`
- `org.kuali.rice.kim.service.PersonService`

These services act as client-side facades to the underlying KIM data and provide important features such as caching.

In the next few sections we will look in-depth at these services. However, for more details, please see the javadocs for these services and the services they delegate to.

## Using the Services

All KIM clients should retrieve service instances using the KIM service locator class **KimApiServiceLocator**. This class contains static methods to retrieve the appropriate Spring bean for the service. An example of retrieving the **IdentityService** service is:

```
IdentityService idmSvc = KimApiServiceLocator.getIdentityService();
```

You would use a similar mechanism for retrieving references to the other KIM services.

## IdentityService

The **IdentityService** is one of the services the client applications will interact with most frequently.

The **IdentityService** contains service methods that allow for the retrieval, creation, and updating of entity information.

Additionally, it also provides caching for the retrieval methods to help increase the performance of service calls for the client application.

## Retrieving Principal Information

To retrieve the principal ID for a user, use the **getPrincipalByPrincipalName** method:

```
Principal info = identityService.getPrincipalByPrincipalName(principalName);
```

Note that KIM, by default, stores principal names in lower case; the `PRNCPL_NM` column of `KRIM_PRNCPL_T` must store values in lower case. If your institution's existing identity systems do not handle lowercase principal names, then there are three points to override that setting:

1. **org.kuali.rice.kim.impl.identity.IdentityServiceImpl** method **getPrincipalByPrincipalName** lowercases the principal name sent in; depending on how principals were integrated into the system it may not need to. Note that **IdentityServiceImpl** method **getPrincipalByPrincipalNameAndPassword** does not lowercase the principal name automatically.
2. **org.kuali.rice.kim.lookup.PersonLookableHelperServiceImpl** method **getSearchResults** also automatically lowercases any principal name sent in; that behavior may also need to be changed
3. Finally, the file **{Rice home}/impl/src/main/resources/org/kuali/ric/kim/bo/datadictionary/KimBaseBeans.xml** hold the data dictionary attribute templates for principal name as **KimBaseBeans-principalName**. The **forceUppercase** attribute is set to false by default, but perhaps should be overridden to true, to force uppercase principal names.

Once these three points have been overridden, you'll be able to use uppercase principal names.

## Retrieving Entity Default Information

To retrieve the default information for an entity, use one of the `getEntityDefaultInfo` methods:

```
EntityDefault infoByEntityId = identityService.getEntityDefault(entityId);
EntityDefault infoByPrincipalId = identityService.getEntityDefaultByPrincipalId(principalId);
```

## Retrieving Reference Information

To retrieve information about a type or status code, use the getter for that type.

Types in KIM are:

- Address type
- Affiliation type
- Citizenship status
- Email type
- Employment status
- Employment type
- Entity name type
- Entity type
- External identifier type
- Phone type

For instance, to retrieve information on an address type code:

```
CodedAttribute addressType = identityService.getAddressType(code);
```

# GroupService

## Retrieving Group Membership Information

To retrieve a list of all groups in which a particular user is a member, use the **getGroupsForPrincipal** method:

```
List<Group> groups = groupService.getGroupsByPrincipalId(principalId);
```

To determine if a user is a member of a particular group, use the **isMemberOfGroup** method:

```
if (groupService.isMemberOfGroup(principalId, groupId)) {
    // Do something special
}
```

To get a list of all members of a group, use the **getMemberPrincipalIds** method:

```
List<String> members = groupService.getMemberPrincipalIds(groupId);
```

## Retrieving Group Information

To retrieve information about a group, use the **getGroup** or **getGroupByNamespaceCodeAndName** methods, depending on whether you know the group's ID or name:

```
Group info = groupService.getGroup(groupId);
Group info = groupService.getGroupByNamespaceCodeAndName(namespaceCode, groupName);
```

# PermissionService

## Checking Permission

To determine if a user has been granted a permission, without considering role qualifications, use the **hasPermission** method:

```
if (permissionService.hasPermission(principalId, namespaceCode, permissionName)) {
    // Do the action
}
```

To determine if a user has been granted a permission, use the **isAuthorized** method:

```
if (permissionService.isAuthorized(principalId, namespaceCode, permissionName, qualification)) {
    // Do the action
}
```

## Retrieving Permission Information

To retrieve a list of principals granted a permission (including any delegates), use the **getPermissionAssignees** method:



```
List<Assignee> people = permissionService.getPermissionAssignees(namespaceCode,
    permissionName, qualification);
```

To retrieve a list of permissions granted to a principal, use the **getAuthorizedPermissions** method:

```
List<Permission> perms = permissionService.getAuthorizedPermissions(principalId,
    namespaceCode, permissionName, qualification);
```

## ResponsibilityService

### Checking Responsibility

To determine if a user has a responsibility, use the **hasResponsibility** method:

```
if (responsibilityService.hasResponsibility(principalId, namespaceCode, responsibilityName, qualification)) {
    // Do the action
}
```

### Retrieving Responsibility Information

To retrieve a list of roles associated with a responsibility, use the **getRoleIdsForResponsibility** method:

```
List<String> roleIds = responsibilityService.getRoleIdsForResponsibility(responsibilityId);
```

## AuthenticationService

### Checking Authentication

The **AuthenticationService** is somewhat different than the other services. The **AuthenticationService** is not typically deployed remotely (unlike the **IdentityService**, **GroupService**, etc.).

Instead, the role of this service is simply to extract the authenticated user's principal name from the **HttpServletRequest** and inform the client-side development framework (typically, the KNS) about this information. KIM itself does not implement full authentication services, but rather relies on other implementations (such as CAS or Shibboleth) to provide this functionality.

The client application can then establish a local session to store the information about the principal that authenticated. This will typically be used in subsequent calls to the KIM services, such as making authorization checks for the principal.

The reference implementation of the **AuthenticationService** simply extracts the **REMOTE\_USER** parameter from the request and presents that as the principal name. This is often sufficient for many authentication providers that are available. However, if necessary this reference implementation can be overridden.

There is only a single method on the **IdentityManagementService** related to authentication.

```
String principalName = authenticationService.getPrincipalName(request);
```

## RoleService

In KIM, Roles are used as a way to associate principals, groups and other roles with permissions and responsibilities. It is therefore not a common or recommended practice to query for whether or not a principal is a member of a Role for the purposes of logic in a client application. It is recommended to use permissions and the **isAuthorized** check to perform this sort of logic.

