Kuali Rice 2.3.0-M4-SNAPSHOT Release Notes

Released: 07-01-2013

Table of Contents

Overview

Welcome to Rice 2.3.0-M4-SNAPSHOT!

Release Highlights

This version of Rice includes many improvements to the Kuali Rapid Application Development (KRAD) Framework. Below is a summary of the larger new features and you can learn more about them in the KRAD Guide which is also now publicly available with this release. Also review the Jira list at the end of this document to see the smaller items.

Highlights of this release include:

- Performance Improvements
- Transactional Document Support

- Tooltip
- State-Based Validation
- Help Framework
- Question Framework
- Collections Redesign
- Messages From Central Repository
- Rich Messages
- Legend Tag for Fieldsets
- Validation Message Refactoring for Accessibility
- Ajax Improvements
- Tooling: Rice Data Object (RDO), Rice Dictionary Validator (RDV)
- Numerous bug fixes and minor improvements

Download

Kuali Rice 2.3.0-M4-SNAPSHOT can be downloaded from the Rice website at http://kuali.org/rice/download.

There are three different distributions of Rice available: source, binary and server. Please read the Installation Guide for more details on each of these distributions.

Applications can also consume Rice from the maven site at http://search.maven.org/#search|ga|1|org.kuali.rice.

Documentation

API Documentation can be found at http://site.kuali.org/rice/2.3.0-M4-SNAPSHOT/apidocs/index.html

Formal documentation can be found at http://site.kuali.org/rice/2.3.0-M4-SNAPSHOT/reference/html/index.html. This documentation is still in the process of review and update which will continue through subsequent releases, so please follow the notes in each document to report any outdated information.

Contact

If you encounter any difficulty, please don't hesitate to contact the Rice team on our public collaboration mailing list at `<rice.collab@kuali.org>`. Please indicate that you are using the 2.3.0-M4-SNAPSHOT version of Rice.

Upgrade Guide: Rice 2.1 to Rice 2.2

Caution

We recommend backing up your database before performing any upgrade!

Upgrading the server database

MySQL and Oracle scripts for performing these updates are available in the following locations in the Rice distribution downloads:

MySQL `scripts/upgrades/2.1 to 2.2/final/update_final_mysql.sql`

Oracle `scripts/upgrades/2.1 to 2.2/final/update_final_oracle.sql`

Bean Conversion Script

With the 2.2-m1 release are two scripts that will automatically convert changed bean names, property names, and CSS classes. These scripts are located in the Rice project within folder: scripts/bean-rename These are groovy scripts. To run a groovy script:

1. Download groovy at: http://groovy.codehaus.org/Download?nc
2. Follow the install instructions at: http://groovy.codehaus.org/Installing+Groovy
3. Change into the directory which contains the script, then run with 'groovy scriptname.groovy'

The first conversion script is named BeanRename.groovy and renames bean ids and property names. This will find all XML files in the directory from which the script is run, and all sub-directories. Therefore you should first copy BeanRename.groovy and the file beanReplacements.txt to the directory which contains your UIF XML files, then run from that directory. **Please note if you have XML files that contain KNS definitions (such as the LookupDefinition or MaintainbleDefinition), this conversion script should not be run on those files.** The second script is named CssRename.groovy and renames CSS classes. This will find all CSS files in the directory from which the script is run, and all sub-directories. Therefore you should first copy CssRename.groovy and the file cssReplacements.txt to the directory which contains your CSS file, then run from that directory.

Items Addressed in Rice 2.3.0-M4-SNAPSHOT

Bug Fix

- [KULRICE-5171] - Lookup not returning supporting attributes
- [KULRICE-7202] - Create new Responsibility Route Name Node has no front or back end validation.
- [KULRICE-8488] - Role Service does not notify that a derived role type service was not found
- [KULRICE-8538] - Responsibility Required routing flag not working unless responsibility defined for exact document
- [KULRICE-8604] - Travel Account Maintenance Document, Parameter Maintenance Document, and Routing Rule Delegation in sampleapp DB Incident Reports on viewing.
- [KULRICE-8641] - Lock error is not shown
- [KULRICE-8742] - When summary title has an expression the title does not render
- [KULRICE-8747] - Table layout manager rowDetailsLinkName property not getting used
- [KULRICE-8803] - Inquiry bases logic on readonly for building inquiry or direct inquiry based on the readonly flag of the Inquiry widget, not the field it applies to

