

Kuali Rice 2.0 Upgrade Notes - Indiana University



This is a work in progress. Cross checking with other notes is highly recommended.

- Overview
- Summary of Changes
 - 1.0.3 -> 1.0.3.1
 - 1.0.3.1 -> 1.0.3.2
 - 1.0.3.2 -> 1.0.3.3
 - 1.0.3.3 -> 2.0
- Link to KualI Source Code
- Upgrade Steps
 - Database Updates
 - Database changes due to the DOC_HDR_ID changing from NUMBER to VARCHAR2(40)
 - KNS Client Applications Database
 - Updates to KNS Database Tables
 - Web Content Updates
 - Library Updates
 - Libraries Added between Rice 1.0.3 and 2.0
 - Library Version Changes
 - Libraries Removed between Rice 1.0.3 and 2.0
 - Libraries Which Have Not Changed
 - If needed, update jquery versions
 - Impacting Changes
 - Configuration
 - Standard configuration of "embedded" client Rice projects
 - Standard configuration of "thin" client Rice projects
 - Encryption Key Update in Override Cases
 - Global Changes
 - Package Names
 - Service locator & Service Name Changes
 - Changes to UserSession
 - Changes to Constant Classes (say that 5 times fast!)
 - RouteHeaderId changed to DocumentId
 - Use of immutable objects and the builder pattern for various APIs
 - General Purpose Lookup APIs
 - RunModes (kew.mode, kim.mode, etc.) & Client Protocol (client.protocol)
 - New required listeners
 - 'Service Namespace' concept replaced with 'Application ID' concept
 - KEN and KCB Changes
 - KEW Changes
 - Workflow services
 - WorkflowAttributeDefinitionDTO has been renamed to WorkflowAttributeDefinition and is now a final class.
 - WorkflowRuleAttribute validation error type changes
 - removal of getStatusDisplayValue
 - What to use as a replacement for the usersInitiator method that used to be on KualIWorkflowDocument
 - Replacement for getActionRequests(documentNumber)
 - Addition of PeopleFlows
 - Changes to document search
 - Remaining numeric identifiers in KEW database Converted to character-based
 - KIM Changes
 - Class name changes
 - Removal of IdentityManagementService
 - Removal of KimAttributeSet
 - KNS Changes
 - Business Objects
 - Maintainables
 - Action Classes
 - Values Finders
 - Transaction Documents
 - Data Dictionary
 - Changes to Notes
 - Changes to GlobalVariables.java
 - Changes to KualI Configuration Service
 - Removal of RiceApplicationConfigurationService
 - Removal of ConfigProperties.java
 - KSB Changes
 - KSB Architecture changes
 - Package / Class Name Changes
 - Class & Method Name Changes
 - Important Package Changes
 - Miscellaneous changes / deletions that have not been mentioned above
 - Troubleshooting:

Overview

From KR 1.0.3 to KR 2.0 there have been the following versions:

- 1.0.3.1 - released February 11, 2011
- 1.0.3.2 - released June 17, 2011
- 1.0.3.3 - released September 9, 2011
- 2.0.0-b1 - released November 7, 2011
- 2.0.0-b2 - released November 21, 2011
- 2.0.0-b3 - released December 5, 2011
- 2.0.0-b4 - released December 21, 2011
- 2.0.0-b6 - released January 9, 2012 (b5 release was skipped due, b6 was released in it's place)
- 2.0.0-b7 - released January 17, 2012

Summary of Changes

1.0.3 -> 1.0.3.1

Release Notes: <https://wiki.kuali.org/download/attachments/309465683/release-notes-1.0.3.1.pdf?version=1&modificationDate=1297449754000>

This release primarily consisted of bug fixes.

1.0.3.1 -> 1.0.3.2

Release Notes: <https://wiki.kuali.org/download/attachments/311494805/rice-1.0.3.2-release-notes.pdf?version=1&modificationDate=1308328955000>

This release primarily consisted of the following items:

- Support for Tomcat 7
- Performance fixes
- Numerous bug fixes

1.0.3.2 -> 1.0.3.3

Release Notes: https://wiki.kuali.org/download/attachments/311824598/rice-1_0_3_3_release-notes.pdf?version=1&modificationDate=1315602768000

This release primarily consisted of bug fixes.

1.0.3.3 -> 2.0

B1 Release Notes: <http://site.kuali.org/rice/2.0.0-b1/reference/html/release-notes.html>

B2 Release Notes: <http://site.kuali.org/rice/2.0.0-b2/reference/html/release-notes.html>

B3 Release Notes: <http://site.kuali.org/rice/2.0.0-b3/reference/html/release-notes.html>

B4 Release Notes: <http://site.kuali.org/rice/2.0.0-b4/reference/html/release-notes.html>

B6 Release Notes: <http://site.kuali.org/rice/2.0.0-b6/reference/html/release-notes.html>

B7 Release Notes: <http://site.kuali.org/rice/2.0.0-b7/reference/html/release-notes.html>

Beta Test Home: <https://wiki.kuali.org/display/KULRICE/Rice+2.0+Beta+Testing>

rc1 Documentation: <http://site.kuali.org/rice/2.0.0-rc1/reference/html/portal.html>

rc2 Documentation: <http://site.kuali.org/rice/2.0.0-rc2/reference/html/portal.html>

Link to Kuali Source Code

[Beta](#)

[Official Releases](#)

Upgrade Steps

The upgrade process for client applications can be broken down into the following steps:

1. Database Updates
2. Web Content Updates
3. Library Updates
4. Account for impacting changes in code

Database Updates

Database changes due to the DOC_HDR_ID changing from NUMBER to VARCHAR2(40)

The DOC_HDR_ID has changed from a NUMBER to a VARCHAR2(40) in the 2.0 code. If the client application database stores DOC_HDR_ID in any tables, we strongly recommend that these data types are modified even if the only values stored for DOC_HDR_ID are numeric. However, it won't be strictly required that a client application change their data types for DOC_HDR_ID since KEW will still generate a value for document id that could be stored as a number.

KNS Client Applications Database

Updates to KNS Database Tables

Two tables have minor changes in the KNS client application database. This change could be done easily in advance of the upgrade, because it only consists of adding two columns to a table. The scripts to perform the KNS database conversion on a client application database can be found in the rice project at the following location:

- /scripts/upgrades/1.0.3 to 2.0.0/db-updates-client/update_client_final_oracle.sql

Web Content Updates

There have been various changes to the web content, which will need to be updated in KNS client applications. Only the KNS related items which have changed are below. This list consists only of items that have had meaningful changes, as there were a number of files which had only copyright or description changes.




If your project contains additional files unrelated to the KNS, it would be wise to simply replace all Kualu Rice web content files in the client application.

In /src/main/webapp/WEB-INF/, the following should NOT be replaced as they are client app specific.

- struts-config.xml
- web.xml

Location/File	Added	Changed	Removed
/src/main/webapp/WEB-INF/			
validator-rules.xml		+	
/src/main/webapp/kr/images/	Added	Changed	Removed
errormark.gif			+
errormark.png	+		
/src/main/webapp/kr/scripts/	Added	Changed	Removed
core.js		+	
objectInfo.js		+	
/src/main/webapp/kr/static/images/	Added	Changed	Removed
buttonsmall_deselectallfromallpages.png	+		

buttonsmall_deselectallfromthispage.png			
buttonsmall_selectallfromallpages.png			
buttonsmall_selectallfromthispage.png			
errormark.png			
errormark.gif			
/src/main/webapp/kr/WEB-INF/	Added	Changed	Removed
struts-config.xml			
validator-rules.xml			
/src/main/webapp/kr/WEB-INF/jsp/	Added	Changed	Removed
AuthorizationException.jsp			
KualiExceptionIncident.jsp			
KualiLookup.jsp			
KualiMaintenanceDocument.jsp			
TextArea.jsp			
tldHeader.jsp			
/web/src/main/webapp/kr/WEB-INF/tlds/ *Note - directory can also be removed	Added	Changed	Removed
c.tld			
displaytag-el.tld			
displaytag.tld			
fnt.tld			
fn.tld			
kuali-func.tld			
sql.tld			
struts*.tld			
x.tld			
/src/main/webapp/WEB-INF/jsp/	Added	Changed	Removed
dummy_login.jsp			
moduleLocked.jsp			
recurseTreeNode.jsp			
/src/main/webapp/WEB-INF/tags/kr/	Added	Changed	Removed

ALL files, copy over the entire folder			
--	--	---	--


Library Updates

 **Applies To:**
























- All applications

There are a few library changes that go along with the Rice 2.0 version. You will need to update the libraries associated with your existing project as necessary in order to react to this. Most of the libraries listed below can be found by downloading the rice 2.0 binary distribution from <http://kuali.org/rice/download/beta> (Beta) or <http://rice.kuali.org/download> (post-beta).

What follows are tables detailing libraries added, removed, and changed from Rice 1.0.3 to Rice 2.0. They also indicate which libraries are required for remote (thin client), embedded, and KNS integration. If a library isn't required in any situation, that likely means it's used only by unit testing or on the Kuali Rice Standalone server.

 **Be Careful!**
 Even if you may not need one of the libraries listed below as a Rice dependency, your application may be using it for something else. So be sure to verify whether or not you are using a particular library in your application independently of Rice before you remove it from your project.

