



# BEA WebLogic Workshop™ Help

Version 8.1 SP2  
November 2003

# Copyright

Copyright © 2003 BEA Systems, Inc. All Rights Reserved.

## Restricted Rights Legend

This software and documentation is subject to and made available only pursuant to the terms of the BEA Systems License Agreement and may be used or copied only in accordance with the terms of that agreement. It is against the law to copy the software except as specifically allowed in the agreement. This document may not, in whole or in part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine readable form without prior consent, in writing, from BEA Systems, Inc.

Use, duplication or disclosure by the U.S. Government is subject to restrictions set forth in the BEA Systems License Agreement and in subparagraph (c)(1) of the Commercial Computer Software–Restricted Rights Clause at FAR 52.227–19; subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227–7013, subparagraph (d) of the Commercial Computer Software—Licensing clause at NASA FAR supplement 16–52.227–86; or their equivalent.

Information in this document is subject to change without notice and does not represent a commitment on the part of BEA Systems. THE SOFTWARE AND DOCUMENTATION ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. FURTHER, BEA Systems DOES NOT WARRANT, GUARANTEE, OR MAKE ANY REPRESENTATIONS REGARDING THE USE, OR THE RESULTS OF THE USE, OF THE SOFTWARE OR WRITTEN MATERIAL IN TERMS OF CORRECTNESS, ACCURACY, RELIABILITY, OR OTHERWISE.

## Trademarks or Service Marks

BEA, Jolt, Tuxedo, and WebLogic are registered trademarks of BEA Systems, Inc. BEA Builder, BEA Campaign Manager for WebLogic, BEA eLink, BEA Liquid Data for WebLogic, BEA Manager, BEA WebLogic Commerce Server, BEA WebLogic Enterprise, BEA WebLogic Enterprise Platform, BEA WebLogic Enterprise Security, BEA WebLogic Express, BEA WebLogic Integration, BEA WebLogic Personalization Server, BEA WebLogic Platform, BEA WebLogic Portal, BEA WebLogic Server, BEA WebLogic Workshop and How Business Becomes E–Business are trademarks of BEA Systems, Inc.

All other trademarks are the property of their respective companies.

# Table of Contents

<b>Page Flow API Reference.....</b>	<b>1</b>
<b>com.bea.wlw.netui.databinding.form Package.....</b>	<b>3</b>
com.bea.wlw.netui.databinding.form RowSetForm Class.....	4
com.bea.wlw.netui.databinding.form RowSetFormException Class.....	10
com.bea.wlw.netui.databinding.jsp Package.....	13
com.bea.wlw.netui.databinding.jsp FilterWindowService Class.....	14
com.bea.wlw.netui.databinding.jsp FilterWindowService.Filter Class.....	19
com.bea.wlw.netui.databinding.jsp FilterWindowService.FilterWindowData Class.....	21
com.bea.wlw.netui.pageflow Package.....	26
com.bea.wlw.netui.pageflow ActionNotFoundException Class.....	29
com.bea.wlw.netui.pageflow ActionResult Interface.....	31
com.bea.wlw.netui.pageflow AutoRegisterActionServlet Class.....	33
com.bea.wlw.netui.pageflow AutoRegisterActionServlet.ModuleConfigLocator Interface.....	40
com.bea.wlw.netui.pageflow DynaFormData Class.....	41
com.bea.wlw.netui.pageflow DynamicSubappActionServlet Class.....	45
com.bea.wlw.netui.pageflow EmptyNestingStackException Class.....	48
com.bea.wlw.netui.pageflow FlowController Class.....	50
com.bea.wlw.netui.pageflow FlowControllerFactory Class.....	66
com.bea.wlw.netui.pageflow FormData Class.....	70
com.bea.wlw.netui.pageflow Forward Class.....	72
com.bea.wlw.netui.pageflow GlobalApp Class.....	84
com.bea.wlw.netui.pageflow HttpRedirector Interface.....	88
com.bea.wlw.netui.pageflow IllegalOutputFormException Class.....	90