- [KULRICE-8805] - Method arguments for field queries are not getting sent with script call
- [KULRICE-8806] - Editing of value with custom editor does not invoke editor on post
- [KULRICE-8807] - Link component should have a property for displaying in lightbox
- [KULRICE-8808] - Rich messages not getting picked up on image caption or cutline text
- [KULRICE-8815] - Move lookup actions to collection group's action column
- [KULRICE-8821] - Refresh timer throws exception if refresh method to call not set
- [KULRICE-8822] - Setter methods for event script not on ScriptEventSupport
- [KULRICE-8976] - Missing ValidationPattern error message key
- [KULRICE-8980] - 'Add all' option for lookup criteria field does not work with rich options
- [KULRICE-8981] - #fp binding prefix does not working correctly on fields that bind to a map
- [KULRICE-8982] - Lookup result group gets initialized twice causing table details to get added twice
- [KULRICE-8998] - Change Lookups to use Uif-DataFields in resultField List
- [KULRICE-9023] - 2.3 KIMServiceLocatorInternalTest.testGetKimTypeService_KimType expected:DataDictionaryTypeServiceBase but was:class $Proxy141
- [KULRICE-9053] - Cannot perform multiple KRAD document actions with certain browsers
- [KULRICE-9055] - "id was null" exceptions for group and role lookups
- [KULRICE-9074] - Dialog response inputs can stomp on each other
- [KULRICE-9097] - Don't display the common section on KRMS Term and TermSpecification maintenance documents
- [KULRICE-9138] - None of the Lookups and inquiry screens are working on the KRAD merged trunk
- [KULRICE-9167] - Client Validation Fails after a save and subsequent save when no errors
- [KULRICE-9218] - Components for pages not displayed not getting ids causes state issues
- [KULRICE-9247] - Data object metadata service throws exception when it encounters interface/abstract type
- [KULRICE-9250] - Revert links for converted screens to the KNS versions
- [KULRICE-9264] - ant build.xml file not finding rice-web/pom.xml artifact

Improvement

- [KULRICE-6932] - ModuleServiceBase needs to be moved out of the rice "impl" module
- [KULRICE-7562] - Improve subset of SpringEL - that we convert for specific functionality to js - by allowing a contains (or similar) operation
- [KULRICE-8169] - KRMS Agenda Editor: Selection removed on proposition description focus during edit
- [KULRICE-8640] - Create new widget to highlight code syntax

- [KULRICE-8748] - Set TableLayoutManager applyDefaultCellWidths to false by default
- [KULRICE-8818] - Provide a way for components to add data generically to the form
- [KULRICE-8819] - Add ByteArrayMultipartFileEditor to registered property editors to allow file types to bind to Byte[] data types
- [KULRICE-8897] - Need setViewHelperService method on View component
- [KULRICE-8927] - Modify DataObjectType so that instances are always created via a static method and are cached as well on the backend
- [KULRICE-8941] - Add flag to StackedLayoutManager to indicate that actions must be added to line group and not linegroup footer
- [KULRICE-9060] - add property "additionalCssClasses" to replace "cssClasses" list-merge
- [KULRICE-9073] - Add horizontal and vertical scrolling to sampleapp code editor
- [KULRICE-9087] - Ensure that our out-of-the-box persistence providers are throwing Spring DataAccessException where appropriate
- [KULRICE-9114] - DocumentBase has unnecessary overrides of refresh and refreshReferenceObject
- [KULRICE-9116] - Deprecate BusinessObject classes
- [KULRICE-9208] - isValidLine flag not available in ViewHelperServiceImpl.processAfterAddLine() method
- [KULRICE-9228] - Allow method to call to not be specified on URL
- [KULRICE-9232] - Add view TopGroup (a group that appears above breadcrumbs)
- [KULRICE-9261] - All request using multi-part encoding and should not