However, in some cases, querying for this information may be desirable. Or, in even more common cases, one may want to use an API to add or remove members from a Role. These kinds of operations are the responsibility of the **RoleManagementService**. Like the **IdentityManagementService**, this service is a façade which provides caching and delegates to underlying services. Specifically, it delegates to:

- RoleService

## Checking Role Assignment

To determine if a role is assigned to a principal, use the **principalHasRole** method:

```
if (roleService.principalHasRole(principalId, roleIds, qualifications)) {
    // Do something
}
```

## Retrieving Role Information

To retrieve information on a role, use the **getRole** or **getRoleByName** method:

```
Role info = roleService.getRole(roleId);
Role info = roleService.getRoleByNamespaceCodeAndName(namespaceCode, roleName);
```

To retrieve the list of principal IDs assigned to a role, use the **getRoleMemberPrincipalIds** method:

```
Collection<String> principals = roleService.getRoleMemberPrincipalIds(namespaceCode, roleName, qualifications);
```

## Updating Role Membership

To assign a principal to a role, use the **assignPrincipalToRole** method:

```
roleService.assignPrincipalToRole(principalId, namespaceCode, roleName, qualifications);
```

To remove a principal from a role, use the **removePrincipalFromRole** method:

```
roleService.removePrincipalFromRole(principalId, namespaceCode, roleName, qualifications);
```

## Person Service

The **PersonService** is used to aggregate **Entity** and **Principal** data into a data structure called a **Person**. A person is essentially a flattened collection of the various attributes on an entity (name, address, principal

id, principal name, etc). This is intended to allow client applications to more easily interact with the data in the underlying KIM data model for entities and principals.

## Retrieving Personal Information

To retrieve information on a person by principal ID, use the **getPerson** method:

```
Person person = perSvc.getPerson(principalId);
```

To retrieve information on a person by principal name, use the **getPersonByPrincipalName** method:

```
Person person = perSvc.getPersonByPrincipalName(principalName);
```

In order to search for people by a given set of criteria you can use the **findPeople** method:

```
List<Person> people = perSvc.findPeople(criteria);
```

In this case, criteria is a **java.util.Map<String, String>** which contains key-value pairs. The key is the name of the Person property to search on, while the value is the value to search for.

---

# Chapter 10. KimTypeService Callbacks

## Implementing Custom KIM Types

KIM uses the concept of "types" to define additional attributes for its various objects (such as groups, roles, permissions, etc.) and to affect their behavior.

All custom type services must implement a sub-interface of `org.kuali.rice.kim.framework.type.KimTypeService` based on the kind of custom type being created and the KIM objects it will be related to. The current type services supported by KIM are as follows:

- `GroupTypeService`
- `RoleTypeService`
- `PermissionTypeService`
- `ResponsibilityTypeService`
- `DelegationTypeService`

In addition to the interfaces provided above, KIM provides a standard set of implementations of each of these which can be extended by your application in order to inherit standard default behavior (including integration with the KNS Data Dictionary for reading and defining custom attributes). More detailed information about these base classes can be found in the KIM javadocs. Your custom type service class should extend the appropriate subclass and only override the methods necessary to implement your custom behavior. Use the methods in these classes as the basis for your custom code.

For example, you might define a custom `PermissionTypeService` by extending `org.kuali.rice.kns.kim.permission.PermissionTypeServiceBase` as follows:

```
import org.kuali.rice.kns.kim.permission.PermissionTypeServiceBase;

public class MyPermissionTypeService extends PermissionTypeServiceBase {

    @Override
    protected boolean performMatch(Map<String, String> inputMap, Map<String, String> storedMap) {
        if (some_condition_is_true) {
            // perform custom matching logic
            ...
        } else {
            return super.performMatch(inputMap, storedMap); // execute the default logic from base class
        }
    }
}
```

Detailed documentation on the specific methods which can be implemented on `KimTypeService` and its various sub-interfaces can be found in the KIM javadocs.

## Configuring Custom KIM Types

Groups, Roles, Permissions, Responsibilities, and Delegations can all have custom types in KIM. These custom types can be mapped back to the KIM type services that you create. In order to do this, there are a few things you must do:

- Register the KIM Type which points to your custom type service
- Update any of the "typed" KIM objects that you want to point to your new KIM type
- Publish your KIM type service so that it is available on the Kuali Service Bus and the Rice resource loader framework

Currently, there is no way to register a new KIM Type without updating the KIM database using SQL. Fortunately, this is a fairly simple thing to do. The database table storing KIM Types is named KRIM\_TYP\_T. An example of how to insert a new KIM Type into this table in Oracle is below:

```
INSERT INTO KRIM_TYP_T (
  KIM_TYP_ID,
  NMSPC_CD,
  NM,
  SRVC_NM,
  OBJ_ID)
VALUES (
  KRIM_TYP_ID_S.NEXTVAL,
  'MyNamespace',
  'MyPermissionType',
  '{http://myapp.myu.edu}myPermissionTypeService',
  SYS_GUID())
```

One of the most important things to note about this is the service name (SRVC\_NM) column. As we can see in the example above, for this KIM type we are linking it to a service named `{http://myapp.myu.edu}myPermissionTypeService`. This is how KIM will look up your custom type service whenever it needs to load and invoke it.<sup>1</sup> It does this through the Rice resource loading framework which includes locally available services defined in Spring as well as services published on the Kuali Service Bus. For KIM type services, it's generally required to deploy them onto the KSB because the user interface components of KIM will use these when determining which custom attributes may need to be displayed and collected on it's various screens.

More information on how to publish these services can be found in the next section.

Once the KIM type has been registered, it will be assigned an ID, this is the value of the KIM\_TYP\_ID column after the record has been inserted. This ID can then be used to associate the type with the appropriate and desired data elements in KIM.

For example, to associate the custom `PermissionTypeService` you created earlier with one of your permission templates, you can execute the following SQL (assuming the ID of your new KIM type is 10000):

```
UPDATE KRIM_PERM_TMPL_T SET KIM_TYP_ID = '10000'
WHERE NMSPC_CD = 'MyNamespace' AND NM = 'MyPermissionTemplate'
```

Once this is complete, any existing or new permissions you create with this template will use your custom KIM type and it's associated type service.

---

<sup>1</sup>While the service name here is a single string value, it will be parsed into a `javax.xml.namespace.QName` object using that classes `valueOf(...)` method. This means that, for our example of `{http://myapp.myu.edu}myPermissionTypeService`, it will get parsed into a `QName` which is equivalent to `new QName("http://myapp.myu.edu", "myPermissionTypeService")`.

## Publishing Custom KIM Types to the Kual Service Bus

As mentioned previously, KIM type services should be published onto the Kual Service Bus in order to allow the KIM user interface functionality (which is typically deployed on the Rice Standalone Server) to access the services remotely. Since KIM type services are considered "callback" services because of the fact that the standalone server makes callbacks to them, the `org.kuali.rice.ksb.api.bus.support.CallbackServiceExporter` should be used.

Information on how to export and publish a callback service can be found in [CallbackServiceExporter section of the KSB Guide](#).

Assuming you have already wired up your custom `PermissionTypeService` implementation in your Spring file under a bean id of "myPermissionTypeService", an example Spring configuration which will publish the service would look like the following:

```
<bean id="myPermissionTypeService.exporter"
  class="org.kuali.rice.ksb.api.bus.support.CallbackServiceExporter"
  p:callbackService-ref="myPermissionTypeService"
  p:serviceNameSpaceURI="http://myapp.myu.edu"
  p:localServiceName="myPermissionTypeService"
  p:serviceInterface="org.kuali.rice.kim.framework.permission.PermissionTypeService"/>
```

---

# Chapter 11. KIM Database Tables

## Table Name Prefixes

The KIM tables in the Rice database are prefixed by KRIM, which stands for **K**uali **R**ice **I**ntity **M**anagement.