# Table of Contents

<b>com.bea.wlw.netui.pageflow IllegalOutputFormTypeException Class.....</b>	<b>93</b>
<b>com.bea.wlw.netui.pageflow IllegalPageInputException Class.....</b>	<b>96</b>
<b>com.bea.wlw.netui.pageflow IllegalRedirectOutputFormException Class.....</b>	<b>100</b>
<b>com.bea.wlw.netui.pageflow InfiniteReturnToActionException Class.....</b>	<b>102</b>
<b>com.bea.wlw.netui.pageflow LoginExpiredException Class.....</b>	<b>104</b>
<b>com.bea.wlw.netui.pageflow NoCurrentPageFlowException Class.....</b>	<b>106</b>
<b>com.bea.wlw.netui.pageflow NoMatchingActionMethodException Class.....</b>	<b>108</b>
<b>com.bea.wlw.netui.pageflow NoPreviousActionException Class.....</b>	<b>110</b>
<b>com.bea.wlw.netui.pageflow NoPreviousPageException Class.....</b>	<b>112</b>
<b>com.bea.wlw.netui.pageflow NotLoggedInException Class.....</b>	<b>114</b>
<b>com.bea.wlw.netui.pageflow PageFlowActionServlet Class.....</b>	<b>116</b>
<b>com.bea.wlw.netui.pageflow PageFlowActionServlet.DefaultModuleConfigLocator Class.....</b>	<b>121</b>
<b>com.bea.wlw.netui.pageflow PageFlowActionServlet.LegacyModuleConfigLocator Class.....</b>	<b>123</b>
<b>com.bea.wlw.netui.pageflow PageFlowConstants Interface.....</b>	<b>125</b>
<b>com.bea.wlw.netui.pageflow PageFlowContextListener Class.....</b>	<b>130</b>
<b>com.bea.wlw.netui.pageflow PageFlowController Class.....</b>	<b>132</b>
<b>com.bea.wlw.netui.pageflow PageFlowException Class.....</b>	<b>142</b>
<b>com.bea.wlw.netui.pageflow PageFlowFileFilter Class.....</b>	<b>146</b>
<b>com.bea.wlw.netui.pageflow PageFlowRequestProcessor Class.....</b>	<b>148</b>
<b>com.bea.wlw.netui.pageflow PageFlowRequestProcessor.DefaultHttpRedirector Class.....</b>	<b>157</b>
<b>com.bea.wlw.netui.pageflow PageFlowRequestProcessor.ExceptionHandledAction Class.....</b>	<b>159</b>
<b>com.bea.wlw.netui.pageflow PageFlowRequestProcessor.ExceptionHandledActionMapping Class...</b>	<b>161</b>
<b>com.bea.wlw.netui.pageflow PageFlowRequestProcessor.GlobalAppActionMapping Class.....</b>	<b>164</b>

# Table of Contents

<b>com.bea.wlw.netui.pageflow PageFlowRequestProcessor.GlobalAppForwardAction Class.....</b>	<b>167</b>
<b>com.bea.wlw.netui.pageflow PageFlowStack Class.....</b>	<b>169</b>
<b>com.bea.wlw.netui.pageflow PageFlowUtils Class.....</b>	<b>172</b>
<b>com.bea.wlw.netui.pageflow PreviousActionInfo Class.....</b>	<b>192</b>
<b>com.bea.wlw.netui.pageflow PreviousPageInfo Class.....</b>	<b>195</b>
<b>com.bea.wlw.netui.pageflow ProcessPopulate.ExpressionUpdateNode Class.....</b>	<b>198</b>
<b>com.bea.wlw.netui.pageflow RequestParameterHandler Interface.....</b>	<b>200</b>
<b>com.bea.wlw.netui.pageflow UnfulfilledRolesException Class.....</b>	<b>201</b>
<b>com.bea.wlw.netui.pageflow UnresolvableForwardException Class.....</b>	<b>204</b>
<b>com.bea.wlw.netui.pageflow.scoping Package.....</b>	<b>207</b>
<b>com.bea.wlw.netui.pageflow.scoping ScopedRequest Interface.....</b>	<b>208</b>
<b>com.bea.wlw.netui.pageflow.scoping ScopedResponse Interface.....</b>	<b>213</b>
<b>com.bea.wlw.netui.pageflow.scoping ScopedServletContext Interface.....</b>	<b>218</b>
<b>com.bea.wlw.netui.pageflow.scoping ScopedServletUtils Class.....</b>	<b>219</b>
<b>com.bea.wlw.netui.pageflow.util Package.....</b>	<b>226</b>
<b>com.bea.wlw.netui.pageflow.util DefaultURLRewriter Class.....</b>	<b>227</b>
<b>com.bea.wlw.netui.pageflow.util TemplateHelper Class.....</b>	<b>230</b>
<b>com.bea.wlw.netui.pageflow.util URLRewriter Class.....</b>	<b>241</b>
<b>com.bea.wlw.netui.pageflow.util URLRewriterService Class.....</b>	<b>246</b>
<b>com.bea.wlw.netui.tags Package.....</b>	<b>252</b>
<b>com.bea.wlw.netui.tags AbstractBaseTag Class.....</b>	<b>253</b>
<b>com.bea.wlw.netui.tags AbstractPageError Class.....</b>	<b>263</b>
<b>com.bea.wlw.netui.tags DataAccessProvider Interface.....</b>	<b>265</b>