Libraries Added between Rice 1.0.3 and 2.0

Library Added	Needed for Remote? (aka. Thin Client)	Needed for Embedded?	Needed for KNS?	Needed for KRAD usage	Notes
aspectjrt-1.6.8.jar					
aspectjweaver-1.6.8.jar					
avalon-framework-4.1.5.jar					
btm-2.1.1.jar					Only needed if using Bitronix for your connection pooling
carol-interceptors-1.0.1.jar					
commons-chain-1.2.jar	??	??	??		
cxfrt-transports-common-2.3.6.jar					
ehcache-core-2.5.0.jar					
guava-11.0.jar	??	??	??		
hibernate-core-3.6.9.Final.jar			??		
hibernate-jpa-2.0-api-1.0.1.Final.jar			??		
hsqldb-1.8.0.1.jar	??	??	??		
irmi-1.1.2.jar	??	??	??		
jackson-core-asl-1.8.7.jar	??	??	??		

jackson-mapper-asl-1.8.7.jar	??	??	??		
jacorb-2.2.3-jonas-patch-20071018.jar	??	??	??		
jacorb-idl-2.2.3-jonas-patch-20071018.jar	??	??	??		
jcl-over-slf4j-1.6.4.jar					Needed only if you use JCL
joda-time-2.0.jar					
logkit-1.2.jar	??	??	??		
serializer-2.7.1.jar	??	??	??		
slf4j-api-1.6.4.jar	??	??			
slf4j-log4j12-1.6.4.jar	??	??			
spring-asm-3.1.0.RC1.jar					
spring-tx-3.1.0.RC1.jar					
spring-expression-3.1.0.RELEASE.jar					
spring-security-cas-3.1.0.RELEASE.jar					
spring-security-core-3.1.0.RELEASE.jar					
spring-security-core-3.1.0.RELEASE.jar					
spring-security-web-3.1.0.RELEASE.jar					
stax2-api-3.1.1.jar					
struts-extras-1.3.10.jar	??	??			
struts-taglib-1.3.10.jar	??	??			
tiles-api-2.2.2.jar	??	??	??		
tiles-core-2.2.2.jar	??	??	??		
tiles-jsp-2.2.2.jar	??	??	??		
tiles-servlet-2.2.2.jar	??	??	??		
tiles-template-2.2.2.jar	??	??	??		
woodstox-core-asl-4.1.1.jar					

Library Version Changes

Old Library Version	New Library Version	Needed for Remote? (aka. Thin Client)	Needed for Embedded?	Needed for KNS?	Notes
---------------------	---------------------	---------------------------------------	----------------------	-----------------	-------

asm-2.2.3.jar	asm-3.3.1.jar				
bcprov-jdk15-1.45.jar	bcprov-jdk16-1.46.jar				The Bouncy Castle Crypto package is used for cryptographic algorithms.
carol-2.0.5.jar	carol-3.0.6.jar				
cas-client-core-3.1.6.jar	cas-client-core-3.2.1.jar				
cglib-nodep-2.1_3.jar	cglib-nodep-2.2.2.jar				
commons-beanutils-1.7.0.jar	commons-beanutils-1.8.3-kuali.jar	??	??		
commons-codec-1.3.jar	commons-codec-1.6.jar	??	??		
commons-collections-3.2.jar	commons-collections-3.2.1.jar				
commons-dbc-1.2.1.jar	commons-dbc-1.4.jar	??	??		
commons-digester-1.6.jar	commons-digester-2.0.jar	??	??	??	
commons-fileupload-1.2.1.jar	commons-fileupload-1.2.2.jar				Struts dependency
commons-httpclient-3.0.1.jar	commons-httpclient-3.1.jar				
commons-io-1.4.jar	commons-io-2.1.jar				Used by Rice standalone server
commons-lang-2.3.jar	commons-lang-2.6.jar				
commons-pool-1.2.jar	commons-pool-1.5.7.jar				Used by OJB (and others)
commons-validator-1.1.4.jar	commons-validator-1.3.1.jar				Struts dependency
cxfrt-api-2.2.9.jar	cxfrt-api-2.3.8.jar				Used for connecting to SOAP services over the KSB
cxfrt-common-schemas-2.2.9.jar	cxfrt-common-schemas-2.3.8.jar				Used for connecting to SOAP services over the KSB
cxfrt-common-utilities-2.2.9.jar	cxfrt-common-utilities-2.3.8.jar				Used for connecting to SOAP services over the KSB
cxfrt-bindings-soap-2.2.9.jar	cxfrt-bindings-soap-2.3.8.jar				Used for connecting to SOAP services over the KSB
cxfrt-bindings-xml-2.2.9.jar	cxfrt-bindings-xml-2.3.8.jar				Used for connecting to SOAP services over the KSB
cxfrt-core-2.2.9.jar	cxfrt-core-2.3.8.jar				Used for connecting to SOAP services over the KSB

cxfrt-databinding-aegis-2.2.9.jar	cxfrt-databinding-aegis-2.3.8.jar	✗	✓	✓	Used for connecting to SOAP services over the KSB
cxfrt-databinding-jaxb-2.2.9.jar	cxfrt-databinding-jaxb-2.3.8.jar	✗	✓	✓	Used for connecting to SOAP services over the KSB
cxfrt-frontend-jaxrs-2.2.9.jar	cxfrt-frontend-jaxrs-2.3.8.jar	✗	✓	✓	Used for connecting to SOAP services over the KSB
cxfrt-frontend-jaxws-2.2.9.jar	cxfrt-frontend-jaxws-2.3.8.jar	✗	✓	✓	Used for connecting to SOAP services over the KSB
cxfrt-frontend-simple-2.2.9.jar	cxfrt-frontend-simple-2.3.8.jar	✗	✓	✓	Used for connecting to SOAP services over the KSB
cxfrt-transport-http-2.2.9.jar	cxfrt-transport-http-2.3.8.jar	✗	✓	✓	Used for connecting to SOAP services over the KSB
cxfrt-ws-addr-2.1.5.jar	cxfrt-ws-addr-2.3.8.jar	✗	✓	✓	Used for connecting to SOAP services over the KSB
cxfrt-ws-security-2.1.5.jar	cxfrt-ws-security-2.3.8.jar	✗	✓	✓	Used for connecting to SOAP services over the KSB
cxftools-common-2.1.5.jar	cxftools-common-2.3.8.jar	✗	✓	✓	Used for connecting to SOAP services over the KSB
db-obj-1.0.4-patch4.jar	db-obj-1.0.4-patch8.jar	✗	✓	✓	
displaytag-1.1.jar	displaytag-1.2.jar	✗	✗	✓	
dwr-1.1.3.jar	dwr-3.0.RC2.jar	??	??	✓	In DWR 3.0 all logging is done through the commons-logging API. Please see the following website for more information: DWR Documentation
groovy-all-minimal-1.5.4.jar	groovy-all-1.8.5.jar	✗	✓	✗	
hibernate-commons-annotations-3.0.0.ga.jar	hibernate-commons-annotations-3.2.0.Final.jar	✗	✓	✓	
hibernate-entitymanager-3.3.1.ga.jar	hibernate-entitymanager-3.6.9.Final.jar	✗	✓	✓	
howl-logger-0.1.11.jar	howl-1.0.1-1.jar	✗	✓	✓	
javassist-3.3.ga.jar	javassist-3.15.0-GA.jar	✗	✗	✗	
jaxb-impl-2.1.9.jar	jaxb-impl-2.1.13.jar	✗	✓	✓	Used for service SOAP binding on the KSB

jaxb-xjc-2.1.9.jar	jaxb-xjc-2.1.13.jar	✗	✓	✓	Used for service SOAP binding on the KSB
jdom-1.0.jar	jdom-1.1.jar	✗	✓	✓	
jettison-1.2.jar	jettison-1.3.jar	✗	✓	✗	
jotm-2.0.10.jar	jotm-core-2.1.10.jar	??	??	✓	
jsr311-api-1.0.jar	jsr311-api-1.1.1.jar	✗	✓	✓	
jstl-1.1.2.jar	jstl-1.2.jar	✗	✗	✗	
jta-1.0.1B.jar	jta-1.1.jar	✗	✓	✓	
log4j-1.2.14.jar	log4j-1.2.16.jar	✓	✓	✓	
mail-1.4.jar	mail-1.4.4.jar	??	??	??	
neethi-2.0.4.jar	neethi-2.0.5.jar	✗	✗	✗	
spring-2.5.6.jar	spring-3.1.0.RELEASE.jar	✓	✓	✓	spring-3.1.0.RC1.jar can be included either as a single jar or the multiple jars listed below
spring-aop-2.5.6.jar	spring-aop-3.1.0.RELEASE.jar	✓	✓	✓	
spring-beans-2.5.6.jar	spring-beans-3.1.0.RELEASE.jar	✓	✓	✓	
spring-context-2.5.6.jar	spring-context-3.1.0.RELEASE.jar	✓	✓	✓	
spring-context-support-2.5.6.jar	spring-context-support-3.1.0.RELEASE.jar	✓	✓	✓	
spring-core-2.5.6.jar	spring-core-3.1.0.RELEASE.jar	✓	✓	✓	
spring-jdbc-2.5.6.jar	spring-jdbc-3.1.0.RELEASE.jar	✓	✓	✓	
spring-orm-2.5.6.jar	spring-orm-3.1.0.RELEASE.jar	✓	✓	✓	
spring-tx-2.5.6.jar	spring-tx-3.1.0.RELEASE.jar	✓	✓	✓	
spring-webmvc-2.5.6.jar	spring-webmvc-3.1.0.RELEASE.jar	✓	✓	✓	
spring-web-2.5.6.jar	spring-web-3.1.0.RELEASE.jar	✓	✓	✓	
struts-1.2.9.jar	struts-core-1.3.10.jar	??	✓	✓	
struts-el-1.2.9.jar	struts-el-1.3.10.jar	✗	✗	✓	
xalan-2.7.0.jar	xalan-2.7.1.jar	??	?	✓	
XmlSchema-1.4.5.jar	XmlSchema-1.4.7.jar	??	??	✗	
xmlsec-1.4.3.jar	xmlsec-1.4.4.jar	✓	✓	✓	XML Security, used for digital signatures

wss4j-1.5.8.jar	wss4j-1.5.11.jar				WS-Security, used for SOAP web services
-----------------	------------------	---	---	---	---



spring-webmvc-struts.jar

Note: As of Spring 2.5, Spring Web MVC is no longer part of the 'spring.jar' file. Spring MVC can be found in 'spring-webmvc.jar' and 'spring-webmvc-portlet.jar' in the lib/modules directory of the distribution. Furthermore, the Struts 1.x support has been factored out into 'spring-webmvc-struts.jar'.