New Feature

- [KULRICE-5804] - Ability to configure results limits differently for different lookups needs to be added to KRAD - it was contributed for KNS in KULRICE-5686
- [KULRICE-8919] - Create krad-data module
- [KULRICE-8921] - Implement ProviderBasedDataObjectService and ProviderRegistry
- [KULRICE-8923] - Implement OJB version of PersistenceProvider
- [KULRICE-8924] - Implement OJB version of MetadataProvider
- [KULRICE-9066] - Implement JPA version of MetadataProvider
- [KULRICE-9168] - Add js function to show group content in tooltip
- [KULRICE-9193] - Implement cross-platform custom JPA id generation
- [KULRICE-9220] - Implement facility for preventing legacy persistence framework use at runtime

Task

- [KULRICE-5389] - Lookup - Implement lookup options

- [KULRICE-5390] - Lookup - Implement search field conversions
- [KULRICE-7991] - Refactor client side state handling
- [KULRICE-8150] - Upgrade Selenium 1 Smoke Tests to Selenium 2 (WebDriver)
- [KULRICE-8690] - Uif-BaseTheme-parentBean defaults to minimized js which are difficult to debug
- [KULRICE-8827] - Dialog methods in UifControllerBase should not throw exception
- [KULRICE-8828] - Dialog group bean definition should force session presistence
- [KULRICE-8868] - Page Titling Redesign
- [KULRICE-8874] - Breadcrumbs with Sub-navigation
- [KULRICE-8875] - Unique bookmarkable URLs for pages (in the left nav or tabs)
- [KULRICE-8877] - UI boilerplate elements - make always in view (left nav, buttons, breadcrumbs), while others scroll away or collapse
- [KULRICE-8917] - Iterate on validation framework UI
- [KULRICE-9037] - Code Report
- [KULRICE-9038] - Lookup - execute search when hitting enter while in any of the criteria text fields
- [KULRICE-9040] - Lookup - add triggerOnChange property on LookupCriteriaInputField that will trigger search on change of that field
- [KULRICE-9121] - Code Report
- [KULRICE-9124] - Code Report
- [KULRICE-9126] - README.txt in project root has out of date information for launching from Eclipse
- [KULRICE-9127] - Convert Address Type to KRAD
- [KULRICE-9128] - Convert Affiliation Type to KRAD
- [KULRICE-9130] - Convert Citizenship Status to KRAD
- [KULRICE-9131] - Convert Email Type to KRAD
- [KULRICE-9132] - Convert Entity Type to KRAD
- [KULRICE-9133] - Convert External Identifier Type to KRAD
- [KULRICE-9134] - Convert Name Type to KRAD
- [KULRICE-9135] - Convert Phone Type to KRAD
- [KULRICE-9136] - Convert Role/Group/Permission/Responsibility Type to KRAD
- [KULRICE-9175] - Code Report
- [KULRICE-9176] - Code Report
- [KULRICE-9177] - Code Report

- [KULRICE-9184] - Add 'order by' to CriteriaLookup
- [KULRICE-9197] - KRAD 2.2.0 Maintenance Document: Not indicating changed fields
- [KULRICE-9209] - Convert Employment Status to KRAD
- [KULRICE-9211] - Convert Employment Type to KRAD
- [KULRICE-9216] - Code Report
- [KULRICE-9217] - Code Report
- [KULRICE-9224] - QuickFinder without Lightbox
- [KULRICE-9227] - Correct spelling of suppressed
- [KULRICE-9236] - Code Report
- [KULRICE-9237] - Restructure sections on clustering and session document service
- [KULRICE-9239] - Correct KRAD Guide on using UIF functions
- [KULRICE-9240] - UifBooleanEditor Property Editor formatted string values
- [KULRICE-9245] - Code Report
- [KULRICE-9246] - Code Report
- [KULRICE-9254] - Add striping to the tables for easier readability
- [KULRICE-9267] - Change build process so generated files are placed under /target
- [KULRICE-9269] - Code Report
- [KULRICE-9278] - Update documentation per 2.3 restructure for KRAD split
- [KULRICE-9293] - JavaScript error on QuickFinder inside Collections
- [KULRICE-9294] - Active indicator is not set properly on maintenance documents

Library Upgrades

- JQuery updated to version 1.8.23
- dataTables upgraded to version 1.9

Impacting Changes

KRAD UIF configuration (XML)

The following changes impact KRAD UIF configuration (XML). Mostly these should be covered by the Bean/CSS conversion script documented in the Upgrade Guide section of this document.

