

Migration Guide

Migrating existing projects to RichFaces 4.0.0.ALPHA2 (*draft*)

by Sean Rogers (Red Hat)

DRAFT

1. Introduction	1
2. Upgrading	3
2.1. Before upgrading	3
2.2. Installation	3
3. Changes and new features	5
3.1. JSF 2 integration	5
3.2. Standardized attributes	5
3.2.1. Name changes	5
3.2.2. Deprecated attributes	7
3.3. Performance	8
3.3.1. Bandwidth for generated markup	8
3.3.2. Resource requirements	8
3.4. Feature changes	8
3.4.1. Server-side process and render mechanisms	8
3.4.2. onbegin event attribute	9
3.4.3. <a4j:commandButton> and <a4j:commandLink>	9
3.4.4. <a4j:log>	9
3.4.5. <a4j:message> and <a4j:messages>	9
3.4.6. <a4j:outputPanel>	9
3.4.7. <a4j:push> and <a4j:poll>	10
3.4.8. <a4j:status>	10
3.5. New components and behaviors	10
3.5.1. <rich:busyBehavior>	10
3.6. Deprecated components and behaviors	10
3.6.1. <a4j:page>	10
A. Revision History	11

Introduction

RichFaces 4.0 introduces many improvements and refinements to the RichFaces framework. Migrating your RichFaces projects to version 4.0 brings several advantages to your applications, including:

- complete integration with JavaServer Faces (JSF) 2.0, using new features and standardizing components;
- performance improvements throughout the components and core features;
- consistency throughout the framework, forming conventions and standards; and
- simplification of project creation, build procedures, and application extension.

This book guides you through the process of migrating an existing project to RichFaces 4.0, and highlights important changes that have been made to the framework and how they may affect your projects and applications.

Upgrading

Upgrading an existing project to use the RichFaces 4.0 framework only requires that the new libraries replace the old ones. For a complete installation guide for new projects, refer to the *Developer Guide*.

2.1. Before upgrading

Ensure you make a back-up copy of any projects and settings before upgrading an existing RichFaces installation.

Check that the environment you are working in meets the RichFaces technical requirements listed in the *Developer Guide*.

2.2. Installation

1. Download RichFaces packages

If you have not already done so, download the RichFaces binaries from the RichFaces downloads page at <http://www.jboss.org/richfaces/download.html>.

2. Install libraries

Copy the following libraries from the RichFaces package to your application's WEB-INF/lib libraries directory:

- richfaces-api
- richfaces-impl
- core-ui

Existing projects should already contain the following libraries; if not, they will need to be included in the libraries directory as well:

- commons-logging
- commons-beanutils
- slf4j-api
- slf4j-log4j12

3. Set up Maven

For existing projects, you should already have a Maven installation set up for RichFaces. If not, refer to the *Developer Guide* for how to set up Maven.

4. Namespace and taglib declaration

The namespace and library inclusions are the same as those for previous versions of RichFaces.

The RichFaces namespaces need to be declared in the project's XHTML pages.

```
<ui:composition xmlns:a4j="http://richfaces.org/a4j" xmlns:rich="http://richfaces.org/rich">  
  ...  
</ui:composition>
```



Note

With RichFaces 4.0, it is no longer necessary to modify the `web.xml` and `faces-config.xml` files of a project to use the framework.

Changes and new features

There have been several changes made to the RichFaces framework for version 4.0 that may affect migrated projects.

3.1. JSF 2 integration

RichFaces 4.0 has full support for JavaServer Faces (JSF) 2. Due to limited support for JavaServer Pages (JSP) in JSF2, RichFaces no longer fully supports JSP either.

3.2. Standardized attributes

Several component and event attribute names have been changed from those used in previous versions of RichFaces. The names have been changed to be more consistent with attribute names used by JavaServer Faces 2, so as to provide standardized usage. Many of the name changes also provide more consistency and cleanliness within the framework itself. Other attributes have been deprecated in favor of consistent functionality.

3.2.1. Name changes

process

Changed to execute in 4.0.

Affected a4j components

<code><a4j:support></code>	<code><a4j:form></code>	<code><a4j:push></code>
<code><a4j:commandButton></code>	<code><a4j:jsFunction></code>	
<code><a4j:commandLink></code>	<code><a4j:poll></code>	

Affected rich components

<code><rich:calendar></code>	<code><rich:panelMenuGroup></code>	<code><rich:tab></code>
<code><rich:dataFilterSlider></code>	<code><rich:panelMenuItem></code>	<code><rich:toggleControl></code>
<code><rich:dataScroller></code>	<code><rich:progressBar></code>	<code><rich:tree></code>
<code><rich:dropSupport></code>	<code><rich:simpleTogglePanel></code>	<code><rich:treeNode></code>
<code><rich:menuItem></code>	<code><rich:suggestionBox></code>	

reRender

Changed to render in 4.0.