Rice itself does not use spring-webmvc-struts.jar, but if you were including the spring-2.0.4.jar and using the spring web mvc struts functionality, spring-webmvc-struts.jar will now have to be included separately.

Libraries Removed between Rice 1.0.3 and 2.0

Library Removed	Notes
abdera-core-1.0.jar	
abdera-extensions-json-1.0.jar	
abdera-extensions-main-1.0.jar	
abdera-i18n-1.0.jar	
abdera-parser-1.0.jar	
acegi-security-1.0.6.jar	
acegi-security-cas-1.0.6.jar	
activation-1.1.jar	
antlr-2.7.6.jar	
asm-attrs-1.5.3.jar	
axiom-api-1.2.5.jar	Used by Abdera
axiom-impl-1.2.5.jar	Used by Abdera
bcel-5.1.jar	
bsf-2.4.0.jar	
casclient-2.1.1.jar	
cas-server-3.0.4.jar	
concurrent-1.3.4.jar	
derby-10.2.2.0.jar	
geronimo-activation_1.1_spec-1.0.2.jar	
geronimo-annotation_1.0_spec-1.1.1.jar	
geronimo-javamail_1.4_spec-1.3.jar	
geronimo-jaxws_2.1_spec-1.0.jar	
geronimo-stax-api_1.0_spec-1.0.1.jar	
geronimo-ws-metadata_2.0_spec-1.1.2.jar	
hibernate-3.2.6.ga.jar	
hibernate-annotations-3.3.0.ga.jar	
hibernate-validator-3.0.0.ga.jar	
jaxb-api-2.1.jar	
jaxb-xjc-2.1.13.jar	

jaxen-1.1.3.jar	
jboss-common-core-2.0.4.GA.jar	
jotm_jrmp_stubs-2.0.10.jar	
js-1.6R5.jar	
nekohtml-0.9.5.jar	
oscache-2.3.2.jar	replaced by ehcache-core-2.5.0.jar
persistence-api-1.0.jar	
saaj-api-1.3.jar	
saaj-impl-1.3.2.jar	
spring-test-2.5.6.jar	
standard-1.1.2.jar	
wstx-asl-3.2.9.jar	
xml-apis-1.0.b2.jar	
xmlbeans-2.4.0.jar	Came in 1.0.3 as a result of the CXF upgrade
xom-1.1.jar	

Libraries Which Have Not Changed

Library	Needed for Remote? (aka. Thin Client)	Needed for Embedded?	Needed for KNS?	Notes
aopalliance-1.0	??	✓	✓	
commons-logging-1.1.jar	✓	✓	✓	
commons-transaction-1.1.jar	??	??		
connector-api-1.5.jar	??	??		
dom4j-1.6.1.jar	✗	✓	✓	
itext-1.3.jar	✗	✓	✓	
jaxb-impl-2.1.13.jar	✗	✓	✓	
oro-2.0.8.jar	✗	✗	✗	
spring-modules-ojb-0.8a.jar	✗	✓	✓	
wSDL4j-1.6.2.jar	??	??	??	
xercesImpl-2.9.1.jar	??	??	??	
xml-resolver-1.2.jar	??	??	??	
xpp3_min-1.1.3.4.O.jar	✗	✓ ?	✓	For embedded kew, required if using the EmailNode
xstream-1.2.2.jar	✗	✓ ?	✓	For embedded kew, required if using the EmailNode

If needed, update jquery versions

For those of you who have jquery items, you will need to upgrade to the following versions:

- jquery-1.6.3.js
- jquery-ui-1.8.13.css
- jquery-current-min.js

Impacting Changes

[Go back to upgrade steps](#)



Applies To:

- All applications

Configuration

Previously, all module configuration was done inside the rice bean (RiceConfigurer). This has been replaced by a modular configuration design, with the rice core being configured in a new CoreConfigurer.

- Each module has it's own configurer as well (KewConfigurer, KimConfigurer, etc..).
- Run modes are set independently for each module, and value modes can be restricted in the respective configurer bean.
- Make sure to update rice.struts.message.resources if you have it in your config file (example value below).
- Additional spring files are now included via module.additionalSpringFiles (eg. rice.kim.additionalSpringFiles)
 - **Note:** Each module is loaded into its own Spring context now so you will need to make sure you import your beans into the relevant Spring context with this rice.<module>.additionalSpringFiles parameter
 - **Note:** The KRAD and KNS modules both share the same Spring context and use the rice.kr.additionalSpringFiles parameter
 - **Note:** The configuration parameters which use the module name are case sensitive and must be lower case! The exception to this is the configuration parameter for KRAD and KNS, which must be in uppercase.

Common config items which have new names:

1.0.3 Name	2.0 Name
ConfigProperties.css.files	ConfigProperties.kns.css.files
ConfigProperties.javascript.files	ConfigProperties.kns.javascript.files
rice.kim.additionalSpringFiles	rice.kim.additionalSpringFiles rice.kew.additionalSpringFiles rice.coreservice.additionalSpringFiles rice.location.additionalSpringFiles rice.KR.additionalSpringFiles

Standard configuration of "embedded" client Rice projects

```
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:p="http://www.springframework.org/schema/p"
  xmlns:util="http://www.springframework.org/schema/util"
  xsi:schemaLocation="http://www.springframework.org/schema/beans
  http://www.springframework.org/schema/beans/spring-beans-3.1.xsd
  http://www.springframework.org/schema/util
  http://www.springframework.org/schema/util/spring-util-3.1.xsd">

  <!-- DATASOURCE CONFIGURATION HERE (short example, others removed for space)
-->
  <bean id="krWorkflowDataSource"
class="org.kuali.rice.core.framework.persistence.jdbc.datasource.XAPoolDataSource">
    <property name="transactionManager" ref="jotm" />
```

```

        <property name="driverClassName" value="\${eden.datasource.driver.name}" />
        <property name="maxSize" value="\${kr.datasource.pool.maxSize}" />
        <property name="minSize" value="\${kr.datasource.pool.minSize}" />
        <property name="maxWait" value="\${kr.datasource.pool.maxWait}" />
        <property name="validationQuery"
value="\${kr.datasource.pool.validationQuery}" />
        <property name="url" value="\${kr.datasource.url}" />
        <property name="username" value="\${kr.datasource.username}" />
        <property name="password" value="\${kr.datasource.password}" />
    </bean>

    <!-- RiceConfigurer NO LONGER EXISTS, coreConfigurer is a pseudo-replacement
-->

    <bean id="coreConfigurer"
class="org.kuali.rice.core.impl.config.module.CoreConfigurer">
        <property name="dataSource" ref="appDataSource" />
        <property name="serverDataSource" ref="krWorkflowDataSource"/>
        <property name="nonTransactionalDataSource" ref="nonTransDataSource" />
        <property name="transactionManager" ref="jotm" />
        <property name="userTransaction" ref="jotm" />
    </bean>

    <bean id="ksbConfigurer"
class="org.kuali.rice.ksb.messaging.config.KSBConfigurer"
depends-on="coreConfigurer" />

    <bean id="knsConfigurer" class="org.kuali.rice.krad.config.KRADConfigurer"
depends-on="coreConfigurer" />

    <bean id="coreServiceConfigurer"
class="org.kuali.rice.coreservice.impl.config.CoreServiceConfigurer"
depends-on="coreConfigurer" />

    <bean id="kimConfigurer" class="org.kuali.rice.kim.config.KIMConfigurer"
depends-on="coreConfigurer" />

    <bean id="kewConfigurer" class="org.kuali.rice.kew.config.KEWConfigurer"
depends-on="coreConfigurer" />

</beans>

<config>

    <param name="kim.mode">EMBEDDED</param>
    <param name="kew.mode">EMBEDDED</param>

    <param name="riceapp.url">https://location.of.rice.standalone/kr2-snd/</param>
    <param name="rice.ksb.registry.serviceUrl"
override="false">\${riceapp.url}/remoting/soap/ksb/v2_0/serviceRegistry</param>
    <param name="rice.ksb.serviceRegistry.security" override="false">>true</param>
    <param name="rice.ksb.config.allowSelfSignedSSL" override="false">>false</param>

    <param
name="rice.kim.additionalSpringFiles">classpath:edu/iu/uis/rice/resources/IUKimOverrid
eSpringBeans.xml</param>
    <param
name="rice.kew.additionalSpringFiles">classpath:edu/iu/uis/rice/resources/IUKewOverrid