## Unmapped LAST\_UPDT\_DT Columns

Many of the KIM tables have an additional column called LAST\_UPDTD\_DT (of type DATE in Oracle, DATETIME in MySQL) that isn't mapped at the ORM layer. Using this column is entirely optional, and it is unmapped by design. Its purpose is to aid implementers with tracking changes, and with doing data synchronization or extracts against KIM tables. The following sample PL/SQL script (Oracle only) adds to all the tables that contain LAST\_UPDATED\_DT an insert and update trigger to populate it:

```
DECLARE
  CURSOR tables IS
    SELECT table_name
      FROM user_tab_columns
     WHERE column_name = 'LAST_UPDATE_DT'
        AND data_type LIKE 'DATE%'
        ORDER BY 1;
BEGIN
  FOR rec IN tables LOOP
    EXECUTE IMMEDIATE 'CREATE OR REPLACE TRIGGER '||LOWER( SUBSTR( rec.table_name, 1, 27) )||'_tr BEFORE
INSERT OR UPDATE ON '
||LOWER( rec.table_name )||' FOR EACH ROW BEGIN :new.last_update_ts := SYSDATE; END;';
  END LOOP;
END;
/
```

---

# Glossary

## A

Action List	A list of the user's notification and workflow items. Also called the user's Notification List. Clicking an item in the Action List displays details about that notification, if the item is a notification, or displays that document, if it is a workflow item. The user will usually load the document from their Action List in order to take the requested action against it, such as approving or acknowledging the document.
Action List Type	This tells you if the Action List item is a notification or a more specific workflow request item. When the Action List item is a notification, the Action List Type is "Notification."
Action Request	A request to a user or Workgroup to take action on a document. It designates the type of action that is requested, which includes: <ul style="list-style-type: none"><li>• Approve: requests an approve or disapprove action.</li><li>• Complete: requests a completion of the contents of a document. This action request is displayed in the Action List after the user saves an incomplete document.</li><li>• Acknowledge: requests an acknowledgment by the user that the document has been opened - the doc will not leave the Action List until acknowledgment has occurred; however, the document routing will not be held up and the document will be permitted to transition into the processed state if necessary.</li><li>• FYI: a notification to the user regarding the document. Documents requesting FYI can be cleared directly from the Action List. Even if a document has FYI requests remaining, it will still be permitted to transition into the FINAL state.</li></ul>
Action Request Hierarchy	Action requests are hierarchical in nature and can have one parent and multiple children.
Action Requested	The action one needs to take on a document; also the type of action that is requested by an Action Request. Actions that may be requested of a user are: <ul style="list-style-type: none"><li>• Acknowledge: requests that the users states he or she has reviewed the document.</li><li>• Approve: requests that the user either Approve or Disapprove a document.</li><li>• Complete: requests the user to enter additional information in a document so that the content of the document is complete.</li><li>• FYI: intended to simply makes a user aware of the document.</li></ul>
Action Taken	An action taken on a document by a <a href="#">Reviewer</a> in response to an Action Request. The Action Taken may be: <ul style="list-style-type: none"><li>• Acknowledged: Reviewer has viewed and acknowledged document.</li><li>• Approved: Reviewer has approved the action requested on document.</li></ul>



- Blanket Approved: Reviewer has requested a blanket approval up to a specified point in the route path on the document.
- Canceled: Reviewer has canceled the document. The document will not be routed to any more reviewers.
- Cleared FYI: Reviewer has viewed the document and cleared all of his or her pending FYI(s) on this document.
- Completed: Reviewer has completed and supplied all data requested on document.
- Created Document: User has created a document
- Disapproved: Reviewer has disapproved the document. The document will not be routed to any subsequent reviewers for approval. Acknowledge Requests are sent to previous approvers to inform them of the disapproval.
- Logged Document: Reviewer has added a message to the Route Log of the document.
- Moved Document: Reviewer has moved the document either backward or forward in its routing path.
- Returned to Previous Node: Reviewer has returned the document to a previous routing node. When a Reviewer does this, all the actions taken between the current node and the return node are removed and all the pending requests on the document are deactivated.
- Routed Document: Reviewer has submitted the document to the workflow engine for routing.
- Saved: Reviewer has saved the document for later completion and routing.
- Superuser Approved Document: [Superuser](#) has approved the entire document, any remaining routing is cancelled.
- Superuser Approved Node: Superuser has approved the document through all nodes up to (but not including) a specific node. When the document gets to that node, the normal Action Requests will be created.
- Superuser Approved Request: Superuser has approved a single pending Approve or Complete Action Request. The document then goes to the next routing node.
- Superuser Cancelled: Superuser has canceled the document. A Superuser can cancel a document without a pending Action Request to him/her on the document.
- Superuser Disapproved: Superuser has disapproved the document. A Superuser can disapprove a document without a pending Action Request to him/her on the document.

	<ul style="list-style-type: none"><li>• Superuser Returned to Previous Node: Superuser has returned the document to a previous routing node. A Superuser can do this without a pending Action Request to him/her on the document.</li></ul>
Activated	The state of an action request when it has been sent to a user's Action List.
Activation	The process by which requests appear in a user's Action List
Activation Type	Defines how a route node handles activation of Action Requests. There are two standard activation types: <ul style="list-style-type: none"><li>• Sequential: Action Requests are activated one at a time based on routing priority. The next Action Request isn't activated until the previous request is satisfied.</li><li>• Parallel: All Action Requests at the route node are activated immediately, regardless of priority</li></ul>
Active Indicator	An indicator specifying whether an object in the system is active or not. Used as an alternative to complete removal of an object.
Ad Hoc Routing	A type of routing used to route a document to users or groups that are not in the Routing path for that Document Type. When the Ad Hoc Routing is complete, the routing returns to its normal path.
Annotation	Optional comments added by a <a href="#">Reviewer</a> when taking action. Intended to explain or clarify the action taken or to advise subsequent Reviewers.
Approve	A type of workflow action button. Signifies that the document represents a valid business transaction in accordance with institutional needs and policies in the user's judgment. A single document may require approval from several users, at multiple route levels, before it moves to final status.
Approver	The user who approves the document. As a document moves through Workflow, it moves one route level at a time. An Approver operates at a particular route level of the document.
Attachment	The pathname of a related file to attach to a Note. Use the "Browse..." button to open the file dialog, select the file and automatically fill in the pathname.
Attribute Type	Used to strongly type or categorize the values that can be stored for the various attributes in the system (e.g., the value of the arbitrary key/value pairs that can be defined and associated with a given parent object in the system).
Authentication	The act of logging into the system. The Out of the box (OOTB) authentication implementation in Rice does not require a password as it is intended for testing purposes only. This is something that must be enabled as part of an implementation. Various authentication solutions exist, such as CAS or Shibboleth, that an implementer may want to use depending on their needs.
Authorization	Authorization is the permissions that an authenticated user has for performing actions in the system.
Author Universal ID	A free-form text field for the full name of the Author of the Note, expressed as "Lastname, Firstname Initial"

