

KRAD Tooling Projects - RDV

Rice Dictionary Validator

The Rice Dictionary Validator will be used by application developers to verify they have correctly configured an XML bean definition. This will apply both for the Data Dictionary and UI Dictionary. Examples types of validations the validator will perform are:

- Required Properties
- Invalid Property Values
- Invalid Property Combinations

The validator will output zero or more errors with information on where they were found and if possible how to correct them.

Usage Scenarios

1. Dictionary configuration is validated on startup of the application. A Rice configuration parameter can be used to turn startup validation on/off.
2. Developers can run validation without starting up the application by running validator as Java main or through Rice Tools GUI.
3. Developers can validate an individual XML file by running validator as Java main or through Rice Tools GUI. (Note this will require also loading core KRAD XML files and any core Module XML files)

Requirements

Component/Class Rule Listing

Note the following rules should apply to all subclasses as well unless explicitly noted otherwise.

**Highest priority to implement (blocker)

* Major priority to implement

No stars - lowest implementation priority

Severity key:

Bold - resulting message output to dev should be an error

Non-bold - resulting message output to dev should be a warning

org.kuali.rice.krad.uif.field.DataField

1. ****propertyName is required**
2. *Both the defaultValue and defaultValues property must not be given (0 or 1 only)
3. *If propertyEditor given, must be a valid class on the classpath
4. **if applyMask is true, maskFormatter must have a value**

org.kuali.rice.krad.uif.view.View

1. *entryPageld must be a valid id for one of the top level components in the view's items list
2. **viewStatus property should not be specified**
3. ****defaultBindingObjectPath must be a valid property (with a valid getter) on the specified form class (formClass property)**
4. ****when singlePageView is true, items cannot contain PageGroups**
5. *when singlePageView is false, items must be PageGroups
6. **when singlePageView is true, page cannot be null**
7. **if growlsMessagingEnabled is true, growls must not be null**
8. items should contain at least 1 item when singlePageView is false

org.kuali.rice.krad.uif.component.BindingInfo

1. can't think of any

org.kuali.rice.krad.uif.component.ComponentBase

1. ****id cannot contain ' or " or [] or . or # in its value**
2. **if selfRendered is false, template must have a value**
3. Warn the following: DO NOT use progressiveRender and conditionalRefresh on the same component unless it is known that the component will always be visible in all cases when a conditionalRefresh happens (ie conditionalRefresh has progressiveRender's condition anded with its own condition). If a component should be refreshed every time it is shown, use the progressiveRenderAndRefresh option with this property instead.
4. ****progressiveRender must contain the @{} springEL format**

5. ****conditionalRefresh must contain the @{} springEL format**
6. ***progressiveRender and conditionalRefresh cannot include calls to springEl functions other than # empty() at this time – because these 2 properties only allow a subset**
7. if progressiveRenderViaAjax or progressiveRenderAndRefresh are true, progressiveRender MUST have a value
8. *methodToCallOnRefresh and setDataOnRefresh should only have a value if progressiveRenderAndRefresh or refreshedByAction or progressiveRenderViaAjax is true, refreshTimer has a value, or conditionalRefresh has a value
9. *colSpan and rowSpan should only be set when this Component is used as an item of Uif-GridGroup or child of Uif-GridGroup
10. need to warn if overriding instead of merging cssClasses on beans that have Uif- beans as a parent (don't know if possible to check)
11. I'm guessing that both disableSessionPersistence and forceSessionPersistence cannot be true at the same time
12. ***if selfRendered is true, renderedHtmlOutput must have a value**

org.kuali.rice.krad.uif.component.PropertyReplacer

1. ****condition must follow SpringEL syntax @{} etc**
2. propertyName, condition, and replacement-ref are all required

org.kuali.rice.krad.uif.container.CollectionGroup

1. collectionObjectClass must be specified (unless collection is abstract)

org.kuali.rice.krad.uif.container.ContainerBase

1. **layoutManager must be set**
2. warn if both instructionalText and instructionalMessage object's messageText are set (object's value will always override)