Pass title property through to tag attribute on all components

Previously the component title property was be used for various purposes. For example on Containers the title was used as a shorthand for setting header.headerText. The title property should be getting passed

to the template and used to populate the title attribute on the corresponding tag (for example on a group this would be the div). Therefore the change was made to pass this through and add to the tags. The container property 'headerText' was added as a shorthand for setting the header.headerText property. The bean conversion script converts title to headerText. However it must be used with care since this should only be converted on containers (view and group), and not other components (like image or link).

Cleanup of Component Property Names

Many component property names were changed to better reflect their purpose. The bean conversion script will take care of renaming the properties in XML configuration. For a full listing of changed property names, see the project file 'beanReplacements.txt'.

Add Upper, Right, and Lower groups to Header

To provide more flexibility for adding header content, three groups were added to the Header component. These groups correspond to their render position relative to the h tag (header text). The first is named upperGroup and will render above the header text. The second is named rightGroup and will render to the right of the header text. The third is named lowerGroup and is rendered below the header text. One or more of these groups can have content for a single header.

The previous Header group named headerGroup has been replaced by the lowerGroup. The bean conversion script will convert any configuration on headerGroup to lowerGroup. If the content was intended to align to the right of the header text, this will need to be manually changed to rightGroup.

Framework Improvements - Introduction of Content Element

Previously all HTML content was rendered through a Field component. The field wrapped the content with a span and could also render a label. For some content types this did not make sense (for example iframe, label, and header). In addition, for others it might be useful to have a label, but general not needed (link, action, image). Therefore a component type named ContentElement was introduced. A content element corresponds directly with the HTML content tag without additional wrapping. For the first set of fields described above (iframe, label, header), a content element was created and the field was removed. For the second set content elements were created, but the field component remains.

Mostly the impact of this was taken care of by the base bean definitions. In some cases the actual bean id changed which is handled by the bean conversion script. In cases where the field support remains but there is now a content element, application XML should be reviewed for places where it makes sense to change to the content element.

CSS Rework

An effort begin towards the end of the 2.0 cycle to rework the KRAD CSS code. This began with implementing a naming convention and standard for applying style classes to components. This work was completed in 2.2 M1 with the rework of the style sheets. This included rewriting styles to correspond with the new classes and removing any old code. Changed style names within the UIF XML can be converted using the CssRename.groovy script. Custom style sheets will need to be updated manually.

Framework Improvements - Validation Message Architecture

Validation message handling has changed significantly in 2.2 M1. The new architecture addresses many issues, primary among these being accessibility. The ErrorsField component has been dropped and replaced with the ValidationMessages component (with subclasses PageValidationMessages, GroupValidationMessages, and FieldValidationMessages). Many properties of ErrorsField are no longer available. These changes include:

- All original properties other than displayMessages, displayFieldLabelWithMessages (now section/page level only), and additionalKeysToMatch have been eliminated as they no longer serve a purpose in the new framework. **Remove these properties from beans if in use**.
- 2 new properties were introduced:
  - collapseAdditionalFieldLinkMessages (section/page level only) - for collapsing fieldLink messages in a summary for messages beyond the first for section level summaries. KRAD and UX recommended default is true.
  - useTooltip (field level only) - for using tooltip to display validation messages. KRAD and UX recommended default is true.
- The way messages are displayed on the page and user interaction with these messages were changed heavily. These changes include:
  - Validation messages are summarized at the page level with links to sections which contain the messages
  - Validation message text for fields are links to those fields which will cause those fields to be focused
  - Validation message text is shown in tooltips for fields when focused or hovered over
  - When server errors are attempted to be fixed by the user, the field is marked as modified (if no client error persists)
  - When displayMessages flag switched to false, the framework will attempt to show the message content in the "next" available message area
  - When a user tries to submit a page that still has client-side errors the new summaries will appear
  - When a page has server side messages these new sumaries will appear
  - Various highlighting and color styling changes associated with the framework
- See the Validation Framework demo view in the sample app to get a full demonstration of all new features introduced

Some property names have changed which is handled by the bean conversion script.

UI Framework - Online Help architecture

The help URL on the document and lookup level has been removed from the KEW document type. This means the "Help Definition URL" and the "Document Search Help URL" on the "Document Type" are depreciated and don't serve any function anymore. Instead the help URL are defined on the UIF-View level either within the data dictionary or the system parameters.