**B**

Base Rule Attribute	<p>The standard fields that are defined and collected for every <a href="#">Routing Rule</a>. These include:</p> <ul style="list-style-type: none"><li>• Active: A true/false flag to indicate if the Routing Rule is active. If false, then the rule will not be evaluated during routing.</li><li>• Document Type: The <a href="#">Document Type</a> to which the Routing Rule applies.</li><li>• From Date: The inclusive start date from which the Routing Rule will be considered for a match.</li><li>• Force Action: a true/false flag to indicate if the review should be forced to take action again for the requests generated by this rule, even if they had taken action on the document previously.</li><li>• Name: the name of the rule, this serves as a unique identifier for the rule. If one is not specified when the rule is created, then it will be generated.</li><li>• Rule Template: The Rule Template used to create the Routing Rule.</li><li>• To Date: The inclusive end date to which the Routing Rule will be considered for a match.</li></ul>
Blanket Approval	<p>Authority that is given to designated <a href="#">Reviewers</a> who can approve a document to a chosen route point. A Blanket Approval bypasses approvals that would otherwise be required in the <a href="#">Routing</a>. For an authorized Reviewer, the <a href="#">Doc Handler</a> typically displays the Blanket Approval button along with the other options. When a Blanket Approval is used, the Reviewers who are skipped are sent Acknowledge requests to notify them that they were bypassed.</p>
Blanket Approve Workgroup	<p>A workgroup that has the authority to Blanket Approve a document.</p>
Branch	<p>A path containing one or more Route Nodes that a document traverses during routing. When a document enters a <a href="#">Split Node</a> multiple branches can be created. A <a href="#">Join Node</a> joins multiple branches together.</p>
Business Rule	<ol style="list-style-type: none"><li>1. Describes the operations, definitions and constraints that apply to an organization in achieving its goals.</li><li>2. A restriction to a function for a business reason (such as making a specific object code unavailable for a particular type of disbursement). Customizable business rules are controlled by Parameters.</li></ol>

**C**

Campus	<p>Identifies the different fiscal and physical operating entities of an institution.</p>
Campus Type	<p>Designates a campus as physical only, fiscal only or both.</p>
Cancel	<p>A workflow action available to document initiators on documents that have not yet been routed for approval. Denotes that the document is void and should be disregarded. Canceled documents cannot be modified in any way and do not route for approval.</p>

Canceled	A routing status. The document is denoted as void and should be disregarded.
CAS - Central Authentication Service	<a href="http://www.jasig.org/cas">http://www.jasig.org/cas</a> - An open source authentication framework. Quali Rice provides support for integrating with CAS as an authentication provider (among other authentication solutions), and Quali also provides an implementation of a CAS server that integrates with Quali Identity Management.
Client	A Java Application Program Interface (API) for interfacing with the Quali Enterprise Workflow Engine.
Client/Server	The use of one computer to request the services of another computer over a network. The workstation in an organization will be used to initiate a business transaction (e.g., a budget transfer). This workstation needs to gather information from a remote database to process the transaction, and will eventually be used to post new or changed information back onto that remote database. The workstation is thus a Client and the remote computer that houses the database is the Server.
Close	A workflow action available on documents in most statuses. Signifies that the user wishes to exit the document. No changes to Action Requests, Route Logs or document status occur as a result of a Close action. If you initiate a document and close it without saving, it is the same as canceling that document.
Comma-separated value	A file format using commas as delimiters utilized in import and export functionality.
Complete	A pending action request to a user to submit a saved document.
Completed	The action taken by a user or group in response to a request in order to finish populating a document with information, as evidenced in the Document Route Log.
Country Restricted Indicator	Field used to indicate if a country is restricted from use in procurement. If there is no value then there is no restriction.
Creation Date	The date on which a document is created.
CSV	See <a href="#">comma-separated value</a>
<b>D</b>	
Date Approved	The date on which a document was most recently approved.
Date Finalized	The date on which a document enters the FINAL state. At this point, all approvals and acknowledgments are complete for the document.
Deactivation	The process by which requests are removed from a user's <a href="#">Action List</a>
Delegate	A user who has been registered to act on behalf of another user. The Delegate acts with the full authority of the Delegator. Delegation may be either <a href="#">Primary Delegation</a> or <a href="#">Secondary Delegation</a> .
Delegate Action List	A separate Action List for Delegate actions. When a Delegate selects a Delegator for whom to act, an Action List of all documents sent to the Delegator is displayed.

For both [Primary](#) and [Secondary Delegation](#) the Delegate may act on any of the entries with the full authority of the Delegator.

Disapprove	A workflow action that allows a user to indicate that a document does not represent a valid business transaction in that user's judgment. The initiator and previous approvers will receive Acknowledgment requests indicating the document was disapproved.
Disapproved	A status that indicates the document has been disapproved by an approver as a valid transaction and it will not generate the originally intended transaction.
Doc Handler	The Doc Handler is a web interface that a Client uses for the appropriate display of a document. When a user opens a document from the Action List or Document Search, the Doc Handler manages access permissions, content format, and user options according to the requirements of the Client.
Doc Handler URL	The URL for the <a href="#">Doc Handler</a> .
Doc Nbr	See <a href="#">Document Number</a> .
Document	Also see <a href="#">E-Doc</a> .  An electronic document containing information for a business transaction that is routed for Actions in KEW. It includes information such as Document ID, Type, Title, Route Status, Initiator, Date Created, etc. In KEW, a document typically has XML content attached to it that is used to make routing decisions.
Document Id	See <a href="#">Document Number</a> .
Document Number	A unique, sequential, system-assigned number for a document
Document Operation	A workflow screen that provides an interface for authorized users to manipulate the XML and other data that defines a document in workflow. It allows you to access and open a document by Document ID for the purpose of performing operations on the document.
Document Search	A web interface in which users can search for documents. Users may search by a combination of document properties such as Document Type or Document ID, or by more specialized properties using the Detailed Search. Search results are displayed in a list similar to an Action List.
Document Status	See also <a href="#">Route Status</a> .
Document Title	The title given to the document when it was created. Depending on the Document Type, this title may have been assigned by the Initiator or built automatically based on the contents of the document. The Document Title is displayed in both the Action List and Document Search.
Document Type	The Document Type defines the routing definition and other properties for a set of documents. Each document is an instance of a Document Type and conducts the same type of business transaction as other instances of that Document Type.  Document Types have the following characteristics: <ul style="list-style-type: none"><li>• They are specifications for a document that can be created in KEW</li></ul>

- They contain identifying information as well as policies and other attributes
- They defines the Route Path executed for a document of that type (Process Definition)
- They are hierarchical in nature may be part of a hierarchy of Document Types, each of which inherits certain properties of its [Parent Document Type](#).
- They are typically defined in XML, but certain properties can be maintained from a graphical interface

Document Type Hierarchy	A hierarchy of Document Type definitions. Document Types inherit certain attributes from their parent Document Types. This hierarchy is also leveraged by various pieces of the system, including the Rules engine when evaluating rule sets and KIM when evaluating certain Document Type-based permissions.
Document Type Label	The human-readable label assigned to a Document Type.
Document Type Name	The assigned name of the document type. It must be unique.
Document Type Policy	These advise various checks and authorizations for instances of a Document Type during the routing process.
Drilldown	A link that allows a user to access more detailed information about the current data. These links typically take the user through a series of inquiries on different business objects.
Dynamic Node	An advanced type of <a href="#">Route Node</a> that can be used to generate complex routing paths on the fly. Typically used whenever the route path of a document cannot be statically defined and must be completely derived from document data.