```

```
eSpringBeans.xml</param>
  <param
name="rice.location.additionalSpringFiles">classpath:edu/iu/uis/rice/resources/IULocationOverrideSpringBeans.xml</param>
  <param
name="rice.coreservice.additionalSpringFiles">classpath:edu/iu/uis/rice/resources/IUCoreServiceOverrideSpringBeans.xml</param>

  <param
name="application.url">https://test.uisapp2.iu.edu/${app.code}-${environment}</param>
  <param
name="secureServiceServletUrl">http://${http.url}/${app.code}-${environment}/remoting/</param>
  <param name="serviceServletUrl">${application.url}/remoting/</param>

  <!-- If you were overriding rice.struts.message.resources before to include your own
properties file, change it to the following -->

  <param name="rice.struts.message.resources"
override="true">org.kuali.rice.krad.ApplicationResources,org.kuali.rice.krad.KRAD-ApplicationResources,org.kuali.rice.kew.ApplicationResources,org.kuali.rice.ksb.ApplicationResources,org.kuali.rice.kim.ApplicationResources,org.kuali.rice.krms.ApplicationResources,org.kuali.rice.core.web.cache.CacheApplicationResources,YOUR.APP.ApplicationResources</param>

  ... remoting urls, various other urls, database platform, etc...

  <!-- workflow database params ; Notice change in the platform package name -->
  <param name="kr.datasource.objb.platform">Oracle9i</param>
  <param
name="tk.datasource.platform">org.kuali.rice.core.framework.persistence.platform.OraclePlatform</param>
  <param name="kr.datasource.driver.name">oracle.jdbc.driver.OracleDriver</param>
  <param name="kr.datasource.pool.validationQuery">select 1 from dual</param>
  <param name="kr.datasource.pool.maxWait">30000</param>
  <param name="kr.datasource.pool.minSize">20</param>
  <param name="kr.datasource.pool.maxSize">50</param>
```

```
</config>
```

Standard configuration of "thin" client Rice projects

```
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans-3.0.xsd">

  <bean id="ksbConfigurer" class="org.kuali.rice.ksb.messaging.config.KSBConfigurer"/>

  <bean id="kimConfigurer" class="org.kuali.rice.kim.config.KIMConfigurer"/>

  <bean id="kewConfigurer" class="org.kuali.rice.kew.config.KEWConfigurer"/>

</beans>
```

```
<config>

  <param name="kim.mode">THIN</param>
  <param name="kew.mode">THIN</param>
  <param name="ksb.mode">THIN</param>

</config>
```

Encryption Key Update in Override Cases

If you override the encryption key, there is an extra step that needs to be done to ensure users have access to existing workflow documents. Due to the move from commons-codec 1.3 to 1.6 in Rice 2.0 and the way the library works in the face of improperly padded base64 strings, the encryption.key configuration property needs to be updated to ensure that it has the correct padding. We've provided a utility for doing this and it's located here...

<http://svn.kuali.org/repos/rice/tools/encryption-key-converter/trunk>

To run this...

1. Check out the app above
2. run mvn package from the command line to create the executable jar in the target folder
3. Run the executable in the target directory with "java -jar encryption-key-converter-1.0-SNAPSHOT.jar <current encryption.key property value>"
 - a. so if the old key was 7IC64w6ksLR the command would be
 - i. java -jar target/encryption-key-converter-1.0-SNAPSHOT.jar 7IC64w6ksLR



The app contains the commons-codec 1.3 inside of the executable so there's no need to grab it elsewhere

Global Changes

Package Names

You can organize imports at the highest level by right clicking on a source folder and then selecting source --> organize imports. For additional

notes, see the [Package Changes](#) section.

Service locator & Service Name Changes

In KR 2.0, service locators with a name of *ServiceLocatorWeb should only be used internally for Rice and not by clients. However, the exception to this is the KRADServiceLocatorWeb which can be used by clients.

1.0.3 Name	2.0 Name	Notes
KEWServiceLocator.getStyleService()	CoreApiServiceLocator.getStyleService()	
KEWServiceLocator.getWorkflowDocumentActionsService()	KewApiServiceLocator.getWorkflowDocumentActionsService()	
KIMServiceLocator.getIdentityManagementService().isAuthorized()	KimApiServiceLocator.getPermissionService().isAuthorized()	
KIMServiceLocator.getGroupService()	KimApiServiceLocator.getGroupService()	
KIMServiceLocator.getIdentityManagementService()	KimApiServiceLocator.getIdentityService()	
KIMServiceLocator.getPersonService()	KimApiServiceLocator.getPersonService()	
KNSServiceLocator.getBusinessObjectService()	KRADServiceLocator.getBusinessObjectService()	
KnsServiceLocator.getDateTimeService()	CoreApiServiceLocator.getDateTimeService()	
KNSServiceLocator.getDocumentService()	KRADServiceLocatorWeb.getDocumentService()	
KNSServiceLocator.getLookupService()	KRADServiceLocatorWeb.getLookupService()	findCollectionBySearch previously took an unparameterized Map, but it now must be <String, String>
KNSServiceLocator.getKualiModuleService()	KRADServiceLocatorWeb.getKualiModuleService()	
KNSServiceLocator.getDataDictionaryService()	KRADServiceLocatorWeb.getDocumentDictionaryService()	
KEWServiceLocator.getEmailService().sendEmail	CoreApiServiceLocator.getMailer().sendEmail	

Changes to UserSession

The kew UserSession (org.kuali.rice.kew.web.session.UserSession) was combined with the kns version of the UserSession.

- All kew related properties that were present in the kew UserSession are now stored in the objectMap on the kns UserSession. These properties can be accessed using the following constants defined in rice:
 - KewApiConstants.ACTION_LIST_FILTER_ATTR_NAME
 - KewApiConstants.UPDATE_ACTION_LIST_ATTR_NAME
 - KewApiConstants.SORT_ORDER_ATTR_NAME
 - KewApiConstants.SORT_CRITERIA_ATTR_NAME
 - KewApiConstants.CURRENT_PAGE_ATTR_NAME
 - KewApiConstants.HELP_DESK_ACTION_LIST_PRINCIPAL_ATTR_NAME
 - KewApiConstants.HELP_DESK_ACTION_LIST_PERSON_ATTR_NAME
 - KewApiConstants.PREFERENCES
- The UserPreferences that the kew UserSession used to initialize is now initialized by a ServletFilter called (UserPreferencesFilter) that must sit behind the UserLoginFilter

Example code:

Old 1.0.3 code :

```

public ActionForward execute(ActionMapping mapping, ActionForm form,
    HttpServletRequest request, HttpServletResponse response) throws Exception {
    request.setAttribute("Constants", new
    JSTLConstants(KEWConstants.class));
    request.setAttribute("preferences",
    this.getUserSession(request).getPreferences());
    initForm(request, form);
    return super.execute(mapping, form, request, response);
}

```

New 2.0 code equivalent :

```

public ActionForward execute(ActionMapping mapping, ActionForm form,
    HttpServletRequest request, HttpServletResponse response) throws Exception {
    request.setAttribute("Constants",
    getServlet().getServletContext().getAttribute("KewApiConstants"));
    request.setAttribute("preferences",
    this.getUserSession().retrieveObject(KewApiConstants.PREFERENCES));
    initForm(request, form);
    return super.execute(mapping, form, request, response);
}

```

Common code in 1.0.3	2.0 Equivalent
org.kuali.rice.kns.UserSession.setWorkflowDocument(KualiWorkflowDocument workflowDocument)	SessionDocumentService.getDocumentFromSession(UserSession userSession, String docId) SessionDocumentService.addDocumentToUserSession(UserSession userSession, KualiWorkflowDocument document)
UserSession.getEditablePropertiesHistoryHolder()	UserSession.getObjectMap().get(KNSConstants.EDITABLE_PROPERTIES_HISTORY HOLDER_ATTR_NAME)

Changes to Constant Classes (say that 5 times fast!)

- Package Names have changed
- Many constants moved from one constant class to another.
- JSTLConstants.java was replaced by ConstantsMap which is designed to be used as a SpringBean and exported to the ServletContext. Constant classes should longer extend JSTLConstants.
- It is no longer possible to extend a kuali rice constant classes such as KEWConstants

1.0.3 Constant in this class	Now may be in this class for 2.0	Notes
org.kuali.rice.kns.util.KNSConstants	org.kuali.rice.krad.util.KRADConstants	
org.kuali.rice.kew.util.KEWConstants	org.kuali.rice.kew.api.KewApiConstants	
KSBCConstants.*	KSBCConstants.Config.*	
org.kuali.rice.core.util.RiceKeyConstants	org.kuali.rice.core.api.util.RiceKeyConstants	Package Change only
org.kuali.rice.core.util.RiceConstants	org.kuali.rice.core.api.util.RiceConstants	Package Change only
org.kuali.rice.kew.batch.XmlConstants	use org.kuali.rice.kew.xml.XmlConstants	Package Change only

RouteHeaderId changed to DocumentId

Be careful when doing a global find and replace. It is recommended that you do a preview to verify you want to change all occurrences of RouteHeaderId.