Combine Inquiry and DirectInquiry

In 2.0, the Inquiry and DirectInquiry were separate widgets with separate properties on InputField. Therefore, when both were required both had to be configured with similar properties. This caused duplicate configuration. In 2.2, the DirectInquiry widget has been removed and the Inquiry widget supports rendering the direct inquiry when the field is not read-only. The direct inquiry behavior can be disabled with the Inquiry property enableDirectInquiry.

Any configuration referencing fieldDirectInquiry will need to be converted to the property named inquiry. Note if the standard inquiry and direct inquiry were both configured for a field, the direct inquiry can

simply be dropped. If you were disabling the direct inquiry by fieldDirectInquiry.render="false", this can be changed to inquiry.enableDirectInquiry. The bean conversion script makes this change.

Growls Processing

The process for adding growl messages server side has been modified. In 2.0, a growl was shown for each message in the GlobalVariables.getMessageMap() that was of type Info. The MessageMap has been enhanced to support a new growl type. For adding growls you now should call one of the following methods:

```
public void addGrowlMessage(String growlTitle, String messageKey);
public void addGrowlMessage(String growlTitle, String messageKey, String... messageParameters);
public void addGrowlMessage(String growlTitle, String growlTheme, String messageKey);
public void addGrowlMessage(String growlTitle, String growlTheme, String messageKey, String...
 messageParameters);
```

No changes were made for adding growls client side (still call the method showGrowl(message, title, theme))

Allow controller to return one way regardless of component refresh or full build

In Rice 2.0, a controller method needed to be built based on whether it would be called as part of a component refresh, or a full request. This was confusing as it required knowledge of the different refresh return method and prevented the method from being used for both types of calls. In 2.2, the controller methods no longer need to make this distinction. They can simply return the standard getUifModelAndView call for both component refresh and full requests.

Any controller methods that were returning updateComponent(uiTestForm, result, request, response) should be replace the return call with getUIFModelAndView(form).

Default action field prototype on Table layout manager to horizontal

The default layout for the action field group rendered by a table layout manager was changed from vertical box to horizontal box. This means for table with multiple line actions, the actions will now render in a horizontal row instead of vertical. If it is desired to have the actions align vertically, the actionFieldPrototype can be overridden.

Cleanup of UIF Footer Beans

In Rice 2.0 the Uif-FormView bean contained a footer defined that included the save, close, and cancel actions. The Uif-FormView bean has been changed to not include any footer actions by default. For these common actions, a bean named 'Uif-FormFooter' was created that can be used for a view or page when needed.

Ajax framework improvements.

Renamed clientSideJs in Action.java to actionScript. Any client side javascript that the user needs to be executed will now have to be set using the "actionScript" property instead of "clientSideJs".

handleIncidentReport() in krad.message was changed so that it no longer replaces the view with the incident report. It checks if the content is incident report view and then returns true else it returns false. The view is now replaced by the updateViewHandler() in krad.ajax.

ajaxSubmitForm(methodToCall, successCallback, additionalData, elementToBlock, errorCallback) has been changed to ajaxSubmitForm(methodToCall, successCallback, additionalData, elementToBlock , preSubmitCall). This in turn calls the ajaxSubmitFormFullOpts(methodToCall, successCallback, additionalData, elementToBlock, errorCallback, validate,preSubmitCall, returnType).

The following new methods have been added to krad.ajax.js

1. actionInvokeHandler(component) - Instead of calling the jQuery('#kualiForm').submit(), it has been replaced by the actionInvokeHandler(component). Component is the element on which the action has been invoked. This method checks based on the data-attributes whether it is an ajax submit or a non ajax one and then calls one of the submit methods.

2. ajaxSubmitForm() - This in turn calls the ajaxSubmitFormFullOpts() with the validate flag set to false.

3. validateAndAjaxSubmitForm() - This in turn calls the ajaxSubmitFormFullOpts() with the validate flag set to true.