## E

ECL	<ol style="list-style-type: none"><li>1. An acronym for Educational Community License.</li><li>2. All Quali software and material is available under the Educational Community License and may be adopted by colleges and universities without licensing fees. The open licensing approach also provides opportunities for support and implementation assistance from commercial affiliates.</li></ol>
E-Doc	An abbreviation for electronic documents, also a shorthand reference to documents created with eDocLite.
eDocLite	A framework for quickly building workflow-enabled documents. Allows you to define document screens in XML and render them using XSL style sheets.
Embedded Client	A type of client that runs an embedded workflow engine.
Employee Status	Found on the Person Document; defines the employee's current employment classification (for example, "A" for Active).
Employee Type	Found on the Person Document; defines the employee's position classification (for example, "P" for Professional).

Entity	An Entity record houses identity information for a given Person, Process, System, etc. Each Entity is categorized by its association with an Entity Type.
Entity Attribute	Entities have directory-like information called Entity Attributes that are associated with them  Entity Attributes make up the identity information for an Entity record.
Entity Type	Provides categorization to Entities. For example, a "System" could be considered an Entity Type because something like a batch process may need to interface with the application.
Exception	A workflow routing status indicating that the document routed to an exception queue because workflow has encountered a system error when trying to process the document.
Exception Messaging	The set of services and configuration options that are responsible for handling messages when they cannot be successfully delivered. Exception Messaging is set up when you configure KSB using the properties outlined in KSB Module Configuration.
Exception Routing	A type of routing used to handle error conditions that occur during the routing of a document. A document goes into Exception Routing when the workflow engine encounters an error or a situation where it cannot proceed, such as a violation of a Document Type Policy or an error contacting external services. When this occurs, the document is routed to the parties responsible for handling these exception cases. This can be a group configured on the document or a responsibility configured in KIM. Once one of these responsible parties has reviewed the situation and approved the document, it will be resubmitted to the workflow engine to attempt the processing again.
Extended Attributes	Custom, table-driven business object attributes that can be established by implementing institutions.
Extension Rule Attribute	One of the rule attributes added in the definition of a rule template that extends beyond the base rule attributes to differentiate the routing rule. A Required Extension Attribute has its "Required" field set to True in the rule template. Otherwise, it is an Optional Extension Attribute. Extension attributes are typically used to add additional fields that can be collected on a rule. They also define the logic for how those fields will be processed during rule evaluation.

## F

Field Lookup	The round magnifying glass icon found next to fields throughout the GUI that allow the user to look up reference table information and display (and select from) a list of valid values for that field.
Final	A workflow routing status indicating that the document has been routed and has no pending approval or acknowledgement requests.
Flexible Route Management	A standard KEW routing scheme based on rules rather than dedicated table-based routing.
FlexRM (Flexible Route Module)	The Route Module that performs the Routing for any Routing Rule is defined through FlexRM. FlexRM generates Action Requests when a Rule matches the

data value contained in a document. An abbreviation of "Flexible Route Module."  
A standard KEW routing scheme that is based on rules rather than dedicated table-based routing.

Force Action

A true/false flag that indicates if previous Routing for approval will be ignored when an [Action Request](#) is generated. The flag is used in multiple contexts where requests are generated (e.g., rules, ad hoc routing). If Force Action is False, then prior Actions taken by a user can satisfy newly generated requests. If it is True, then the user needs to take another Action to satisfy the request.

FYI

A workflow action request that can be cleared from a user's Action List with or without opening and viewing the document. A document with no pending approval requests but with pending Acknowledge requests is in Processed status. A document with no pending approval requests but with pending FYI requests is in Final status. See also [Ad Hoc Routing](#) and [Action Request](#).

## G

Group

A Group has members that can be either [Principals](#) or other Groups (nested). Groups essentially become a way to organize Entities (via Principal relationships) and other Groups within logical categories.

Groups can be given authorization to perform actions within applications by assigning them as members of [Roles](#).

Groups can also have arbitrary identity information (i.e., [Group Attributes](#) hanging from them. Group Attributes might be values for "Office Address," "Group Leader," etc.

Groups can be maintained at runtime through a user interface that is capable of workflow.

Group Attribute

Groups have directory-like information called Group Attributes hanging from them. "Group Phone Number" and "Team Leader" are examples of Group Attributes.

Group Attributes make up the identity information for a Group record.

Group Attributes can be maintained at runtime through a user interface that is capable of workflow.

## H

Hierarchical Tree Structure

A hierarchical representation of data in a graphical form.

## I

Initialized

The state of an Action Request when it is first created but has not yet been Activated (sent to a user's Action List).

Initiated

A workflow routing status indicating a document has been created but has not yet been saved or routed. A Document Number is automatically assigned by the system.



**Initiator** A user role for a person who creates (initiates or authors) a new document for routing. Depending on the permissions associated with the Document Type, only certain users may be able to initiate documents of that type.

**Inquiry** A screen that allows a user to view information about a business object.

## J

**Join Node** The point in the routing path where multiple branches are joined together. A Join Node typically has a corresponding [Split Node](#) for which it joins the branches.

## K

**KC - Kuali Coeus** TODO

**KCA - Kuali Commercial Affiliates** A designation provided to commercial affiliates who become part of the Kuali Partners Program to provide for-fee guidance, support, implementation, and integration services related to the Kuali software. Affiliates hold no ownership of Kuali intellectual property, but are full KPP participants. Affiliates may provide packaged versions of Kuali that provide value for installation or integration beyond the basic Kuali software. Affiliates may also offer other types of training, documentation, or hosting services.

**KCB – Kuali Communications Broker** KCB is logically related to KEN. It handles dispatching messages based on user preferences (email, SMS, etc.).

**KEN - Kuali Enterprise Notification** A key component of the Enterprise Integration layer of the architecture framework. Its features include:

- Automatic Message Generation and Logging
- Message integrity and delivery standards
- Delivery of notifications to a user's Action List

**KEW – Kuali Enterprise Workflow** Kuali Enterprise Workflow is a general-purpose electronic routing infrastructure, or workflow engine. It manages the creation, routing, and processing of electronic documents (eDocs) necessary to complete a transaction. Other applications can also use Kuali Enterprise Workflow to automate and regulate the approval process for the transactions or documents they create.

**KFS – Kuali Financial System** Delivers a comprehensive suite of functionality to serve the financial system needs of all Carnegie-Class institutions. An enhancement of the proven functionality of Indiana University's Financial Information System (FIS), KFS meets GASB and FASB standards while providing a strong control environment to keep pace with advances in both technology and business. Modules include financial transactions, general ledger, chart of accounts, contracts and grants, purchasing/accounts payable, labor distribution, budget, accounts receivable and capital assets.

**KIM – Kuali Identity Management** A Kuali Rice module, Kuali Identity Management provides a standard API for persons, groups, roles and permissions that can be implemented by an institution. It also provides an out of the box reference implementation that allows for a university to use Kuali as their Identity Management solution.