Affected a4j components

<code><a4j:support></code>	<code><a4j:form></code>	<code><a4j:push></code>
<code><a4j:commandButton></code>	<code><a4j:jsFunction></code>	
<code><a4j:commandLink></code>	<code><a4j:poll></code>	

Affected rich components

<rich:ajaxValidator>	<rich:dropSupport>	<rich:scrollableDataTable>
<rich:calendar>	<rich:extendedDataTable>	<rich:simpleTogglePanel>
<rich:dataFilterSlider>	<rich:menuItem>	<rich:suggestionBox>
<rich:dataScroller>	<rich:panelMenuGroup>	<rich:tab>
<rich:dataTable>	<rich:panelMenuItem>	<rich:tree>
<rich:dragSupport>	<rich:progressBar>	<rich:treeNode>

eventsQueue

Changed to queue in 4.0.

Affected a4j components

<a4j:support>	<a4j:form>	<a4j:push>
<a4j:commandButton>	<a4j:jsFunction>	
<a4j:commandLink>	<a4j:poll>	

Affected rich components

<rich:ajaxValidator>	<rich:menuItem>	<rich:suggestionBox>
<rich:calendar>	<rich:panelMenuGroup>	<rich:tab>
<rich:dataFilterSlider>	<rich:panelMenuItem>	<rich:toggleControl>
<rich:dataScroller>	<rich:progressBar>	<rich:tree>
<rich:dragSupport>	<rich:scrollableDataTable>	
<rich:dropSupport>	<rich:simpleTogglePanel>	

limitToList

Changed to limitRender in 4.0.

Affected a4j components

<a4j:support>	<a4j:form>	<a4j:push>
<a4j:commandButton>	<a4j:jsFunction>	
<a4j:commandLink>	<a4j:poll>	

Affected rich components

<rich:ajaxValidator>	<rich:menuItem>	<rich:suggestionBox>
<rich:calendar>	<rich:panelMenuGroup>	<rich:tab>
<rich:dataFilterSlider>	<rich:panelMenuItem>	<rich:toggleControl>
<rich:dataScroller>	<rich:progressBar>	<rich:tree>
<rich:dragSupport>	<rich:scrollableDataTable>	<rich:treeNode>
<rich:dropSupport>	<rich:simpleTogglePanel>	

onbeforedomupdate

Changed to onSuccess in 4.0.

Affected a4j components

All components.

Affected rich components
All components.

3.2.2. Deprecated attributes

ignoreDupResponses

Deprecated. Functionality moved to queue in 4.0.

Affected a4j components

<a4j:support>	<a4j:form>	<a4j:push>
<a4j:commandButton>	<a4j:jsFunction>	<a4j:queue>
<a4j:commandLink>	<a4j:poll>	

Affected rich components

<rich:ajaxValidator>	<rich:menuItem>	<rich:suggestionBox>
<rich:calendar>	<rich:panelMenuGroup>	<rich:tab>
<rich:dataFilterSlider>	<rich:panelMenuItem>	<rich:toggleControl>
<rich:dataScroller>	<rich:progressBar>	<rich:tree>
<rich:dragSupport>	<rich:scrollableDataTable>	<rich:treeNode>
<rich:dropSupport>	<rich:simpleTogglePanel>	

requestDelay

Deprecated. Functionality moved to queue in 4.0.

Affected a4j components

<a4j:support>	<a4j:commandLink>	<a4j:jsFunction>
<a4j:commandButton>	<a4j:form>	<a4j:queue>

Affected rich components

<rich:ajaxValidator>	<rich:dropSupport>	<rich:simpleTogglePanel>
<rich:calendar>	<rich:menuItem>	<rich:suggestionBox>
<rich:dataFilterSlider>	<rich:panelMenuGroup>	<rich:tab>
<rich:dataScroller>	<rich:panelMenuItem>	<rich:tree>
<rich:dragSupport>	<rich:scrollableDataTable>	

ajaxSingle

Deprecated. Use execute="@this" in 4.0 instead.

Affected a4j components

<a4j:support>	<a4j:form>	<a4j:push>
<a4j:commandButton>	<a4j:jsFunction>	
<a4j:commandLink>	<a4j:poll>	

Affected rich components

<rich:calendar>	<rich:panelMenuGroup>	<rich:toggleControl>
-----------------	-----------------------	----------------------

```
<rich:dataFilterSlider> <rich:panelMenuItem> <rich:toolTip>
<rich:dataScroller> <rich:progressBar> <rich:tree>
<rich:dropSupport> <rich:simpleTogglePanel><rich:treeNode>
<rich:fileUpload> <rich:suggestionBox>
<rich:menuItem> <rich:tab>
```

3.3. Performance

The RichFaces framework has been extensively refined to provide better performance for applications. This includes improvements to bandwidth requirements for generated markup, size and number of required resources for Javascript and Cascading Style Sheets (CSS), and other general performance improvements.