org.kuali.rice.krad.uif.container.Group

1. ****Group cannot contain PageGroup or NavigationGroup in its items**

org.kuali.rice.krad.uif.container.PageGroup

1. ****PageGroup cannot contain PageGroup or NavigationGroup in its items**

org.kuali.rice.krad.uif.control.CheckboxControl

1. When used in InputField warn if both label and checkboxLabel are set

org.kuali.rice.krad.uif.control.ControlBase

1. warn when tabIndex is set as this can cause problems with accessibility and page flow

org.kuali.rice.krad.uif.element.Action

1. ***actionLabel and/or actionImage must have a value**
2. only 1 jumpTo attribute should be set

org.kuali.rice.krad.uif.element.AjaxAction

1. only 1: refreshId or refreshPropertyName should be set

org.kuali.rice.krad.uif.element.PageValidationMessages

1. warn if displayMessages is false at the page level – not recommended user experience

org.kuali.rice.krad.uif.element.Header

1. ***allowed values for header level are (case-insensitive) h1, h2, h3, h4, h5, h6, label**
2. header text must be set
3. lowerGroup should not be null when header.items are set. Or, initialize lowerGroup when there are items

org.kuali.rice.krad.uif.element.Iframe

1. ***source must be set**

org.kuali.rice.krad.uif.element.Image

1. ***source must be set**
2. altText should be set, warn that it violates accessibility standards if not set

org.kuali.rice.krad.uif.element.Label

1. ****if render == true, labelText should have a value**

org.kuali.rice.krad.uif.element.Link

1. ****linkText and href must be set**

org.kuali.rice.krad.uif.element.Message

1. messageText must have a value – warn

org.kuali.rice.krad.uif.field.ActionField

1. action should be non-null
2. if label is null, warn that they should use Action instead

org.kuali.rice.krad.uif.field.AttributeQuery

1. one (not sure if multiple) should have a value: dataObjectClass, queryMethodToCall, queryMethodInvokerConfig

org.kuali.rice.krad.uif.field.InputField

1. ***control must be non-null**

org.kuali.rice.krad.uif.field.LinkField

1. ***link must be non-null**
2. if label is null, warn that they should use Link instead

org.kuali.rice.krad.uif.field.MessageField

1. ***message must be non-null**
2. if label is null, warn that they should use Message instead

org.kuali.rice.krad.uif.field.SpaceField

1. if label is null, warn that they should use Space instead

org.kuali.rice.krad.uif.layout.TableLayoutManager

1. ***if rowDetailsGroup is set, richTable must not be null and richTable.render must be true**

org.kuali.rice.krad.datadictionary.state.StateMappingBase

1. states must be set when not abstract (used on a View or DataDictionaryEntry)
2. **statePropertyName must be set**

org.kuali.rice.krad.datadictionary.validation.constraint.CaseConstraint

1. **whenConstraint must include at least 1 whenConstraint**

org.kuali.rice.krad.datadictionary.validation.constraint.WhenConstraint

1. ***constraint cannot be null**
2. **values or valuePath cannot be null**

org.kuali.rice.krad.datadictionary.validation.constraint.MustOccurConstraint

1. **prerequisiteConstraints or mustOccurConstraints cannot both be empty/null**
2. **max must be greater than 0**

org.kuali.rice.krad.datadictionary.validation.constraint.FixedPointPatternConstraint

1. **precision must be greater than scale**

org.kuali.rice.krad.datadictionary.validation.constraint.CharsetPatternConstraint

1. **validChars must be set**

org.kuali.rice.krad.datadictionary.validation.constraint.ValidCharactersConstraint

1. children of this class must return a value when getValue is called

org.kuali.rice.krad.datadictionary.validation.constraint.BaseConstraint

1. ***BaseConstraints defined in the constraintStateOverrides list MUST have their states property set**
2. warn if labelKey is not set on non-abstract children

Outstanding Issues

1. Can we validate a single XML file? That is, can we do any validation without loading the whole container
2. How should we deal with expressions? These cause the original property configuration to get lost and are not replaced until a view is requested