KNS – Kuali Nervous System	A core technical module composed of reusable code components that provide the common, underlying infrastructure code and functionality that any module may employ to perform its functions (for example, creating custom attributes, attaching electronic images, uploading data from desktop applications, lookup/search routines, and database interaction).
KPP - Kuali Partners Program	The Kuali Partners Program (KPP) is the means for organizations to get involved in the Kuali software community and influence its future through voting rights to determine software development priorities. Membership dues pay staff to perform Quality Assurance (QA) work, release engineering, packaging, documentation, and other work to coordinate the timely enhancement and release of quality software and other services valuable to the members. Partners are also encouraged to tender functional, technical, support or administrative staff members to the Kuali Foundation for specific periods of time.
KRAD - Kuali Rapid Application Development	TODO
KRMS - Kuali Rules Management System	TODO
KS - Kuali Student	Delivers a means to support students and other users with a student-centric system that provides real-time, cost-effective, scalable support to help them identify and achieve their goals while simplifying or eliminating administrative tasks. The high-level entities of person (evolving roles-student, instructor, etc.), time (nested units of time - semesters, terms, classes), learning unit (assigned to any learning activity), learning result (grades, assessments, evaluations), learning plan (intentions, activities, major, degree), and learning resources (instructors, classrooms, equipment). The concierge function is a self-service information sharing system that aligns information with needs and tasks to accomplish goals. The support for integration of locally-developed processes provides flexibility for any institution's needs.
KSB – Kuali Service Bus	Provides an out-of-the-box service architecture and runtime environment for Kuali Applications. It is the cornerstone of the Service Oriented Architecture layer of the architectural reference framework. The Kuali Service Bus consists of: <ul style="list-style-type: none"> <li>• A services registry and repository for identifying and instantiating services</li> <li>• Run time monitoring of messages</li> <li>• Support for synchronous and asynchronous service and message paradigms</li> </ul>
Kuali	<ol style="list-style-type: none"> <li>1. Pronounced "ku-wah-lee". A partnership organization that produces a suite of community-source, modular administrative software for Carnegie-class higher education institutions. See also <a href="#">Kuali Foundation</a></li> <li>2. (n.) A humble kitchen wok that plays an important role in a successful kitchen.</li> </ol>
Kuali Foundation	Employs staff to coordinate partner efforts and to manage and protect the Foundation's intellectual property. The Kuali Foundation manages a growing portfolio of enterprise software applications for colleges and universities. A lightweight Foundation staff coordinates the activities of Foundation members for critical software development and coordination activities such as source code control, release engineering, packaging, documentation, project management,

software testing and quality assurance, conference planning, and educating and assisting members of the Kualo Partners program.

Kualo Rice

Provides an enterprise-class middleware suite of integrated products that allow both Kualo and non-Kualo applications to be built in an agile fashion, such that developers are able to react to end-user business requirements in an efficient manner to produce high-quality business applications. Built with Service Oriented Architecture (SOA) concepts in mind, KR enables developers to build robust systems with common enterprise workflow functionality, customizable and configurable user interfaces with a clean and universal look and feel, and general notification features to allow for a consolidated list of work "action items." All of this adds up to providing a re-usable development framework that encourages a simplified approach to developing true business functionality as modular applications.

## L

Last Modified Date

The date on which the document was last modified (e.g., the date of the last action taken, the last action request generated, the last status changed, etc.).

## M

Maintenance Document

An e-doc used to establish and maintain a table record.

Message

The full description of a [notification message](#). This is a specific field that can be filled out as part of the Simple Message or Event Message form. This can also be set by the programmatic interfaces when sending notifications from a client system.

Message Queue

Allows administrators to monitor messages that are flowing through the Service Bus. Messages can be edited, deleted or forwarded to other machines for processing from this screen.

## N

Namespace

A Namespace is a way to scope both [Permissions](#) and [Entity Attributes](#). Each Namespace instance is one level of scoping and is one record in the system. For example, "KRA" or "KC" or "KFS" could be a Namespace. Or you could further break those up into finer-grained Namespaces such that they would roughly correlate to functional modules within each application. Examples could be "KRA Rolodex", "KC Grants", "KFS Chart of Accounts".

Out of the box, the system is bootstrapped with numerous Rice namespaces which correspond to the different modules. There is also a default namespace of "KUALI".

Namespaces can be maintained at runtime through a maintenance document.

Note Text

A free-form text field for the text of a Note

Notification Content

This section of a [notification message](#) which displays the actual full message for the notification along with any other content-type-specific fields.

**Notification Message** The overall Notification item or Notification Message that a user sees when she views the details of a notification in her Action List. A Notification Message contains not only common elements such as Sender, Channel, and Title, but also content-type-specific fields.

## O

**OOTB** Stands for "out of the box" and refers to the base deliverable of a given feature in the system.

**Optimistic Locking** A type of "locking" that is placed on a database row by a process to prevent other processes from updating that row before the first process is complete. A characteristic of this locking technique is that another user who wants to make modifications at the same time as another user is permitted to, but the first one who submits their changes will have them applied. Any subsequent changes will result in the user being notified of the optimistic lock and their changes will not be applied. This technique assumes that another update is unlikely.

**Optional Rule Extension Attribute** An Extension Attribute that is not required in a Rule Template. It may or may not be present in a [Routing Rule](#) created from the Template. It can be used as a conditional element to aid in deciding if a Rule matches. These Attributes are simply additional criteria for the Rule matching process.

**Org Doc #** The originating document number.

**Organization** Refers to a unit within the institution such as department, responsibility center, campus, etc.

**Organization Code** Represents a unique identifier assigned to units at many different levels within the institution (for example, department, responsibility center, and campus).

## P

**Parameter Component Code** Code identifying the parameter Component.

**Parameter Description** This field houses the purpose of this parameter.

**Parameter Name** This will be used as the identifier for the parameter. Parameter values will be accessed using this field and the namespace as the key.

**Parameter Type Code** Code identifying the parameter type. Parameter Type Code is the primary key for its' table.

**Parameter Value** This field houses the actual value associated with the parameter.

**Parent Document Type** A Document Type from which another [Document Type](#) derives. The child type can inherit certain properties of the parent type, any of which it may override. A Parent Document Type may have a parent as part of a hierarchy of document types.

**Parent Rule** A Routing Rule in KEW from which another Routing Rule derives. The child Rule can inherit certain properties of the parent Rule, any of which it may override. A Parent Rule may have a parent as part of a hierarchy of Rules.

**Permission** Permissions represent fine grained actions that can be mapped to functionality within a given system. Permissions are scoped to [Namespace](#) which roughly correlate to modules or sections of functionality within a given system.

A developer would code authorization checks in their application against these permissions.

Some examples would be: "canSave", "canView", "canEdit", etc.

Permissions are aggregated by [Roles](#).

Permissions can be maintained at runtime through a user interface that is capable of workflow; however, developers still need to code authorization checks against them in their code, once they are set up in the system.