# Table of Contents

<b>com.bea.wlw.netui.tags DataSourceTag Class.....</b>	<b>268</b>
<b>com.bea.wlw.netui.tags DefaultableDataSourceTag Class.....</b>	<b>272</b>
<b>com.bea.wlw.netui.tags ErrorReporter Interface.....</b>	<b>276</b>
<b>com.bea.wlw.netui.tags EvalErrorInfo Class.....</b>	<b>278</b>
<b>com.bea.wlw.netui.tags IAttributeConsumer Interface.....</b>	<b>280</b>
<b>com.bea.wlw.netui.tags JavaScriptUtils Class.....</b>	<b>281</b>
<b>com.bea.wlw.netui.tags OptionsDataSourceTag Class.....</b>	<b>285</b>
<b>com.bea.wlw.netui.tags ScriptReporter Interface.....</b>	<b>289</b>
<b>com.bea.wlw.netui.tags TagErrorInfo Class.....</b>	<b>291</b>
<b>com.bea.wlw.netui.tags TagValue Interface.....</b>	<b>293</b>
<b>com.bea.wlw.netui.tags.databinding Package.....</b>	<b>294</b>
<b>com.bea.wlw.netui.tags.databinding SortFilterService Class.....</b>	<b>295</b>
<b>com.bea.wlw.netui.tags.databinding.bundle Package.....</b>	<b>306</b>
<b>com.bea.wlw.netui.tags.databinding.bundle BundleContext.BundleNode Class.....</b>	<b>307</b>
<b>com.bea.wlw.netui.tags.databinding.bundle BundleContext.NetUIBundleNode Class.....</b>	<b>309</b>
<b>com.bea.wlw.netui.tags.databinding.bundle BundleContext.StrutsBundleNode Class.....</b>	<b>311</b>
<b>com.bea.wlw.netui.tags.databinding.bundle DeclareBundle Class.....</b>	<b>313</b>
<b>com.bea.wlw.netui.tags.databinding.cellrepeater Package.....</b>	<b>322</b>
<b>com.bea.wlw.netui.tags.databinding.cellrepeater CellRepeater Class.....</b>	<b>323</b>
<b>com.bea.wlw.netui.tags.databinding.grid Package.....</b>	<b>337</b>
<b>com.bea.wlw.netui.tags.databinding.grid Columns Class.....</b>	<b>338</b>
<b>com.bea.wlw.netui.tags.databinding.grid Grid Class.....</b>	<b>346</b>
<b>com.bea.wlw.netui.tags.databinding.grid GridComponent Class.....</b>	<b>361</b>

