User Messages 2

System Messages

To change system messages

The `ApplicationResources.properties` file at the root of the project will need to be edited. This file is referenced by the `org.kuali.kfs.KFSKeyConstants.java` class. This class contains a list of public static final Strings which form the keys. These keys are used in the code to reference a system message. The values for these keys are the keys of the `ApplicationResources.properties` file.

If the String value that is used in the code is known, the error message can be found by finding the line in `org.kuali.kfs.KFSKeyConstants.java` that contains that String. The key which matches the value of the String can then be found in the `ApplicationResources.properties` file and the value of the system message changed there.

If the text of the message is known, then the process may be simpler. Find the line in `ApplicationResources.properties` that has the value of the system message desired. Make any changes necessary. (***Note***: *many system messages contain replacement parameters from* `java.text.MessageFormat`.)

ⓘ Here is an example of how the `KFSKeyConstants.java` can be used to add an error to the global error map.

```
GlobalVariables.getErrorMap().putError(KFSPropertyConstants.DISB_VCHR_BANK_ROUTING_NUMBER, KFSKeyConstants.ERROR_DV_BANK_ROUTING_NUMBER);
```

In this example, the `KFSKeyConstants.ERROR_DV_BANK_ROUTING_NUMBER` parameter is passed in as the key that will be used to reference the associated error message in `ApplicationResources.properties`.

Note: *See Validation and Error Handling 2#Adding Error Messages for additional information regarding adding error messages to the error map.*

⚠ You may wish to verify that you have changed the correct line, since it is possible for there to be duplicate instances of the same message listed under different keys.

To add a new message

Add the appropriate values to `ApplicationResources.properties` and `KFSKeyConstants.java` files. (See above for details) Cite the new message using the line

```
MessageFormat.format( SpringContext.getBean(
KualiConfigurationService.class).getPropertyString( KFSKeyConstants.<KeyString> ),
Object[] arguments);
```

Note: *<`KeyString`> is the key defined in the `KFSKeyConstants.java` file.*

The property.files key in the configuration.properties file controls this behavior. It accepts a comma delimited list of URL's that are used by Spring's Resource loader. This format has a special protocol of classpath for resources that are on the classpath (See the javadoc for org.springframework.core.io.DefaultResourceLoader). The remainder of the URL being the relative location with respect to the classpath. Use the file protocol for resources that are on the file system, but not on the classpath. Changing or adding to the value of this key will change where Kuali looks for these properties. See rfc3986 for details on URL's.

Struts Messages

The `struts.message.resources` value in the `configuration.properties` file is used to define a comma delimited list of properties files that contain messages that struts can provide to jsp tags. The items in this list implement the `java.util.PropertyResourceBundle` conventi on. The usual value is located in `ApplicationResources.properties` and so struts (in version 1.x, this behavior is changed slightly in version 2.x) will use the `ApplicationResources.properties` file to find message Strings. To change the message, modify the String value in `ApplicationResources.properties` for the appropriate key.

The key is determined by inspecting the jsp tag. For example:

> In this code excerpt
>
> ```
> <a class="portal_link" href="<bean:message key="app.feedback.link"/>"
> target="_blank" title="<bean:message key="app.feedback.linkText"
> />"><bean:message key="app.feedback.linkText" /></a>
> ```
>
> the keys in this html statement are `'app.feedback.link'` and `'app.feedback.linkText'`.

As before, if the message is known, `ApplicationResources.properties` may be searched and directly modified.

> The same caveats for KeyConstants apply: messages may contain `java.text.MessageFormat` placeholders and therefore, there may be duplicate messages.

To create a new message, add an appropriate entry in `ApplicationResources.properties` and use it in the jsp tag as shown below:

```
<bean:message key="your.new.key">
```

Unable to render {include} The included page could not be found.