
Kuali Rice 2.3.0-M2-SNAPSHOT KNS to KRAD Conversion Guide

Released: 05-07-2013

Table of Contents

Introduction	1
Data Dictionary	1
Inquiry	1
Lookup	1
LookupDefinition in KNS Data Dictionary to LookupView in KRAD	1
Base Lookup Bean	3
Criteria Fields	3
Results Fields	4
Lookupable / LookupableHelper Service	4
Lookupable	4
LookupableHelper	7
Transactional Documents	8
Conversion Tools	8
Introduction	8
Lookup	8
Inquiry	8
Maintenance	8
Transactional	8
KNS to KRAD Conversion Script	8

Introduction

Identify audience

Data Dictionary

Inquiry

Lookup

LookupDefinition in KNS Data Dictionary to LookupView in KRAD

Note that the code example is very inclusive to show the configurations that have changed. As a result the lookups themselves have conflicting settings (i.e. why have custom search buttons when hiding them).

KNS Code example:

Kuali Rice 2.3.0-M2-SNAPSHOT KNS to KRAD Conversion Guide

```
1 <bean id="EntityType-lookupDefinition" parent="LookupDefinition">
2   <property name="title" value="Entity Type Lookup"/>
3   <property name="menubar" value="<a href=&quot;;javascript:void(0)&quot;
4     onclick=&quot;alert('JavaScript triggered action.*)&quot;&gt;Custom Button</a&gt;"/>
5   <property name="numOfColumns" value="2"/>
6   <property name="extraButtonSource" value="images/tinybutton-createnew.gif"/>
7   <property name="extraButtonParams" value="createNew"/>
8   <property name="disableSearchButtons" value="true"/>
9   <property name="lookupFields">
10    <list>
11     <bean parent="FieldDefinition" p:attributeName="code"/>
12     <bean parent="FieldDefinition" p:attributeName="name" p:noLookup="true"/>
13     <bean parent="FieldDefinition" p:attributeName="active" p:defaultValue="Y"/>
14    </list>
15  </property>
16  <property name="resultFields">
17    <list>
18     <bean parent="FieldDefinition" p:attributeName="code" p:triggerOnChange="true"/>
19     <bean parent="FieldDefinition" p:attributeName="name" p:noLookup="true"/>
20     <bean parent="FieldDefinition" p:attributeName="sortCode" p:forceInquiry="true"/>
21     <bean parent="FieldDefinition" p:attributeName="active"/>
22    </list>
23  </property>
24  <property name="defaultSort">
25    <bean parent="SortDefinition">
26      <property name="sortAscending" value="false" />
27      <property name="attributeNames">
28        <list>
29         <value>code</value>
30        </list>
31      </property>
32    </bean>
33  </property>
34 </bean>
```

KRAD code example:

```
1 <bean id="EntityTypeLookupView" parent="Uif-LookupView">
2   <property name="dataObjectClassName" value="org.kuali.rice.kim.impl.identity.EntityTypeBo"/>
3   <property name="headerText" value="Entity Type Lookup" />
4   <property name="page.header.lowerGroup.items">
5     <list merge="true">
6       <bean parent="Uif-SecondaryActionButton" p:actionLabel="Custom Button"
7         p:actionScript="alert('JavaScript triggered action.*)&quot;"/>
8     </list>
9   </property>
10  <property name="criteriaGroup.layoutManager.numberOfColumns" value="4"/>
11  <property name="criteriaGroup.footer">
12    <bean parent="Uif-LookupCriteriaFooter">
13      <property name="items">
14        <list merge="true">
15         <bean parent="Uif-PrimaryActionButton" p:methodToCall="createNew" p:actionLabel="create new"/>
16        </list>
17      </property>
18    </bean>
19  </property>
20  <property name="renderSearchButtons" value="false"/>
21  <property name="criteriaFields">
22    <list>
23     <bean parent="Uif-LookupCriteriaInputField" p:propertyName="code" p:triggerOnChange="true"/>
24     <bean parent="Uif-LookupCriteriaInputField" p:propertyName="name" p:quickfinder.render="false"/>
25     <bean parent="Uif-LookupCriteriaInputField" p:propertyName="active" p:defaultValue="Y"/>
26    </list>
27  </property>
28  <property name="resultFields">
29    <list>
30     <bean parent="Uif-DataField" p:propertyName="code"/>
31     <bean parent="Uif-DataField" p:propertyName="name"/>
32     <bean parent="Uif-DataField" p:propertyName="sortCode"/>
33     <bean parent="Uif-DataField" p:propertyName="active"/>
34    </list>
35  </property>
36  <property name="defaultSortAscending" value="false"/>
37  <property name="defaultSortAttributeNames">
38    <list>
39     <value>code</value>
40    </list>
41  </property>
42 </bean>
```

```
43 </property>  
46 </bean>
```