# Table of Contents

<b>com.bea.wlw.netui.tags.databinding.grid GridContext Class.....</b>	<b>366</b>
<b>com.bea.wlw.netui.tags.databinding.grid.column Package.....</b>	<b>371</b>
<b>com.bea.wlw.netui.tags.databinding.grid.column AnchorColumn Class.....</b>	<b>372</b>
<b>com.bea.wlw.netui.tags.databinding.grid.column BasicColumn Class.....</b>	<b>382</b>
<b>com.bea.wlw.netui.tags.databinding.grid.column ExpressionColumn Class.....</b>	<b>388</b>
<b>com.bea.wlw.netui.tags.databinding.grid.column FormattableColumn Class.....</b>	<b>395</b>
<b>com.bea.wlw.netui.tags.databinding.grid.column GridColumn Class.....</b>	<b>400</b>
<b>com.bea.wlw.netui.tags.databinding.grid.column ImageColumn Class.....</b>	<b>409</b>
<b>com.bea.wlw.netui.tags.databinding.grid.column SortFilterColumn Class.....</b>	<b>419</b>
<b>com.bea.wlw.netui.tags.databinding.grid.pager Package.....</b>	<b>430</b>
<b>com.bea.wlw.netui.tags.databinding.grid.pager Pager Class.....</b>	<b>431</b>
<b>com.bea.wlw.netui.tags.databinding.grid.style Package.....</b>	<b>439</b>
<b>com.bea.wlw.netui.tags.databinding.grid.style GridStyle Class.....</b>	<b>440</b>
<b>com.bea.wlw.netui.tags.databinding.grid.style GridStyleContext Class.....</b>	<b>445</b>
<b>com.bea.wlw.netui.tags.databinding.invoke Package.....</b>	<b>448</b>
<b>com.bea.wlw.netui.tags.databinding.invoke AbstractCallMethod Class.....</b>	<b>449</b>
<b>com.bea.wlw.netui.tags.databinding.invoke CallControl Class.....</b>	<b>458</b>
<b>com.bea.wlw.netui.tags.databinding.invoke CallMethod Class.....</b>	<b>464</b>
<b>com.bea.wlw.netui.tags.databinding.invoke CallPageFlow Class.....</b>	<b>470</b>
<b>com.bea.wlw.netui.tags.databinding.invoke DeclareControl Class.....</b>	<b>475</b>
<b>com.bea.wlw.netui.tags.databinding.invoke MethodParameter Class.....</b>	<b>481</b>
<b>com.bea.wlw.netui.tags.databinding.invoke ObjectNotFoundException Class.....</b>	<b>488</b>
<b>com.bea.wlw.netui.tags.databinding.message Package.....</b>	<b>491</b>

# Table of Contents

<b>com.bea.wlw.netui.tags.databinding.message Message Class.....</b>	<b>492</b>
<b>com.bea.wlw.netui.tags.databinding.message MessageArg Class.....</b>	<b>497</b>
<b>com.bea.wlw.netui.tags.databinding.pageinput Package.....</b>	<b>501</b>
<b>com.bea.wlw.netui.tags.databinding.pageinput DeclarePageInput Class.....</b>	<b>502</b>
<b>com.bea.wlw.netui.tags.databinding.repeater Package.....</b>	<b>507</b>
<b>com.bea.wlw.netui.tags.databinding.repeater Repeater Class.....</b>	<b>508</b>
<b>com.bea.wlw.netui.tags.databinding.repeater RepeaterComponent Class.....</b>	<b>519</b>
<b>com.bea.wlw.netui.tags.databinding.repeater RepeaterFooter Class.....</b>	<b>525</b>
<b>com.bea.wlw.netui.tags.databinding.repeater RepeaterHeader Class.....</b>	<b>530</b>
<b>com.bea.wlw.netui.tags.databinding.repeater RepeaterItem Class.....</b>	<b>535</b>
<b>com.bea.wlw.netui.tags.databinding.repeater.choice Package.....</b>	<b>543</b>
<b>com.bea.wlw.netui.tags.databinding.repeater.choice Choice Class.....</b>	<b>544</b>
<b>com.bea.wlw.netui.tags.databinding.repeater.choice ChoiceMethod Class.....</b>	<b>552</b>
<b>com.bea.wlw.netui.tags.databinding.repeater.pad Package.....</b>	<b>560</b>
<b>com.bea.wlw.netui.tags.databinding.repeater.pad Pad Class.....</b>	<b>561</b>
<b>com.bea.wlw.netui.tags.databinding.repeater.pad PadContext Class.....</b>	<b>569</b>
<b>com.bea.wlw.netui.tags.databinding.script Package.....</b>	<b>572</b>
<b>com.bea.wlw.netui.tags.databinding.script GetData Class.....</b>	<b>573</b>
<b>com.bea.wlw.netui.tags.html Package.....</b>	<b>579</b>
<b>com.bea.wlw.netui.tags.html Anchor Class.....</b>	<b>581</b>
<b>com.bea.wlw.netui.tags.html Attribute Class.....</b>	<b>605</b>
<b>com.bea.wlw.netui.tags.html Base Class.....</b>	<b>610</b>
<b>com.bea.wlw.netui.tags.html BindingUpdateErrors Class.....</b>	<b>616</b>