Use of immutable objects and the builder pattern for various APIs

Builders are objects used to assemble and construct an instance of another object. Typically the object being constructed will be an immutable object (and therefore does not contain "setters" which can be used to mutate it's state). The builder pattern is a creation pattern that can be used to aid in the construction of these complex immutable objects.

Old 1.0.3 code example:

```
for (KimEntityEntityTypeInfo type : kimEntity.getEntityTypes()) {
    if (KimConstants.EntityTypes.PERSON.equals(type.getEntityTypeCode())) {
        entityType = type;
    }
}
```

New 2.0 code equivalent :

```
for (EntityTypeContactInfo type : kimEntity.getEntityTypeContactInfos()) {
    if (KimConstants.EntityTypes.PERSON.equals(type.getEntityTypeCode())) {
        entityType = EntityTypeContactInfo.Builder.create(type).build();
    }
}
```

General Purpose Lookup APIs

In Rice 1.0.3, many of the services had general-purpose lookup methods which took a `Map<String, String>`. In this map, the key was intended to represent the property name on the data element being looked up and the value was intended to be the value to search for. Some work was done to design a general purpose lookup API that could be applied in a standard fashion across the various Rice services which needed it. The following page describes more about the various possibilities for implementing this as well as the solution which was chosen: <https://wiki.kuali.org/x/PR9yEg>

The implementation of this api can be found on the core-api module under the package `org.kuali.rice.core.api.criteria`.

Old 1.0.3 code :

```
private List<String> getPrincipalIdsForPrincipalName(String principalName){
    Map<String, String> criteria = new HashMap<String, String>();
    criteria.put("principals.principalName", principalName);
    List<? extends KimEntityDefaultInfo> entities =
    KIMServiceLocator.getIdentityService().lookupEntityDefaultInfo(criteria, false);

    List<String> principalIds = new ArrayList<String>();
    for (KimEntityDefaultInfo entity : entities) {
        for (KimPrincipal principal : entity.getPrincipals()) {
            principalIds.add(principal.getPrincipalId());
        }
    }
    return principalIds;
}
```

New 2.0 code equivalent :

```

private List<String> getPrincipalIdsForPrincipalName(String principalName) {
    QueryByCriteria.Builder qb = QueryByCriteria.Builder.create();
    qb.setPredicates(equal("principals.principalName", principalName));
    List<EntityDefault> entities =
    KimApiServiceLocator.getIdentityService().findEntityDefaults(qb.build()).getResults();

    List<String> principalIds = new ArrayList<String>();
    for (EntityDefault entity : entities) {
        for (Principal principal : entity.getPrincipals()) {
            principalIds.add(principal.getPrincipalId());
        }
    }
    return principalIds;
}

```

RunModes (kew.mode, kim.mode, etc.) & Client Protocol (client.protocol)

These are now java enums. The properties must be in all caps (ex: LOCAL, EMBEDDED).

New required listeners

A new listener is required to use along with the Spring Web-Aware contexts

```

<listener>

<listener-class>org.springframework.web.context.request.RequestContextListener</listen
er-class>
</listener>

```

Also, org.kuali.rice.kns.web.listener.KualiHttpSessionListener is now org.kuali.rice.krad.web.listener.KualiHttpSessionListener

'Service Namespace' concept replaced with 'Application ID' concept

Old 1.0.3 code :

```
String serviceNamespace = document.getDocumentType().getServiceNamespace();
```

New 2.0 code equivalent :

```
String applicationId = document.getDocumentType().getApplicationId();
```

KEN and KCB Changes

- The KEN and KCB services have not yet gone through the refactoring to align them with Kualu Rice 2.0 service compatibility standards. The services that are published for these modules will likely change in subsequent beta releases. However, they should be finalized prior to the final release of Rice 2.0.
- Both have been converted to utilize standard SOAP interfaces.

KEW Changes



Detailed notes on the reasoning and further impact of some of the KEW changes can be found in Rice documentation here: [Rice 2.0 - Compatibility Refactoring - Kualu Enterprise Workflow](#)

Workflow services

The Kualu Enterprise Workflow services have been refactored for version compatibility in a similar fashion to the rest of the services in Rice. The major changes made to the workflow services and APIs are as follows:

- The WorkflowDocument class is now an interface and is created using WorkflowDocumentFactory.createDocument or WorkflowDocumentFactory.loadDocument instead of creating new instances through the WorkflowDocument class.

Old 1.0.3 code:

```
WorkflowDocument workflowDocument =  
WorkflowDocument.loadDocument(GlobalVariables.getUserSession().getPrincipalId(),  
docForm.getRouteHeaderId());
```

New 2.0 code equivalent :

```
WorkflowDocument workflowDocument =  
WorkflowDocumentFactory.loadDocument(GlobalVariables.getUserSession().getPrincipalId(), docForm.getDocumentId());
```

- SimpleDocumentActionsWebService and WorkflowDocumentActions were combined to create the new WorkflowDocumentActionsService remotable service
- WorkflowInfo and WorkflowUtility were removed and replaced with the following services:
 - WorkflowDocumentService
 - ActionListService
 - DocumentTypeService
 - NoteService
 - RuleService

Old 1.0.3 code:

```
DocumentTypeDTO docType = workflowInfo.getDocType(docTypeName);
```

New 2.0 code equivalent :

```
DocumentType docType =  
KewApiServiceLocator.getDocumentTypeService().getDocumentTypeByName(docTypeName);
```

- The various KEW framework **services** were refactored to conform to service version compatibility standards. This includes the following services:
 - document search customization
 - action list customization
 - workflow notes
 - document security
 - rule validation

- workflow rule attributes
- The various KEW message **queues** were refactored to conform to service version compatibility standards. This includes following queues:
 - document routing
 - blanket approval
 - immediate email reminders
 - searchable attribute indexing
 - document requeuing
- In the KNS, both KualWorkflowInfo and KualWorkflowDocument were removed. In the case of KualWorkflowDocument, existing references were changed to simply WorkflowDocument. In the case of KualWorkflowInfo, code was changed to call one of the new KEW services that was added in Rice 2.0.

WorkflowAttributeDefinitionDTO has been renamed to WorkflowAttributeDefinition and is now a final class.

Old 1.0.3 code:

```
import org.kuali.rice.kew.dto.WorkflowAttributeDefinitionDTO;

public class AccountDefinition extends WorkflowAttributeDefinitionDTO {
    static final long serialVersionUID = 1000;
    private static final String ACCOUNT_ATTRIBUTE_CLASS =
"edu.iu.uis.eden.routetemplate.attribute.AccountAttribute";
    public AccountDefinition(String finCoaCd, String accountNbr, String
totalDollarAmount) {
        this();
        this.addConstructorParameter(finCoaCd);
        this.addConstructorParameter(accountNbr);
        this.addConstructorParameter(totalDollarAmount);
    }
    private AccountDefinition() {
        super(ACCOUNT_ATTRIBUTE_CLASS);
    }
}
```

New 2.0 code equivalent :

```
public class AccountDefinition {
    static final long serialVersionUID = 1000;
    private static final String ACCOUNT_ATTRIBUTE_CLASS =
"edu.iu.uis.eden.routetemplate.attribute.AccountAttribute";

    public static WorkflowAttributeDefinition createAttributeDefinition(String finCoaCd,
String accountNbr, String totalDollarAmount) {
        WorkflowAttributeDefinition.Builder builder =
WorkflowAttributeDefinition.Builder.create(ACCOUNT_ATTRIBUTE_CLASS);
        builder.addParameter(finCoaCd);
        builder.addParameter(accountNbr);
        builder.addParameter(totalDollarAmount);
        return builder.build();
    }

    private AccountDefinition() {}
}
```

Old 1.0.3 code:

```
WorkflowAttributeDefinition attributeDefinition = new
WorkflowAttributeDefinition(VendorTypeDefinition.VENDOR_TYPE_ATTRIBUTE_CLASS);
attributeDefinition.addConstructorParameter(vendorTypeCode);
document.addAttributeDefinition(attributeDefinition);
```

New 2.0 code equivalent :

```
WorkflowAttributeDefinition attributeDefinition =
VendorTypeDefinition.createAttributeDefinition(vendorTypeCode);
document.addAttributeDefinition(attributeDefinition );
```

WorkflowRuleAttribute validation error type changes

WorkflowAttributeValidationError on the WorkflowRuleAttribute interface has been replaced by AttributeError and RemoteAttributeError. Custom workflow attributes will have to be updated due to this change.

removal of getStatusDisplayValue

Old 1.0.3 code:

```
EDLXmlUtils.createTextElementOnParent(workflowDocumentStatus, "status",
document.getStatusDisplayValue());
```

New 2.0 code equivalent :

```
EDLXmlUtils.createTextElementOnParent(workflowDocumentStatus, "status",
document.getStatus().getLabel());
```

What to use as a replacement for the userIsInitiator method that used to be on KualiiWorkflowDocument

```
KewApiServiceLocator.getWorkflowDocumentService().getDocumentInitiatorPrincipalId(String
documentId)
```

Replacement for getActionRequests(documentNumber)

```
KewApiServiceLocator.getWorkflowDocumentService().getRootActionRequests(String
documentId)
```

Addition of PeopleFlows

PeopleFlows are a new concept that was added to KEW as part of workflow routing requirements for Quali Coeus. The requirement was essentially to replicate the functionality of "Maps" in the existing Coeus system upon which Quali Coeus is based. These maps are simply lists of people or roles who can approve in a particular order. This order is defined by a series of "stops". At each stop there can be a number of possible approvers which may have delegates. There was not really an equivalent concept in KEW today since this is essentially a mini-workflow. So the concept of PeopleFlows were added along with a new activation type called "Priority-Parallel" activation. This priority-parallel activation allows for action requests to be activated at a node in parallel if they have the same priority, but in sequence if they have different priority values.