Base Lookup Bean

- Lookups are now defined via **Uif-LookupView** definitions instead of the KNS **LookupDefinition** beans. [KNS line: 1, KRAD line: 1] [script conversion](#)
- The title of the view is specified via the **headerText** property instead of the **title** property. [KNS line: 2, KRAD line: 3] [script conversion](#)
- The data object class is specified via the **dataObjectClassName** property and is no longer specified via the **BusinessObjectEntry** as it doesn't exist in KRAD anymore. A **businessObject** is no longer required and any object can be used. [KNS line: n/a, KRAD line: 2] [script conversion](#)
- Instead of specifying the supplemental menu bar via the **menubar** property, the **page.header.lowerGroup.items** property is used. [KNS line: 3-4, KRAD line: 4-9] [script conversion](#)
- To use multiple columns for the criteria fields use the **criteriaGroup.layoutManager.numberOfColumns** property to configure the layout manager instead of specifying the **numOfColumns** property. Note that in the layout manager the field label and the field itself have their own columns and therefore the old **numOfColumns** value needs to be doubled. [KNS line: 5, KRAD line: 10] [script conversion](#)

Criteria Fields

- The property that contains the lookup criteria fields was renamed from **lookupFields** to **criteriaFields**. [KNS line: 9, KRAD line: 21] [script conversion](#)
- Criteria fields changed from **FieldDefinition** to **Uif-LookupCriteriaInput**. [KNS line: 11-13, KRAD line: 23-25] [script conversion](#)
- The **attributeName** property on the field is now a **propertyName** property. [KNS line: 11-13, KRAD line: 23-25] [script conversion](#)
- Not rendering a quickfinder on a lookup criteria field is specified via the **quickfinder.render** property instead of the **noLookup** property. [KNS line: 12, KRAD line 24] [script conversion](#)
- To use multiple columns for the criteria fields use the **criteriaGroup.layoutManager.numberOfColumns** property to configure the layout manager instead of specifying the **numOfColumns** property. Note that in the layout manager the field label and the field itself have their own columns and therefore the old **numOfColumns** value needs to be doubled. [KNS line: 5, KRAD line: 10] [script conversion](#)
- Adding additional buttons to the bottom of the search criterias is done by adding a **Uif-PrimaryActionButton** to the **criteriaGroup.footer** item list instead of using the **extraButtonSource** and **extraButtonParms** properties. [KNS line: 6-7, KRAD line: 11-19] [script conversion](#)
- The property that specifies whether or not to show the search buttons changed from **disableSearchButtons** to **renderSearchButtons**. [KNS line: 8, KRAD line: 20] [script conversion](#)

Results Fields

- Criteria fields changed from **FieldDefinition** to **Uif-DataField**. [KNS line: 18-21, KRAD line: 30-33] [script conversion](#)
- The **attributeName** property on the field is now a **propertyName** property. [KNS line: 18-21, KRAD line: 30-33] [script conversion](#)
- The **forceInquiry** property does not need to be set as inquiry links are rendered automatically. Set the **inquiry.render** property on the **Uif-DataField** to false to suppress the rendering of the inquiry link. [KNS line: 20, KRAD line: n/a] [script conversion](#)
- Custom actions in the result rows are no longer specified by overriding the **getCustomActionUrls** method of **LookupableHelperService**. Instead the **resultsGroup.lineActions** property list is extended or overridden. Look in **UifLookupDefinition.xml** to see how the edit, copy and delete actions are defined. [manual conversion](#)
- The **SortDefinition** bean with the **sortAscending** and **attributeName** properties has been replaced. The sort order is specified via the **defaultSortAscending** property and the sort fields via the **defaultSortAttributeName** property list on the **Uif-Lookup-View**. [KNS line: 24-33, KRAD line: 36-43] [script conversion](#)

Lookupable / LookupableHelper Service

KRAD combines the **Lookupable** and **LookupableHelperService** from KNS. The **LookupableImpl** does not need to be extended unless advanced customizations are required.