3.3.1. Bandwidth for generated markup

Several considerations have been made to improve the use of bandwidth when generating markup.

- XHTML markup has been simplified and streamlined using best practices, such as avoiding tables for layout.
- Default attribute values are not rendered.
- Javascript use has been simplified with shorter function names and only one proxy call per form.

3.3.2. Resource requirements

The three Javascript libraries previously used by the RichFaces framework have been consolidated and standardized into a single library for better performance.

Usage of Cascading Style Sheets (CSS) has been simplified, trimming out duplicate or otherwise unnecessary entries and better utilizing the cascading approach for style values.

3.4. Feature changes

Features have been altered or added to existing components and behaviors in the RichFaces 4.0 framework. Refer to the *Developer Guide* for full details on how to make use of any new features.

3.4.1. Server-side process and render mechanisms

Some of the mechanisms for server-side processing and rendering have been altered.

3.4.1.1. Regions

It is now possible to define zones for processing on the server-side without specifying execute definitions for every component. The `execute="@region"` property can be used to process regions defined on the server side.



Important

This approach is different to that in JSF 2, which uses default values of `execute="@this"` and `render="@this"` if no other value is defined.

3.4.1.2. `execute`

The `execute` property has been extended to include the `@region` keyword, which facilitates the region rendering changes described in [Section 3.4.1.1, "Regions"](#).

3.4.1.3. `render`

The `render` property can also be defined from the server side, using actions and listeners.

3.4.2. `onbegin` event attribute

The `onbegin` attribute has been added to support JSF events with the type `begin`.

3.4.3. `<a4j:commandButton>` and `<a4j:commandLink>`

Both the `<a4j:commandButton>` and `<a4j:commandLink>` components use the `onclick` event attribute instead of the `onbegin` attribute.

3.4.4. `<a4j:log>`

The `<a4j:log>` component has had several features modified.

- It is now possible to switch levels on the client side.
- The log can now be displayed inline with page markup, in a new browser window, or in the browser console.
- An application context parameter can now be defined to log all application pages.

3.4.5. `<a4j:message>` and `<a4j:messages>`

Both the `<a4j:message>` and `<a4j:messages>` components can be automatically updated by any Ajax request, except those requests with `limitRender = true`. Attributes from previously-passed states are now cleared.

3.4.6. `<a4j:outputPanel>`

The `<a4j:outputPanel>` component has had several features modified.

- The default value for the `layout` attribute is now `block`.

- The default value for the `ajaxRendered` attribute is now `false`.
- If a child component is not to be rendered, a placeholder element will be included to preserve the layout.

3.4.7. `<a4j:push>` and `<a4j:poll>`

Both the `<a4j:push>` and `<a4j:poll>` components now raise a client-side event. For `<a4j:push>`, this is the `ondataavailable` event, while for `<a4j:poll>` it is the `ontimeevent`.

3.4.8. `<a4j:status>`

The `<a4j:status>` component has had several features modified.

- The `<a4j:status>` component now has a representation in the Document Object Model (DOM) tree.
- The `<a4j:status>` component no longer needs to be associated with a `<a4j:region>`.
- The `<a4j:status>` component can now report request errors. An additional `errorText` attribute and error facet have been added to show if an error was raised when a request completed.
- The event attributes now include `onstart`, `onstop`, `onerror`, and `onsuccess`.
- Multiple statuses can be presented for the same view or form, or a separate status attached to each component.

3.5. New components and behaviors

Several new components and behaviors have been added to RichFaces 4.0, some of which replace the functionality of deprecated elements. For full details on how to use these new components and behaviors, refer to the *Developer Guide*.

3.5.1. `<rich:busyBehavior>`

The `<rich:busyBehavior>` component is used when heavy processing takes place to provide feedback to the user. It can be attached to a component's events to trigger appropriate messages.

3.6. Deprecated components and behaviors

Some components and behaviors have been deprecated in RichFaces 4.0. For the most part these items have their functionality replicated by another component or behavior.

3.6.1. `<a4j:page>`

RichFaces 4.0 drops support for the `<a4j:page>` component. The component was previously used for solving incompatibility in the JavaServer Pages (JSP) environment with Apache MyFaces in early Ajax4jsf versions.

Appendix A. Revision History

Revision History

Revision 0.1

Tue Oct 20 2009

SeanRogers<serogers@redhat.com>

Basic first draft

Revision 0.2

Fri May 07 2010

SeanRogers<serogers@redhat.com>

Revised draft