For more information about PeopleFlows, please see the [PeopleFlow documentation](#).

Changes to document search

SQL-level customization of document search will no longer be allowed in Rice 2.0 and the CustomDocumentSearchGenerator has been removed. Instead all document search customization is done through dynamic criteria customization. The new way to do this would be to create a DocumentSearchCustomizer and implement the "customizeCriteria" method to return a DocumentSearchCriteria which returns your additional document types you want to search on in getAdditionalDocumentTypeNames(). In addition, the isCustomizeCriteriaEnabled method needs to return true.

Here is a link to [more information](#)

Remaining numeric identifiers in KEW database Converted to character-based

In the KREW tables and corresponding code, all remaining numeric identifiers have been converted to character-based. Some of the IDs that may impact client code include the following:

Table	Fields
KREW_DLGN_RSP_T	<ul style="list-style-type: none">• DLGN_RULE_ID• RSP_ID• DLGN_RULE_BASE_VAL_ID
KREW_RULE_RSP_T	<ul style="list-style-type: none">• RULE_RSP_ID• RSP_ID• RULE_ID
KREW_RULE_EXT_VAL_T	<ul style="list-style-type: none">• RULE_EXT_VAL_ID• RULE_EXT_ID

For a complete list, please see JIRA [KULRICE-5201](#)

KIM Changes

Class name changes

The Kim that used to start most KIM class names has been removed. i.e. KimPrincipal is now Principal, KimEntityAddress is now EntityAddress, etc

Removal of IdentityManagementService

The Identity Management Service has been replaced with individual service calls in KIM. For a complete list of changes, please see the [Service Locator Changes Section](#)

Old 1.0.3 code :

```
@Override
KimPrincipal principal =
KIMServiceLocator.getIdentityManagementService().getPrincipalByPrincipalName(shahess);
```

New 2.0 code equivalent :

```
Principal principal =
KimApiServiceLocator.getIdentityService().getPrincipalByPrincipalName(shahess);
```


Removal of KimAttributeSet

KimAttributeSet was replaced with a simple Map<String, String>

KNS Changes

Business Objects

- Remove toStringMapper methods
- If you customize the BO Notes, you'll need to update the way you do so
- Changes in PersistableBusinessObjectBase
 1. Package has changed from `org.kuali.rice.kns.bo` to `org.kuali.rice.krad.bo`
 2. Some method are now final, such as `beforeInsert()` and `beforeUpdate()`. Must override `prePersist()` or `preUpdate()` instead.

Old 1.0.3 code :

```
@Override
public void beforeInsert(PersistenceBroker persistenceBroker) throws
PersistenceBrokerException {
    this.beforePersist();
}
```

New 2.0 code equivalent :

```
@Override
public void prePersist() throws PersistenceBrokerException {
    this.beforePersist();
}
```

Maintainables

- Convert service locator calls to appropriate module
- Verify WorkflowDocument usage

Action Classes

- Check any usage of UserSession. See [Changes to UserSession](#) for more information.

Values Finders

- import `org.kuali.rice.core.api.util.KeyValue` and `org.kuali.rice.core.api.util.ConcreteKeyValue` instead of `org.kuali.rice.core.util.KeyLabelPair`
- Convert return type to 'KeyValue' from 'KeyLabelPair'
- Convert returned list to use 'ConcreteKeyValue' as the storage structure from 'KeyLabelPair'
- ConcreteKeyValue constructor now only takes String, String

Old 1.0.3 code :

```

public List<KeyLabelPair> getKeyValues() {
    List retval = new ArrayList();
    retval.add(new KeyLabelPair("", ""));
    retval.add(new KeyLabelPair("Even", "Even"));
    retval.add(new KeyLabelPair("Odd", "Odd"));
    return retval;
}

```

New 2.0 code equivalent :

```

public List<KeyValue> getKeyValues() {
    List<KeyValue> retval = new ArrayList<KeyValue>();
    retval.add(new ConcreteKeyValue("", ""));
    retval.add(new ConcreteKeyValue("Even", "Even"));
    retval.add(new ConcreteKeyValue("Odd", "Odd"));
    return retval;
}

```

Transaction Documents

- These are now deprecated. Consider migrating to a standard maintenance document if possible or, more likely, to KRAD.
- route utility methods (eg. adHcRouteDocumentToPrincipal) are not on the TransactionDocumentBase anymore. Change these to use the WorkflowDocument APIs instead. (//TODO: verify)

Data Dictionary

- formatterClass definitions moved from kns to core packages

```

<property name="formatterClass"
value="org.kuali.rice.core.web.format.DateFormatter" />

```

- businessObjectClass definition moved from kns to krad package

```

<property name="businessObjectClass"
value="org.kuali.rice.krad.bo.DocumentHeader" />

```

- documentAuthorizer definitions moved from kns to krad.uif packages

Changes to Notes

- Not directly on the BO anymore, use NoteService and getByRemoteObjectId
- Note-related code in KNS does not depend on the document framework
- PersistableBusinessObject code in the KNS no longer depends on Notes

Changes to GlobalVariables.java

- Package has changed from org.kuali.rice.krad.util.GlobalVariables to org.kuali.rice.krad.util.GlobalVariables
- Some items were deprecated and moved to KNSGlobalVariables.java, such as messageLists

Old 1.0.3 code :

```
GlobalVariables.getMessageList().add(RiceKeyConstants.MESSAGE_SEND_NOTE_NOTIFICATION_SUCCESSFUL);
return (LookupForm) GlobalVariables.getKualiForm();
```

New 2.0 code equivalent :

```
KNSGlobalVariables.getMessageList().add(RiceKeyConstants.MESSAGE_SEND_NOTE_NOTIFICATION_SUCCESSFUL);
return (LookupForm) KNSGlobalVariables.getKualiForm();
```

- Various calls and methods have changed for MessageMap

Old 1.0.3 code :

```
if (!GlobalVariables.getMessageMap().isEmpty()) ...
if (GlobalVariables.getMessageMap().containsKey("biodemo.stepone")) { ...
GlobalVariables.getMessageMap().put(errorPrefix + ".allAccountingLines.residual",
"error.earning.accountinglines.residual.multiple"); ...
```

New 2.0 code equivalent :

```
if (!GlobalVariables.getMessageMap().hasNoErrors()) ...
if (GlobalVariables.getMessageMap().containsMessageKey("biodemo.stepone")) { ...
GlobalVariables.getMessageMap().putError(errorPrefix +
".allAccountingLines.residual",
"error.earning.accountinglines.residual.multiple"); ....
```

Changes to Kuali Configuration Service

- Method names have changed. i.e. getPropertyString is now getPropertyValueAsString

Old 1.0.3 code :

```
initiateURL =
UrlFactory.parameterizeUrl(getKualiConfigurationService().getPropertyString(KRADConstants.APPLICATION_URL_KEY) + "/" + "MaintainPersonDocument.do", parameters);
```

New 2.0 code equivalent :

```
initiateURL =
UrlFactory.parameterizeUrl(getKualiConfigurationService().getPropertyValueAsString(KRADConstants.APPLICATION_URL_KEY) + "/" + "MaintainPersonDocument.do",
parameters);
```

Removal of RiceApplicationConfigurationService

In Rice 1.0.3, the RiceApplicationConfigurationService was a service published by all KNS applications which provided callbacks into the application from the Kualu Rice standalone server. This service was removed in Rice 2.0 for various reasons.

The impact of this change will be as follows:

- A client application which is utilizing docHandlerUrl variable replacement via ApplicationConfigurationService.getConfigParameter will need to create a system parameter instead. Using the example of a docHandlerUrl containing `\${application.url}` on a document type with an application id of "KFS", a system parameter would need to be created :
 - With a namespace code of "KR-WKFLW"
 - With a component code of "All"
 - With an application id of "KFS"
 - With a name of "application.url"
 - With a value that is equal to the appropriate value for the application url
- A client application which is using StandardGenericXmlSearchableAttribute using a businessObjectClass in that definition (in order to render a lookup on a doc search field) will need to ensure to set the applicationId of the entry in KREW_RULE_ATTR_T which points to their attribute such that it is the application id of their application. This tells KEW that it needs to execute the xml-based attribute code inside of the client application (where it can successfully resolve lookup url)
- Derived components will still be calculated the same way but will be pushed to the rice server from the client application on startup and persisted into a database. Specific information can be found in the ParameterService section.
- KualuAction will no longer always fall back to resolving lookup and inquiry urls remotely if it can't find a business object. Client applications need to be sure that their ModuleServices are returning proper inquiry and lookup url values for their ExternalizableBusinessObjects.