# Table of Contents

<b>com.bea.wlw.netui.tags.html Button Class.....</b>	<b>621</b>
<b>com.bea.wlw.netui.tags.html CheckBox Class.....</b>	<b>643</b>
<b>com.bea.wlw.netui.tags.html CheckBoxGroup Class.....</b>	<b>663</b>
<b>com.bea.wlw.netui.tags.html CheckBoxOption Class.....</b>	<b>673</b>
<b>com.bea.wlw.netui.tags.html Content Class.....</b>	<b>695</b>
<b>com.bea.wlw.netui.tags.html Error Class.....</b>	<b>700</b>
<b>com.bea.wlw.netui.tags.html Errors Class.....</b>	<b>707</b>
<b>com.bea.wlw.netui.tags.html Exceptions Class.....</b>	<b>713</b>
<b>com.bea.wlw.netui.tags.html FileUpload Class.....</b>	<b>718</b>
<b>com.bea.wlw.netui.tags.html Form Class.....</b>	<b>737</b>
<b>com.bea.wlw.netui.tags.html FormatDate Class.....</b>	<b>762</b>
<b>com.bea.wlw.netui.tags.html FormatNumber Class.....</b>	<b>767</b>
<b>com.bea.wlw.netui.tags.html FormatString Class.....</b>	<b>772</b>
<b>com.bea.wlw.netui.tags.html Formattable Interface.....</b>	<b>778</b>
<b>com.bea.wlw.netui.tags.html FormatTag Class.....</b>	<b>779</b>
<b>com.bea.wlw.netui.tags.html GetNetuiTagName Class.....</b>	<b>785</b>
<b>com.bea.wlw.netui.tags.html Hidden Class.....</b>	<b>789</b>
<b>com.bea.wlw.netui.tags.html Html Class.....</b>	<b>797</b>
<b>com.bea.wlw.netui.tags.html Image Class.....</b>	<b>805</b>
<b>com.bea.wlw.netui.tags.html ImageAnchor Class.....</b>	<b>829</b>
<b>com.bea.wlw.netui.tags.html ImageButton Class.....</b>	<b>845</b>
<b>com.bea.wlw.netui.tags.html Label Class.....</b>	<b>872</b>
<b>com.bea.wlw.netui.tags.html Node Class.....</b>	<b>890</b>

# Table of Contents

<b>com.bea.wlw.netui.tags.html Parameter Class.....</b>	<b>900</b>
<b>com.bea.wlw.netui.tags.html ParameterMap Class.....</b>	<b>906</b>
<b>com.bea.wlw.netui.tags.html RadioButtonGroup Class.....</b>	<b>911</b>
<b>com.bea.wlw.netui.tags.html RadioButtonOption Class.....</b>	<b>922</b>
<b>com.bea.wlw.netui.tags.html RewriteName Class.....</b>	<b>943</b>
<b>com.bea.wlw.netui.tags.html RewriteURL Class.....</b>	<b>948</b>
<b>com.bea.wlw.netui.tags.html ScriptContainer Class.....</b>	<b>953</b>
<b>com.bea.wlw.netui.tags.html Select Class.....</b>	<b>960</b>
<b>com.bea.wlw.netui.tags.html SelectOption Class.....</b>	<b>982</b>
<b>com.bea.wlw.netui.tags.html TextArea Class.....</b>	<b>998</b>
<b>com.bea.wlw.netui.tags.html TextBox Class.....</b>	<b>1019</b>
<b>com.bea.wlw.netui.tags.html Tree Class.....</b>	<b>1041</b>
<b>com.bea.wlw.netui.tags.html TreeNode Class.....</b>	<b>1057</b>
<b>com.bea.wlw.netui.tags.html URLParams Interface.....</b>	<b>1069</b>
<b>com.bea.wlw.netui.tags.template Package.....</b>	<b>1070</b>
<b>com.bea.wlw.netui.tags.template Attribute Class.....</b>	<b>1071</b>
<b>com.bea.wlw.netui.tags.template IncludeSection Class.....</b>	<b>1076</b>
<b>com.bea.wlw.netui.tags.template Section Class.....</b>	<b>1082</b>
<b>com.bea.wlw.netui.tags.template SetAttribute Class.....</b>	<b>1089</b>
<b>com.bea.wlw.netui.tags.template Template Class.....</b>	<b>1095</b>
<b>com.bea.wlw.netui.tags.template Template.TemplateContext Class.....</b>	<b>1102</b>
<b>com.bea.wlw.netui.tags.template TemplateConstants Interface.....</b>	<b>1103</b>
<b>com.bea.wlw.netui.tags.template Visible Class.....</b>	<b>1104</b>