4. ajaxSubmitFormFullOpts() - Submits the form through an ajax submit, the response is the new page html runs all hidden scripts passed back (this is to get around a bug with premature script evaluation). It is similar to the old ajaxSubmitForm() but has some additional parameter which allow for providing hooks for successCallback,errorCallback and preSubmitCalls. It also takes a validate flag as well as a returnType. A returnType is used to request data from the server but the server may override it. If the validate flag is set it validates the form and proceeds if the form is valid. If a presubmit call is specified then it executes that and proceeds if it returns true. If the returnType is not given then it defaults to "update-page" and sets it on the data which will be submitted to the server. It then calls the invokeAjaxReturnHandler() to determine which handler function to call. The successCallback and errorCallback are handled as they were before in ajaxSubmitForm(). The elementToBlock and the lightBox processing remain the same.

5. submitForm() - This is used for non-ajax calls. This in turn calls the submitFormFullOpts() with the validate flag set to false.

6. validateAndSubmitForm() - This is used for non-ajax calls. This in turn calls the submitFormFullOpts() with the validate flag set to true.

7. submitFormFullOpts() - Does a non ajax submit. The data-attributes that are passed in as additional data are written as hidden params to the form before it is submitted.

8. invokeAjaxReturnHandler() - This method iterates over divs in the content that is passed in to determine which handler functions to call. The handler functions are initialized in krad.initialize.js

9. updatePageHandler() - Called if the returnType is "update-page". Finds the page content in the returned content and updates the page, then processes breadcrumbs and hidden scripts. While processing, the page contents are hidden (works similar to the the updatePageCallback()).

10. updateViewHandler() - Replaces the view with the given content and runs the hidden scripts. Called when the returnType is "update-view".

11. redirectHandler() - Called when the returnType is "redirect". Replaces the contents of the window with those of the redirected URL.

12. updateComponentHandler() - Called when the returnType is "update-component". Retrieves the component with the matching id from the server and replaces a matching _refreshWrapper marker span with the same id with the result. In addition, if the result contains a label and a displayWith marker span has a matching id, that span will be replaced with the label content and removed from the component. This allows for label and component content seperation on fields (Works the same as updateRefreshableComponentCallback)

Changes and removal of some methods in DictionaryValidationService (KRAD version) and ViewValidationService as a result of State based validation

Please see these classes for the changes made here. Some method signatures were changed for this support and some methods were removed for clarity.

KULRICE-7485 - Refactoring of Session Form Handling

Previously to clear (or reset) form properties between requests the postBind form method could be implemented. This method has been removed in M2. Now to indicate a property should not be persisted in session (thus reset for each request) the annotation @SessionTransient should be added before the property declaration.

KULRICE-7527 - Removal of some default nested beans

To help with performance, some nested beans that were being initialized in the XML were pulled out. In the code, if determined the component is needed, a call is made to then initialize the bean.

This means for these removed beans, if you were using the nested notation to set a property (for example, 'nested.property'), this will now throw an exception. You must first initialize the nested bean, then set the property. Note in cases where the default initialization was removed, a top level bean was created making it easy to initialize if needed.

The following nested beans were removed:

- Uif-InputField - constraintMessage, instructionalMessage, and suggest property
- Uif-Image - captionHeader and cutlineMessage property
- Uif-Header* - upperGroup, rightGroup, and lowerGroup properties
- All containers (Uif-Group, sections, subsections) - instructionalMessage property

Example Conversion

```
<bean parent="Uif-VerticalBoxSection">
    <property name="header.upperGroup.items">
        <list>    ...   </list>
    </property>
</bean>
```

change to

```
<bean parent="Uif-VerticalBoxSection">
    <property name="header.upperGroup">
        <bean parent="Uif-HeaderUpperGroup">
            <property name="items">
                <list>          ...       </list>
            </property>
        </bean>
    </property>
</bean>
```

KULRICE-7351 - JSP Templates converted to FreeMarker

For performance reasons the KRAD JSP templates were converted to use the FreeMarker templating language. Any custom components (or template overrides) that were created will need to be converted to FreeMarker.

See this page for a comparison of JSP and FreeMarker: https://wiki.kuali.org/display/KULRICE/Freemarker+and+JSP+comparison

Templates are associated with a component (bean) using two properties now. The first is the template property which gives the location of the template source. The second is the templateName property which is the name of the template to invoke (in FreeMarker this is the macro name). If you override one of the default templates, make sure to also give the template a unique templateName.

KULRICE-7246 - Default escapeHtmlInPropertyValue to True

For security reasons the escapeHtmlInPropertyValue property on DataField was defaulted to true. Therefore if you have a property which has HTML that should be rendered, you will need to set this property to false on the data field.