#### Attributes

1. Id - a system generated unique identifier that is the primary key for any Permission record in the system
2. Name - the name of the permission; also a human understandable unique identifier
3. Description - a full description of the purpose of the Permission record
4. Namespace - the reference to the associated [Namespace](#)

#### Relationships

1. Permission to [Role](#) - many to many; this relationship ties a Permission record to a Role that is authorized for the Permission
2. Permission to [Namespace](#) - many to one; this relationship allows for scoping of a Permission to a Namespace that contains functionality which keys its authorization checking off of said

Person Identifier	The username of an individual user who receives the document ad hoc for the Action Requested
Person Role	Creates or maintains the list used in selection of personnel when preparing the Routing Form document.
Pessimistic Locking	A type of lock placed on a database row by a process to prevent other processes from reading or updating that row until the first process is finished. This technique assumes that another update is likely.
Plugins	A plugin is a packaged set of code providing essential services that can be deployed into the Rice standalone server. Plugins usually contains only classes used in routing such as custom rules or searchable attributes, but can contain client application specific services. They are usually used only by clients being implemented by the 'Thin Client' method
Post Processor	A routing component that is notified by the workflow engine about various events pertaining to the routing of a specific document (e.g., node transition, status change, action taken). The implementation of a Post Processor is typically specific to a particular set of Document Types. When all required approvals are completed, the engine notifies the Post Processor accordingly. At this point, the Post Processor is responsible for completing the business transaction in the manner appropriate to its Document Type.

Posted Date/Time Stamp	A free-form text field that identifies the time and date at which the Notes is posted.
Postal Code	Defines zip code to city and state cross-references.
Preferences	User options in an Action List for displaying the list of documents. Users can click the Preferences button in the top margin of the Action List to display the Action List Preferences screen. On the Preferences screen, users may change the columns displayed, the background colors by Route Status, and the number of documents displayed per page.
Primary Delegation	The Delegator turns over full authority to the Delegate. The Action Requests for the Delegator only appear in the Action List of the Primary Delegate. The Delegation must be registered in KEW or KIM to be in effect.
Principal	<p>A Principal represents an <a href="#">Entity</a> that can authenticate into the system. One can roughly correlate a Principal to a login username. Entities can exist in KIM without having permissions or authorization to do anything; therefore, a Principal must exist and must be associated with an Entity in order for it to have access privileges. All authorization that is not specific to <a href="#">Groups</a> is tied to a Principal.</p> <p>In other words, an Entity is for identity while a Principal is for access management.</p> <p>Also note that an Entity is allowed to have multiple Principals associated with it. The use case typically given here is that a person may apply to a school and receive one log in for the application system; however, once accepted, they may receive their official login, but use the same identity information set up for their Entity record.</p>
Processed	A routing status indicating that the document has no pending approval requests but still has one or more pending acknowledgement requests.

## R

Recipient Type	The type of entity that is receiving an Action Request. Can be a user, workgroup, or role.
Required Rule Extension Attribute	An Extension Attribute that is required in a Rule Template. It will be present in every Routing Rule created from the Template.
Responsibility	See <a href="#">Responsible Party</a> .
Responsibility Id	A unique identifier representing a particular responsibility on a rule (or from a <a href="#">route module</a> ) This identifier stays the same for a particular responsibility no matter how many times a rule is modified.
Responsible Party	The Reviewer defined on a routing rule that receives requests when the rule is successfully executed. Each routing rule has one or more responsible parties defined.
Reviewer	A user acting on a document in his/her <a href="#">Action List</a> and who has received an <a href="#">Action Request</a> for the document.
Rice	An abbreviation for Kualu Rice.
Role	Roles aggregate <a href="#">Permissions</a> When Roles are given to <a href="#">Entities</a> (via their relationship with Principals) or <a href="#">Groups</a> an authorization for all associated Permissions is granted.

Route Header Id	Another name for the <a href="#">Document Id</a> .
Route Log	Displays information about the routing of a document. The Route Log is usually accessed from either the Action List or a Document Search. It displays general document information about the document and a detailed list of Actions Taken and pending <a href="#">Action Requests</a> for the document. The Route Log can be considered an audit trail for a document.
Route Module	A routing component that the engine uses to generate action requests at a particular <a href="#">Route Node</a> . <a href="#">FlexRM</a> (Flexible Route Module) is a general Route Module that is rule-based. Clients can define their own Route Modules that can conduct specialized Routing based on routing tables or any other desired implementation.
Route Node	<p>Represents a step in the routing process of a document type. Route node "instances" are created dynamically as a document goes through its routing process and can be defined to perform any function. The most common functions are to generate Action Requests or to split or join the route path.</p> <ul style="list-style-type: none"><li>• Simple: do some arbitrary work</li><li>• Requests: generate action requests using a Route Module or the Rules engine</li><li>• Split: split the route path into one or more parallel branches</li><li>• Join: join one or more branches back together</li><li>• Sub Process: execute another route path inline</li><li>• Dynamic: generate a dynamic route path</li></ul>
Route Path	The path a document follows during the routing process. Consists of a set of route nodes and branches. The route path is defined as part of the <a href="#">document type</a> definition.
Route Status	<p>The status of a document in the course of its routing:</p> <ul style="list-style-type: none"><li>• Approved: These documents have been approved by all required reviewers and are waiting additional postprocessing.</li><li>• Cancelled: These documents have been stopped. The document's initiator can 'Cancel' it before routing begins or a reviewer of the document can cancel it after routing begins. When a document is cancelled, routing stops; it is not sent to another Action List.</li><li>• Disapproved: These documents have been disapproved by at least one reviewer. Routing has stopped for these documents.</li><li>• Enroute: Routing is in progress on these documents and an action request is waiting for someone to take action.</li><li>• Exception: A routing exception has occurred on this document. Someone from the Exception Workgroup for this Document Type must take action on this document, and it has been sent to the Action List of this workgroup.</li><li>• Final: All required approvals and all acknowledgements have been received on these documents. <u>No changes are allowed to a document that is in Final status.</u></li></ul>

- **Initiated:** A user or a process has created this document, but it has not yet been routed to anyone's Action List.
- **Processed:** These documents have been approved by all required users, and is completed on them. They may be waiting for Acknowledgements. No further action is needed on these documents.
- **Saved:** These documents have been saved for later work. An author (initiator) can save a document before routing begins or a reviewer can save a document before he or she takes action on it. When someone saves a document, the document goes on that person's Action List.

**Routed By User** The user who submits the document into routing. This is often the same as the Initiator. However, for some types of documents they may be different.

**Routing** The process of moving a document through its route path as defined in its Document Type. Routing is executed and administered by the workflow engine. This process will typically include generating Action Requests and processing actions from the users who receive those requests. In addition, the Routing process includes callbacks to the Post Processor when there are changes in document state.

**Routing Priority** A number that indicates the routing priority; a smaller number has a higher routing priority. Routing priority is used to determine the order that requests are activated on a route node with sequential activation type.

**Routing Rule** A record that contains the data for the [Rule Attributes](#) specified in a [Rule Template](#). It is an instance of a Rule Template populated to determine the appropriate Routing. The Rule includes the Base Attributes, Required Extension Attributes, Responsible Party Attributes, and any Optional Extension Attributes that are declared in the Rule Template. Rules are evaluated at certain points in the routing process and, when they fire, can generate Action Requests to the responsible parties that are defined on them.