For more information please see <https://wiki.kuali.org/x/vA6tEg>

Removal of ConfigProperties.java

The org.kuali.rice.kns.ConfigProperties.java class has been replaced with the following:

```
<bean id="ConfigProperties"
class="org.kuali.rice.core.api.util.collect.PropertiesMap">
  <property name="properties" value="#{kualiConfigurationService.allProperties}"/>
</bean>
```

KSB Changes



Detailed notes on the reasoning and further impact of this work can be found in Rice documentation here: <https://wiki.kuali.org/display/KULRICE/Rice+2.0+-+Compatibility+Refactoring+-+Kuali+Service+Bus>

KSB Architecture changes

1. Migration from Java serialization to SOAP web services (integration of standalone <-> clients)
2. As a result of the above, client applications should now connect to the service registry through an exposes SOAP service, NOT a direct DB connection
3. Ability to configure KSB usage in several ways (consumer only, publisher only, full use, etc..)
4. Ability to use the 'publishService' method directly (via the API) to expose services
 - This eliminates (but not forcibly) the need to configure Spring serviceDefinitions for each service
5. Use of two run modes
 - LOCAL - used by standalone (or KSB registry dedicated instance) (connects directly to the DB)
 - REMOTE - used by ALL client applications (connects through SOAP interfaces)

Package / Class Name Changes

Class & Method Name Changes

Name in 1.0.3	Name in 2.0	Method Name changes
---------------	-------------	---------------------

org.kuali.rice.kew.docsearch.StandardDocumentSearchGenerator	org.kuali.rice.kew.impl.document.search.DocumentSearchGeneratorImpl	<ul style="list-style-type: none"> • addExtraDocumentTypesToSearch --> addDocumentTypeNameToSearchOn
org.kuali.rice.kew.docsearch.DocumentSearchContext		
org.kuali.rice.kew.rule.WorkflowAttribute	org.kuali.rice.kew.rule.WorkflowRuleAttribute	
org.kuali.rice.kew.docsearch.xml.StandardGenericXMLSearchableAttribute	org.kuali.rice.kew.docsearch.xml.StandardGenericXMLSearchableAttribute	<ul style="list-style-type: none"> • getSearchContent --> generateSearchContent • getSearchStorageValues --> extractDocumentAttributes
org.kuali.rice.kew.rule.bo.RuleAttribute	org.kuali.rice.kew.rule.bo.RuleAttribute	<ul style="list-style-type: none"> • getClassName --> getResourceDescriptor
org.kuali.rice.kew.docsearch.DocumentSearchContext	org.kuali.rice.kew.api.document.DocumentWithContent	
org.kuali.rice.kew.dto.DocumentTypeDTO	org.kuali.rice.kew.api.doctype.DocumentType	
org.kuali.rice.kew.dto.DocumentRouteStatusChangeDTO	org.kuali.rice.kew.framework.postprocessor.DocumentRouteStatusChange	
org.kuali.rice.kew.dto.ActionRequestDTO	org.kuali.rice.kew.api.action.ActionRequest	
org.kuali.rice.kew.dto.DocumentContentDTO	org.kuali.rice.kew.api.document.DocumentContent	
org.kuali.rice.kew.dto.RouteHeaderDTO		
org.kuali.rice.kew.exception.InvalidXmlException	org.kuali.rice.core.api.util.xml.XmlException	
org.kuali.rice.kew.rule.RuleExtension	org.kuali.rice.kew.rule.RuleExtensionBo	This is also true for many other classes. They went from the ClassName to ClassNameBo.
org.kuali.rice.kew.rule.Role	org.kuali.rice.kew.api.rule.RoleName	
org.kuali.rice.core.util.KeyLabelPair	org.kuali.rice.core.api.util.KeyValue	<ul style="list-style-type: none"> • getLabel --> getValue
org.kuali.rice.kim.bo.entity.KimPrincipal	org.kuali.rice.kim.api.identity.principal.Principal	
org.kuali.rice.kim.bo.group.dto.GroupInfo	org.kuali.rice.kim.api.group.Group	<ul style="list-style-type: none"> • getGroupId --> getId • getGroupName --> getName
org.kuali.rice.kns.lookup.keyvalues.KeyValuesBase	org.kuali.rice.krad.keyvalues.KeyValuesBase	
org.kuali.rice.kns.bo.BusinessObjectRelationship	org.kuali.rice.krad.bo.DataObjectRelationship	
org.kuali.rice.kns.workflow.service.KualiWorkflowDocument	org.kuali.rice.kew.api.WorkflowDocument	<ul style="list-style-type: none"> • getInitiatorNetworkId --> getInitiatorPrincipalId • logDocumentAction --> logAnnotation • stateIsInitiated --> isInitiated • stateIsSaved --> isSaved • stateIsEnroute --> isEnroute • stateIsProcessed --> isProcessed • stateIsFinal --> isFinal • saveRoutingData --> saveDocumentData

org.kuali.rice.kns.util.MessageMap org.kuali.rice.kns.util.ErrorMap	org.kuali.rice.krad.util.MessageMap	<ul style="list-style-type: none"> • clear --> clearErrorMessages • entrySet --> getAllPropertiesAndErrors • get --> getErrorMessagesForProperty • keySet --> getAllPropertiesWithErrors • remove --> removeAllErrorMessagesForProperty • size --> getNumberOfPropertiesWithErrors • put --> no replacement • putAll --> no replacement • containsValue --> no replacement • values --> no replacement
org.kuali.rice.ksb.message.KSBExporter	org.kuali.rice.ksb.api.bus.support.ServiceBus Exporter	

Important Package Changes

Class Name	Package Name in 1.0.3	Package Name in 2.0
UserSession	org.kuali.rice.kns	org.kuali.rice.krad
<ul style="list-style-type: none"> • Document • TransactionalDocumentBase 	org.kuali.rice.kns.document	org.kuali.rice.krad.document
<ul style="list-style-type: none"> • BusinessObject • PersistableBusinessObject • PersistableBusinessObjectBase • DocumentHeader • Note 	org.kuali.rice.kns.bo	org.kuali.rice.krad.bo
<ul style="list-style-type: none"> • DataDictionary • DocumentEntry • PrimitiveAttributeDefinition • RelationshipDefinition • SupportAttributeDefinition 	org.kuali.rice.kns.datadictionary	org.kuali.rice.krad.datadictionary
<ul style="list-style-type: none"> • DocumentAuthorizer • DocumentAuthorizerBase 	org.kuali.rice.kns.document.authorization	org.kuali.rice.krad.document.authorization
<ul style="list-style-type: none"> • DocumentAuthorizationException • ValidationException 	org.kuali.rice.kns.exception	org.kuali.rice.krad.exception
TransactionalDocumentRuleBase	org.kuali.rice.kns.rules	org.kuali.rice.krad.rules
<ul style="list-style-type: none"> • BlanketApproveDocumentEvent • AddNoteEvent • KualiDocumentEvent 	org.kuali.rice.kns.rule.event	org.kuali.rice.krad.rule.event
DocumentDao	org.kuali.rice.kns.dao	org.kuali.rice.krad.dao
<ul style="list-style-type: none"> • GlobalVariables • KualiDecimal • MessageMap • UriFactory 	org.kuali.rice.kns.util	org.kuali.rice.krad.util
DateTimeService	org.kuali.rice.kns.service	org.kuali.rice.core.api
ModuleServiceBase	org.kuali.rice.kns.service.impl	org.kuali.rice.krad.service.impl
<ul style="list-style-type: none"> • Config • ConfigContext 	org.kuali.rice.core.config	org.kuali.rice.core.api.config.property
ConfigFactoryBean	org.kuali.rice.core.config.spring	org.kuali.rice.core.impl.config.property
<ul style="list-style-type: none"> • Lifecycle • BaseLifecycle 	org.kuali.rice.core.lifecycle	org.kuali.rice.core.api.lifecycle
ContextClassLoaderBinder	org.kuali.rice.core.resource.loader	org.kuali.rice.core.util.
WorkflowException	org.kuali.rice.kew.exception	org.kuali.rice.kew.api.exception

WorkflowDocument	org.kuali.rice.kew.service	org.kuali.rice.kew.api
<ul style="list-style-type: none"> PrincipalId Id 	org.kuali.rice.kew.identity	org.kuali.rice.kew.api.identity
Person	org.kuali.rice.kim.bo	org.kuali.rice.kim.api.identity
AuthenticationService	org.kuali.rice.kim.service	org.kuali.rice.kim.api.identity
InvalidXmlException	org.kuali.rice.kew.exception	org.kuali.rice.core.util.xml
InvalidActionTakenException	org.kuali.rice.kew.exception.	org.kuali.rice.kew.api.exception
<ul style="list-style-type: none"> StrutsExceptionHandler AuthorizationExceptionHandler 	org.kuali.rice.kns.web.struts.pojo	org.kuali.rice.kns.web.struts.form.pojo
JavaServiceDefinition	org.kuali.rice.ksb.messaging	org.kuali.rice.ksb.api.bus.support