Lookupable

- **set/getBusinessObjectClass** have been renamed to **set/getDataObjectClass**. Since now all data objects are supported the requirement that the object is a **BusinessObject** has been removed. The rename of these methods reflect this change. However, don't use this method. In KRAD the data object is specified via the **dataObjectClassName** property of the extended **Uif-LookupView** bean.

```
<bean id="Sample-LookupView" parent="Uif-LookupView">
  <property name="dataObjectClassName" value="org.kuali.rice.SampleBo" />
  ...
</bean>
```

- **getHtmlMenuBar/getSupplementalMenuBar** have been removed. Instead these menu bars are configured via **Uif**. The **HtmlMenuBar** used to add additional HTML content to the right of the "Create New" button, while the **SupplementalMenuBar** replaces the "Create New" button with the specified HTML content. Note that with KRAD the "Create New" has been moved out of the lookup header area and instead is right below the header, still at the right side. The following sample displays how to add content after the "Create New". In this example a **Message** is used but any **Uif** component could be used. Append a custom button after the "Create New":

```
<bean id="Sample-LookupView" parent="Uif-LookupView">
  ...
  <property name="page.header.lowerGroup.items">
    <list>
      <bean parent="Uif-CreateNewLink" />
      <bean parent="Uif-SecondaryActionButton" p:actionLabel="Custom Button"
        p:actionScript="alert('JavaScript triggered action.')" />
    </list>
  </property>
  ...
</bean>
```

Append a custom message to the right in the header

```
<bean id="Sample-LookupView" parent="Uif-LookupView">
  ...
  <property name="header.rightGroup">
    <bean parent="Uif-HeaderRightGroup">
      <property name="items">
        <list>
          <bean parent="Uif-Message" p:messageText="Right Group of headerText"/>
        </list>
      </property>
    </bean>
  </property>
  ...
</bean>
```

Instead of "rightGroup", "upperGroup" and "lowerGroup" can be used to position components above and below the header.

- **getRows** has been removed. Criteria fields can be conditionally displayed and configured via Uif.
- **getColumnns** has been removed. The result columns are now specified via the Uif-LookupView (see resultFields property).

```
<bean id="Sample-LookupView" parent="Uif-LookupView">
  ...
  <property name="resultFields">
    <list>
      <bean parent="Uif-DataField" p:propertyName="cd" />
      <bean parent="Uif-DataField" p:propertyName="description" />
    </list>
  </property>
  ...
</bean>
```

- **validateSearchParameters** takes the LookupForm as an additional parameter and returns a boolean value indicating that no validation error message has occurred (true = no error). Warning and informational messages do not affect this return indicator. A ValidationException is no longer thrown. Try using Uif configurations for more complex validation that the default KRAD validation can't handle (e.g. [Constraints](#))
- **performLookup** has been renamed to performSearch. The resultTable parameter has been removed and takes searchCriteria as an additional parameter.
- **getSearchResults** takes the LookupForm and indicator for bounded/unbounded search as additional parameters. **getSearchResultsUnbounded** has been removed.
- **performClear** takes the **searchCriteria** map as an additional parameter since it is no longer stored in the Lookupable. The **searchCriteria** map is returned after clearing the criteria and setting their default values.
- **getReturnUrl** has been renamed to getReturnUrlForResults which accepts the LinkField that returns the result from the lookup and the model. Configuration via the Uif-LookupView is also possible:

```
<bean id="Sample-LookupView" parent="Uif-LookupView">
  ...
  <property name="resultsReturnField">
    <bean parent="Uif-linkField" p:href="http://www.kuali.org" p:linkText="Kuali"/>
  </property>
  ...
</bean>
```

- **getCreateNewUrl** has been removed. The create new url is now specified via the Uif-LookupView:

```
<bean id="Sample-LookupView" parent="Uif-LookupView">
  ...
  <property name="page.header.lowerGroup.items">
```

```
<list>
  <bean parent="Uif-Link" p:linkText="Create New" p:href="http://www.kuali.org">
    <property name="cssClasses">
      <list merge="true">
        <value>uif-createNewLink</value>
      </list>
    </property>
  </bean>
</list>
</property>
...
</bean>
```

- **getTitle** has been removed. The title is now specified via the Uif-LookupView (see headerText property).

```
<bean id="Sample-LookupView" parent="Uif-LookupView">
  <property name="headerText" value="Sample Lookup" />
  ...
</bean>
```