Preloading of View Objects

Loading the view object from bean definitions can take up to 4 or 5 seconds in some cases (although this is improving with the removal of default nested beans). To boast performance, an enhancement was put in to 'pre-load' the view objects. This means when the application starts up, it will fetch one or more view objects for a particular view and hold until a request is made. The number of views to preload is configured by the view property preloadPoolSize. This is set to 1 by default for all views except lookups and inquiries.

Setting Preload Pool Size

```
<bean id="CourseView" parent="Uif-FormView">
    ...
    <property name="preloadPoolSize" value="3"/>
</bean>
```

The views are loaded in a separate thread so the startup time is not impacted. Whenever a request for a view is made, a view from the pool is returned and a new view object is pulled in a separate thread to replace it. If no view objects exist in the pool at the time of a request, a call is made to build the view object. For high load views you can experiment with increasing the pool size so that a view is always available (Note a log statement is made when a view is not available from the pool, thus this can be used to help tune the parameter).

Note though storing the preloaded view objects consumes memory. In Rice (with Sampleapp) the storage was 91mb (and Rice does not have many views compared to what other applications like the KFS would).

Limiting Form Storage Size

In previous versions of Rice a user can keep requested view objects whose forms get added to the user session (note this is only for views that have session persistence enabled). Although there are clear points implemented, they do not catch many exit actions. Therefore more and more memory continues to be consumed, even though the user might be long done with the form.

To prevent this problem a configuration parameter was introduced that limits the number of forms per user session. The parameter name is maxNumberOfSessionForms and is set to 5 by default. This parameter can be tuned as needed however it is not recommended to go lower than 5 (think of nested lookups that might take place from a view).

Setting Max Form Storage Size in your rice config XML:

```
<param name="maxNumberOfSessionForms">10</param>
```

Whenever the number of session forms is greater than the configured maximum allowed, a form will be removed from the session. For determining which form to remove in these situations, the forms are arranged according to when they were accessed. In other words, the form whose last requested action time is oldest will be removed (most recently accessed forms will be kept).

Database Updates

Instructions on how to run the scripts to apply these database changes are in the Upgrade Guide in this document

Assignment of "Add Message to Route Log" permission to the KR-SYS technical administrator returned to bootstrap dataset

The "Add Message to Route Log" permission assignment to the KR-SYS "Technical Administrator" role was present in the main 1.0.3.3 dataset but missing in the bootstrap dataset for 2.0.0. The sql script removes the permission from the role to make upgraded databases consistent with new ones.

DB updates for external messages

A new message table and locale system parameter was added to support external messages.

KIM role and guest user to support guest access

To support guest access a new KIM user with id 'guest' was created along with a guest role.

Apply SQL for new KRMS tables

New tables and sequences were added to the KRMS schema to support Type-Type Relations, Natural Language Translation, Reference Object Bindings features (Rice 2.x - KRMS for KS Analysis).

Cleanup of Ajax Actions

AjaxAction and its bean Uif-AjaxActionButton was removed. All bean references should change to use 'Uif-PrimaryActionButton' (ajax is the default now). The bean update script can be run to apply this change.

Property validatedLineActions on CollectionGroup was removed. Now actions that are defined in lineActions can have the property 'performClientSideValidation' flag set to true to indicate the line should be validated.

Resolving ambiguous method call in MessageMap.addGrowlMessage()

- Removed MessageMap#public void addGrowlMessage(String growlTitle, String messageKey)
- Removed MessageMap#public void addGrowlMessage(String growlTitle, String growlTheme, String messageKey)
- Removed MessageMap#public void addGrowlMessage(String growlTitle, String growlTheme, String messageKey, String... messageParameters)
- Use MessageMap#addGrowl(GrowlMessage growl)

Central Message Repository

Property labelKey of BaseConstraint (and therefore all constraints) was renamed to messageKey. The bean update script can be run to apply this change.

Property baselinePackages of DataDictionaryService was renamed to additionalDictionaryFiles. In addition changed from a List<String> to Map<String, List<String>> where the map key is the namespace the dictionary beans should be associated with.

Transactional Document Support

- Renamed MaintenanceForm to MaintenanceDocumentForm
- Renamed MaintenanceView to MaintenanceDocumentView