Miscellaneous changes / deletions that have not been mentioned above

class Name in 1.0.3	notes
org.kuali.rice.ken.exception.NotificationMessageDelivererNotFoundException	no replacement
org.kuali.rice.kns.util.spring.AutoPopulatingList	use org.springframework.util.AutoPopulatingList
org.kuali.rice.kns.util.TypedArrayList	use org.springframework.util.AutoPopulatingList
org.kuali.rice.kns.util.Guid	use java.util.UUID.randomUUID()
org.kuali.rice.kns.service.impl.ConfigurableDateTimeServiceImpl	no replacement
org.kuali.rice.kns.service.ConfigurableDateTimeService	no replacement
org.kuali.rice.kns.mail.UnableToSendMessageError	no replacement
org.kuali.rice.kns.datadictionary.exception.OverrideEntryException	no replacement
org.kuali.rice.kns.exception.BusinessObjectNotFoundException	no replacement
org.kuali.rice.kns.exception.IllegalObjectStateException	no replacement
org.kuali.rice.kns.exception.MetadataException	no replacement
org.kuali.rice.kns.exception.ObjectNotABusinessObjectException	no replacement
org.kuali.rice.kns.exception.ReferentialIntegrityException	no replacement
org.kuali.rice.kns.exception.UnknownBusinessClassException	no replacement
org.kuali.rice.core.lifecycle.SpringLifeCycle	no replacement
org.kuali.rice.core.resource.loader.SpringLoader	no replacement
org.kuali.rice.core.config.event.*	no replacement
org.kuali.rice.core.exception.InvalidXmlException	no replacement
org.kuali.rice.core.exception.RiceException	no replacement
org.kuali.rice.kew.bo.BaseWorkflowPersistable	no replacement
org.kuali.rice.kns.web.RiceConfigurationListener	no replacement
org.kuali.rice.ken.bo.KenPersistableBusinessObjectBase	no replacement
org.kuali.rice.core.resource.loader.ContextualConfigLock	use org.kuali.rice.core.config.ContextualConfigLock
org.kuali.rice.core.web.struts.RiceActionServlet	extend org.apache.struts.action.ActionServlet directly
org.kuali.rice.kns.util.comparator.BeanPropertyComparator	use org.kuali.rice.kns.util.BeanPropertyComparator

org.kuali.rice.kns.exception.BeanComparisonException	org.kuali.rice.kns.util.comparator.BeanPropertyComparator.BeanComparisonException
org.kuali.rice.kns.bo.ConcurrentLockId	no replacement
org.kuali.rice.kns.service.ConfigurableDateService	no replacement
org.kuali.rice.kns.util.spring.ClassPathXmlApplicationContext	use org.springframework.context.support.ClassPathXmlApplicationContext
DocSearchUtils.getSqlFormattedDate()	SQLUtils.getSqlFormattedDate()
DocSearchUtils.getEntryFormattedDate()	SQLUtils.getEntryFormattedDate()
DocSearchUtils.convertStringDateToTimestamp()	SQLUtils.convertStringDateToTimestamp()
Utilities.convertTimestamp()	SQLUtils.convertTimestamp()
Utilities.convertCalendar()	SQLUtils.convertCalendar()
org.kuali.rice.core.dao.impl.GenericDaoJpa	no replacement
org.kuali.rice.kew.bo.KewPersistableBusinessObjectBase	org.kuali.rice.kns.bo.PersistableBusinessObjectBase
org.kuali.rice.kew.dto.RouteTemplateEntryDTO	no replacement
AuthenticationUserId	org.kuali.rice.kew.identity.PrincipalName
org.kuali.rice.kew.user.UserCapabilities	no replacement
org.kuali.rice.kim.bo.reference.DefaultableKimCode	no replacement
org.kuali.rice.core.config.SimpleNodeSettingsStore	no replacement
org.kuali.rice.core.config.spring.SimpleConfigFactoryBean	no replacement
org.kuali.rice.core.database.SpringConnectionFactoryManagedImpl	ask eric
org.kuali.rice.kns.util.spring.WorkflowSpringConnectionFactoryManagedImpl	ask eric
org.kuali.rice.core.jdbc.PreparedObject	no replacement
org.kuali.rice.core.jta.AtomikosTransactionManagerLookup	ask eric
org.kuali.rice.core.resourceloader.ResourceLoaderServiceFactoryBean	GRLFactoryBean
org.kuali.rice.core.resourceloader.RootResourceLoaderLifecycle	no replacement
org.kuali.rice.core.resourceloader.RiceSpringResourceLoaderConfigurer	no replacement
org.kuali.rice.core.util.ConcurrencyDetector	was a debugging tool that we felt was not that useful. no replacement
org.kuali.rice.core.versions.*	an unfinished POC, no replacement
org.kuali.rice.core.web.RequestForwardingServlet	legacy, no replacement
org.kuali.rice.kew.batch.KEWXmlDataLoaderLifecycle	move to internal testing tools since it should only be used by rice unit tests, no replacement
org.kuali.rice.kim.service.PersonService	removed the parameterized type (? extends Person), ee KULRICE-4871 for more details
ExceptionUtils	use org.apache.commons.lang.exception.ExceptionUtils
Util.readFully()	use org.apache.commons.lang.io.IOUtils.toByteArray()
Utilities.collectionEquivalent	CollectionUtils.collectionEquivalent
Utilities.isEmpty(Collection)	use org.apache.commons.lang.collection.CollectionUtils.isEmpty
Utilities.equals (Object a, Object b)	use org.apache.common.lang.ObjectUtils.equals()

ObjectUtils.nullsafeEquals()	use org.apache.common.lang.ObjectUtils.equals()
Utilities.isEmpty(String)	org.apache.common.lang.StringUtils.isEmpty()
org.kuali.rice.kns.util.DateUtils	use org.apache.common.lang.DateUtils if possible
org.kuali.rice.kns.util.NumberUtils	use org.apache.common.lang.NumberUtils if possible
BaseResourceLoader.wrap()	use BaseResourceLoader.postProcessService()
ConfigContext.getRootConfig()	ConfigContext.getCurrentContextConfig()
Config.DEFAULT_NOTE_CLASS	use Config.DEFAULT_KEW_NOTE_CLASS
Config.getDefaultNoteClass()	Config.getDefaultKewNoteClass()
Config.overrideProperty()	Config.putProperty()
IdentityHelperService.getGroupId()	use IdentityManagementService
IdentityHelperService.getPrincipal()	use IdentityManagementService
IdentityHelperService.getGroupRecipient()	use IdentityManagementService
KualiWorkflowDocument.getInitiatorNetworkId()	use KualiWorkflowDocument.getInitiatorPrincipalId instead *note this returns the principalID not principalName
KualiWorkflowDocument.getRoutedByUserNetworkId()	use KualiWorkflowDocument.getRoutedByPrincipalId() *note this returns the principalID not principalName
actionlist.outbox.default.preference.on is no longer a valid parameter	use userOptions.default.useOutBox
org.kuali.rice.core.util.WriterOutputStream	use org.apache.commons.io.output.WriterOutputStream
RiceDebugUtils	use org.apache.commons.lang.ExceptionUtils.getStackTrace()
ParameterServerService	use: KNSServiceLocatorWeb.getRiceApplicationConfigurationMediationService().getNonDatabaseComponents();
ParameterService.setParameterForTesting()	no direct replacement
org.kuali.rice.kns.util.Timer	no direct replacement. Used org.apache.commons.lang.StopWatch to replace in rice code.
Utilities.getKNSParameterValue()	use ClientParameterService.getParameterValueAsString()
Utilities.getKNSParameterBooleanValue()	use ClientParameterService.getParameterValueAsBoolean()
DerbyPlatform	use DerbyDatabasePlatform
MySQLPlatform	use MySQLDatabasePlatform
Oracle9iPlatform	use Oracle9iDatabasePlatform
Formatter.isBlank()	use commons StringUtils.isBlank()
KRAMetaRuleExpression	use MetaRuleExpression
UiDocumentServiceImpl.getFullMemberName()	UiDocumentServiceImpl.getMemberFullName()
DocumentDaoObjb.saveMaintainableBusinessObject	use BusinessObjectService.linkAndSave()
DocumentDaoJpa.saveMaintainableBusinessObject	use BusinessObjectService.linkAndSave()
LookupUtils.applySearchResultsLimit(Criteria criteria, DatabasePlatform platform)	LookupUtils.applySearchResultsLimit(Class businessObjectClass, Criteria criteria, DatabasePlatform platform)
DocumentServiceImpl.setKualiRuleService(KualiRuleService)	no replacement
DocumentServiceImpl.setDictionaryValidationService(DictionaryValidationService)	no replacement
DocumentServiceImpl.setMaintenanceDocumentService(MaintenanceDocumentService)	no replacement

ContextClassLoaderProxy.getInterfacesToProxyIncludeSpring()	use ContextClassLoaderProxy.getInterfacesToProxy()
org.kuali.rice.kew.rule.OddSearchAttribute	no replacement
DocumentTypeDTO.getDocTypeName()	use DocumentTypeDTO.getDocTypeLabel()
DocumentTypeDTO.setDocTypeName(String)	DocumentTypeDTO.setDocTypeLabel(String)
jstl function \${kfunc:getKNSParameterValue}	use struts form properties to hold parameter values or scriplets in extreme cases - if this is needed we can revisit introducing jstl functions for parameters
StandaloneInitializeListener	changed kew.bootstrap.spring.file to web.bootstrap.spring.file configuration parameter name - KULRICE-5014
workflowDocument.logDocumentAction	workflowDocument.logAnnotation
Util.jotNode, jotDocument, writeNode Utilities.jotNode, jotDocument, writeNode XmlJotter.jotNode, jotDocument, writeNode	XmlHelper.* writeNode is replaced by jotNode

Troubleshooting:

See [Rice 2.0 - Upgrade Troubleshooting](#)