- **getReturnKeys** takes the **LookupView**, **LookupForm** and the data object as an additional parameter.
- **getReturnLocation** has been removed. The return location is now stored on the form. Use **LookupForm.getReturnLocation**.
- **getExtraButtonSource** has been removed. Buttons are configured via **Uif-LookupCriteriaGroup** (see footer property).
- **getExtraButtonParams** has been removed. Buttons are configured via **Uif-LookupCriteriaGroup** (see footer property).
- **checkForAdditionalFields** has been removed. Use [progressive disclosure](#) of the Uif.
- **getDefaultSortColumns** has been removed. Sort columns are configured via Uif-LookupView (see defaultSortAttributeNames property).

```
<bean id="Sample-LookupView" parent="Uif-LookupView">
  ...
  <property name="defaultSortAttributeNames">
    <list>
      <value>cd</value>
      <value>description</value>
    </list>
  </property>
  ...
</bean>
```

- **set/getDocFormKey** has been removed. The document form key is stored on the form. Use **LookupForm.getFormKey**.
- **setFieldConversions** has been removed. The field conversion is specified via Uif (see quickfinder.fieldConversions property).

```
<bean parent="Uif-InputField">
  ...
  <property name="quickfinder.fieldConversions">
    <map>
      <entry key="cd" value="sampleCd" />
      <entry key="description" value="sample.description" />
    </map>
  </property>
  ...
</bean>
```

- **setReadOnlyFieldsList** has been removed. Read only criteria fields are specified via Uif (see **readOnly** property).

```
<bean id="Sample-LookupView" parent="Uif-LookupView">
  ...
  <property name="criteriaFields">
    <list>
      <bean parent="Uif-LookupCriteriaInputField" p:propertyName="namespace"
        p:readOnly="true" />
      <bean parent="Uif-LookupCriteriaInputField" p:propertyName="cd" />
    </list>
  </property>
  ...
</bean>
```

- **set/getLookupableHelperService** has been removed. Lookupable helper services are specified via **Uif-LookupView**:

```
<bean id="Sample-LookupView" parent="Uif-LookupView">
  <property name="viewHelperServiceClass"
    value="org.kuali.rice.SampleLookupableHelperServiceImpl" />
  ...
</bean>
```

- **isSearchUsingOnlyPrimaryKeyValues** has been removed.
- **getPrimaryKeyFieldLabels** has been removed.
- **shouldDisplayHeaderNonMaintActions** has been removed.
- **shouldDisplayLookupCriteria** has been removed.
- **performCustomAction** has been removed. Create **methodToCall** methods for the actions in the [controller](#) for the data object.
- **getExtraField** has been removed (see [getHtmlMenuBar/getSupplementalMenuBar](#) above).
- **get/setExtraOnLoad** has been removed. OnLoad scripts can be specified via the **Uif-LookupView**:

```
<bean id="Sample-LookupView" parent="Uif-LookupView">
  <property name="onLoadScript" value="alert('Hi!')" />
  ...
</bean>
```

- **applyFieldAuthorizationsFromNestedLookups** has been removed.
- **applyConditionalLogicForFieldDisplay** has been removed. Conditional displaying of criteria field is done through Uif configuration.

```
<bean id="Sample-LookupView" parent="Uif-LookupView">
  ...
  <property name="criteriaFields">
    <list>
      <bean parent="Uif-LookupCriteriaInputField" p:propertyName="cd"
        p:readOnly="@{!#empty(#dp.lookupCriteria['cd'])}" />
      <bean parent="Uif-DataField" p:propertyName="description"
        p:required="@{#dp.lookupCriteria['CD'] == 'A_CD'}" />
      <bean parent="Uif-LookupCriteriaInputField" p:propertyName="namespace"
        p:render="@{#dp.lookupCriteria['CD'] == 'A_CD'}" />
    </list>
  </property>
  ...
</bean>
```

LookupableHelper

- **getInquiryUrlblah**

Transactional Documents

Conversion Tools

Introduction

Lookup

Inquiry

Maintenance

Transactional

KNS to KRAD Conversion Script

KNS to KRAD conversion is setup into four steps:

1. Convert data dictionary into krad compliant format

- Convert validation patterns to constraint
- Convert control definitions into uif controls

2. Convert KNS definitions into KRAD UIF components

- Convert inquiry and lookup definitions into view
- Convert section definition into uif groups
- Convert field definitions into uif input fields

3. Convert struts to spring mvc

- Convert struts and their actions into uif controllers
- Convert action paths into request mapping annotations in the controller

4. Convert jsp and tags into uif components

- Convert jsp tags into UIF components
- Convert jstl calls into spring el conditionals
- Convert document into view
- Convert kul:tabs into UIF disclosure
- Convert html tables into grid layouts
- Convert attributes/control tags into UIF input fields
- Convert image submit tags into action buttons

The existing script for KRAD conversion is included in KRADConversion.groovy. The script will process lookup, inquiry, and MaintenanceDocumentEntries into KRAD-compliant XML files.