# Table of Contents

<b>com.bea.wlw.netui.util Package.....</b>	<b>1110</b>
<b>com.bea.wlw.netui.util Bundle Class.....</b>	<b>1111</b>
<b>com.bea.wlw.netui.util HtmlExceptionFormatter Class.....</b>	<b>1113</b>
<b>com.bea.wlw.netui.util PopulateContextHelper Class.....</b>	<b>1115</b>
<b>com.bea.wlw.netui.util TemplateHelper Class.....</b>	<b>1117</b>
<b>com.bea.wlw.netui.util.iterator Package.....</b>	<b>1118</b>
<b>com.bea.wlw.netui.util.iterator ArrayIterator Class.....</b>	<b>1119</b>
<b>com.bea.wlw.netui.util.iterator AtomicObjectIterator Class.....</b>	<b>1121</b>
<b>com.bea.wlw.netui.util.iterator EmptyIterator Class.....</b>	<b>1123</b>
<b>com.bea.wlw.netui.util.iterator EnumerationIterator Class.....</b>	<b>1125</b>
<b>com.bea.wlw.netui.util.iterator IteratorFactory Class.....</b>	<b>1127</b>
<b>com.bea.wlw.netui.util.iterator IteratorFactoryException Class.....</b>	<b>1130</b>
<b>com.bea.wlw.netui.util.iterator MapIterator Class.....</b>	<b>1132</b>
<b>com.bea.wlw.netui.util.tags Package.....</b>	<b>1134</b>
<b>com.bea.wlw.netui.util.tags GroupOption Class.....</b>	<b>1135</b>
<b>com.bea.wlw.netui.util.type Package.....</b>	<b>1140</b>
<b>com.bea.wlw.netui.util.type TypeConversionException Class.....</b>	<b>1141</b>
<b>com.bea.wlw.netui.util.type TypeConverter Interface.....</b>	<b>1144</b>
<b>com.bea.wlw.netui.util.type TypeConverterNotFoundException Class.....</b>	<b>1145</b>
<b>com.bea.wlw.netui.util.type TypeUtils Class.....</b>	<b>1147</b>

# Page Flow API Reference

The page flow API provides classes and interfaces that implement a programming framework to build dynamic web applications. The Javadoc in these topics includes the page flow server API and the NetUI tag API.

## Topics Included in This Section

The following links go to the Javadoc summary page for each API package.

[com.bea.wlw.netui.databinding.form](#)

[com.bea.wlw.netui.databinding.jsp](#)

[com.bea.wlw.netui.pageflow](#)

[com.bea.wlw.netui.pageflow.scoping](#)

[com.bea.wlw.netui.pageflow.util](#)

[com.bea.wlw.netui.tags](#)

[com.bea.wlw.netui.tags.databinding](#)

[com.bea.wlw.netui.tags.databinding.bundle](#)

[com.bea.wlw.netui.tags.databinding.cellrepeater](#)

[com.bea.wlw.netui.tags.databinding.grid](#)

[com.bea.wlw.netui.tags.databinding.grid.column](#)

[com.bea.wlw.netui.tags.databinding.grid.pager](#)

[com.bea.wlw.netui.tags.databinding.grid.style](#)

[com.bea.wlw.netui.tags.databinding.invoke](#)

[com.bea.wlw.netui.tags.databinding.message](#)

[com.bea.wlw.netui.tags.databinding.pageinput](#)

[com.bea.wlw.netui.tags.databinding.repeater](#)

[com.bea.wlw.netui.tags.databinding.repeater.choice](#)

[com.bea.wlw.netui.tags.databinding.repeater.pad](#)

[com.bea.wlw.netui.tags.databinding.script](#)