Technical considerations for a Routing Rules are:

- Configured via a GUI (or imported from XML)
- Created against a Rule Template and a Document Type
- The Rule Template and its list of Rule Attributes define what fields will be collected in the Rule GUI
- Rules define the users, groups and/or roles who should receive action requests
- Available Action Request Types that Rules can route
  - Complete
  - Approve
  - Acknowledge
  - FYI
- During routing, Rule Evaluation Sets are "selected" at each node. Default is to select by Document Type and Rule Template defined on the Route Node



- Rules match (or 'fire') based on the evaluation of data on the document and data contained on the individual rule
- Examples
  - If dollar amount is greater than \$10,000 then send an Approval request to Joe.
  - If department is "HR" request an Acknowledgment from the HR.Acknowledgers workgroup.

#### Rule Attribute

Rule attributes are a core KEW data element contained in a document that controls its Routing. It participates in routing as part of a Rule Template and is responsible for defining custom fields that can be rendered on a routing rule. It also defines the logic for how rules that contain the attribute data are evaluated.

Technical considerations for a Rule Attribute are:

- They might be backed by a Java class to provide lookups and validations of appropriate values.
- Define how a Routing Rule evaluates document data to determine whether or not the rule data matches the document data.
- Define what data is collected on a rule.
- An attribute typically corresponds to one piece of data on a document (i.e dollar amount, department, organization, account, etc.).
- Can be written in Java or defined using XML (with matching done by XPath).
- Can have multiple GUI fields defined in a single attribute.

#### Rule QuickLinks

A list of document groups with their document hierarchies and actions that can be selected. For specific document types, you can create the rule delegate.

#### Rule Template

A Rule Template serves as a pattern or design for the routing rules. All of the Rule Attributes that include both Required and `_Optional_` are contained in the Rule Template; it defines the structure of the routing rule of FlexRM. The Rule Template is also used to associate certain Route Nodes on a document type to routing rules.

Technical considerations for a Rule Templates are:

- They are a composition of Rule Attributes
- Adding a 'Role' attribute to a template allows for the use of the Role on any rules created against the template
- When rule attributes are used for matching on rules, each attribute is associated with the other attributes on the template using an implicit 'and' logic attributes
- Can be used to define various other aspects to be used by the rule creation GUI such as rule data defaults (effective dates, ignore previous, available request types, etc)

**S**

Save	A workflow action button that allows the Initiator of a document to save their work and close the document. The document may be retrieved from the initiator's Action List for completion and routing at a later time.
Saved	A routing status indicating the document has been started but not yet completed or routed. The Save action allows the initiator of a document to save their work and close the document. The document may be retrieved from the initiator's action list for completion and routing at a later time.
Searchable Attributes	<p>Attributes that can be defined to index certain pieces of data on a document so that it can be searched from the <a href="#">Document Search screen</a>.</p> <p>Technical considerations for a Searchable Attributes are:</p> <ul style="list-style-type: none"><li>• They are responsible for extracting and indexing document data for searching</li><li>• They allow for custom fields to be added to Document Search for documents of a particular type</li><li>• They are configured as an attribute of a Document Type</li><li>• They can be written in Java or defined in XML by using Xpath to facilitate matching</li></ul>
Secondary Delegation	<p>The Secondary Delegate acts as a temporary backup Delegator who acts with the same authority as the primary Approver/the Delegator when the Delegator is not available. Documents appear in the Action Lists of both the Delegator and the Delegate. When either acts on the document, it disappears from both Action Lists.</p> <p>Secondary Delegation is often configured for a range of dates and it must be registered in KEW or KIM to be in effect.</p>
Service Registry	Displays a read-only view of all of the services that are exposed on the Service Bus and includes information about them (for example, IP Address, or Endpoint URL).
Simple Node	A type of node that can perform any function desired by the implementer. An example implementation of a simple node is the node that generates Action Requests from route modules.
SOA	An acronym for Service Oriented Architecture.
Special Condition Routing	This is a generic term for additional route levels that might be triggered by various attributes of a transaction. They can be based on the type of document, attributes of the accounts being used, or other attributes of the transaction. They often represent special administrative approvals that may be required.
Split Node	A node in the routing path that can split the route path into multiple branches.
Spring	The <a href="#">Spring Framework</a> is an open source application framework for the Java platform.
State	Defines U.S. Postal Service codes used to identify states.
Status	On an Action List; also known as Route Status. The current location of the document in its routing path.

Submit	A workflow action button used by the initiator of a document to begin workflow routing for that transaction. It moves the document (through workflow) to the next level of approval. Once a document is submitted, it remains in 'ENROUTE' status until all approvals have taken place.
Superuser	A user who has been given special permission to perform Superuser Approvals and other Superuser actions on documents of a certain Document Type.
Superuser Approval	Authority given Superusers to approve a document of a chosen Route Node. A Superuser Approval action bypasses approvals that would otherwise be required in the Routing. It is available in Superuser Document Search. In most cases, reviewers who are skipped are not sent Acknowledge Action Requests.
Superuser Document Search	A special mode of Document Search that allows Superusers to access documents in a special Superuser mode and perform administrative functions on those documents. Access to these documents is governed by the user's membership in the Superuser Workgroup as defined on a particular Document Type.

## T

Thread pool	A technique that improves overall system performance by creating a pool of threads to execute multiple tasks at the same time. A task can execute immediately if a thread in the pool is available or else the task waits for a thread to become available from the pool before executing.
Title	<p>A short summary of the notification message. This field can be filled out as part of the Simple Message or Event Message form. In addition, this can be set by the programmatic interfaces when sending notifications from a client system.</p> <p>This field is equivalent to the "Subject" field in an email.</p>

## U

URL	An acronym for Uniform Resource Locator.
User	A person who can log in and use the application. This term is synonymous with "Principal" in KIM. "Whereas Entity Id represents a unique Person, Principal Id represents a set of login information for that Person."

## V

Viewer	A user(s) who views a document during the routing process. This includes users who have action requests generated to them on a document.
--------	--

## W

Web Service Client	A type of client that connects to a standalone KEW server using Web Services.
Wildcard	A character that may be substituted for any of a defined subset of all possible characters.
Workflow	Electronic document routing, approval and tracking. Also known as Workflow Services or Kualu Enterprise Workflow (KEW). The Kualu infrastructure service

that electronically routes an e-doc to its approvers in a prescribed sequence, according to established business rules based on the e-doc content. See also [Kuali Enterprise Workflow](#).

Workflow Engine

The component of KEW that handles initiating and executing the route path of a document.

Workflow QuickLinks

A web interface that provides quick navigation to various functions in KEW. These include:

- Quick EDoc Watch: The last five Actions taken by this user. The user can select and repeat these actions.
- Quick EDoc Search: The last five EDocs searched for by this user. The user can select one and repeat that search.
- Quick Action List: The last five document types the user took action with. The user can select one and repeat that action.

## X

XML

See also [XML Ingestor](#).

1. An acronym for Extensible Markup Language.
2. Used for data import/export.

XML Ingestor

A workflow function that allows you to browse for and upload XML data.

XML RuleAttribute

Similar in functionality to a RuleAttribute but built using XML only