`com.bea.wlw.netui.tags.html`

`com.bea.wlw.netui.tags.template`

`com.bea.wlw.netui.util`

`com.bea.wlw.netui.util.iterator`

`com.bea.wlw.netui.util.tags`

`com.bea.wlw.netui.util.type`

Related Topics

Getting Started with Page Flows

Using Form Beans to Encapsulate Data

Designing User Interfaces in JSPs

Using Data Binding Tags in Page Flows

Presenting Complex Data Sets in JSPs

JSP Tags Reference

# **com.bea.wlw.netui.databinding.form Package**

## **Class Summary**

**RowSetForm** A `RowSetForm` is a `FormData` base class that provides support for `RowSet` data sets.

## **Exception Summary**

**RowSetFormException** An exception thrown when an error occurs applying values from a `RowSet` to a `RowSetForm` or from a `RowSetForm` to a `RowSet`.

# RowSetForm Class

*public abstract class RowSetForm*

extends `FormData`

A `RowSetForm` is a `FormData` base class that provides support for `RowSet` data sets. This support allows an extension of this class to specify strongly-typed `JavaBean` properties that map to data in the `RowSet`; this base class is then able to populate these `JavaBean` properties from a `RowSet` and populate a `RowSet` from the `JavaBean` properties. This two-way population ability is important because of the `PageFlow` request lifecycle. If in an action a `RowSet` is created or is returned by a instance, the action form needs to be populated with values from this `RowSet`. This is done with one method, `RowSetForm.applyValuesToForm(RowSet)`. This method inspects the `RowSetMetaData` for the `RowSet` and for every column whose name matches a `JavaBean` property name on an instance of this class, that property is populated with the value from the `RowSet`. Note, the types in the `RowSet` and the type of the property in this form must match. Then, on a request, the parameters from the request are populated into an instance of this form. Then, the instance of the form can be used to populate a row in the `RowSet`. This can be done with one of two methods depending on how the `RowSet` is being used. The `RowSetForm.applyInsertValuesToRowSet(RowSet)` should be used when a new row is being added to the `RowSet`. The values will be inserted into the `RowSet`'s insert row. The `RowSetForm.applyUpdateValuesToRowSet(RowSet)` should be used when the "current" row in the `RowSet` should be updated. In both cases, the properties in the form that have changed as a result of being populated with request data will be applied to the `RowSet`. In order to populate a `RowSet` with exactly the data that has changed, a subclass of the `RowSetForm` can track the changes in the `JavaBean` setter methods by using the `RowSetForm.registerChange(String)` method. This method allows the `RowSetForm` base class to keep track of all of the setters which have been called before populating a `RowSet` with the changed data. Once a `RowSet` has been populated with the data, the list of changed values is cleared.

---

## Hierarchy

```

Object
  ActionForm
    ValidatorForm
      FormData
      RowSetForm
  
```

## All Implemented Interfaces

```

Serializable
  
```

---

## Field Summary

Fields from `org.apache.struts.validator.ValidatorForm`

`page`, `validatorResults`

**Fields from `org.apache.struts.action.ActionForm`**

`multipartRequestHandler`, `servlet`

## Constructor

### Summary

***RowSetForm()***

Construct a  
RowSetForm.

## Method

### Summary

public  
void ***applyInsertValuesToRowSet***(*RowSet rowSet*)  
Populate the provided RowSet with the properties that have changed since the last time the RowSetForm was initialized or `RowSetForm.clearChangedValues()` was called.

public  
void ***applyUpdateValuesToRowSet***(*RowSet rowSet*)  
Populate the provided RowSet with the properties that have changed since the last time the RowSetForm was initialized or `RowSetForm.clearChangedValues()` was called.

public  
void ***applyValuesToForm***(*RowSet rowSet*)  
Populate the JavaBean properties of the form with the column values from the first row in the RowSet.

public  
void ***clearChangedValues***()  
Clear the list of changed values.

protected  
final ***registerChange***(*String propertyName*)  
void Register a change on a property with the name `propertyName`.

protected  
Object ***writeForm***(*String columnName*, *Object rowSetValue*)  
A callback that is executed before a value is applied to the form from a RowSet.

protected  
Object ***writeRowSet***(*String columnName*, *Object formValue*, *Object oldRowSetValue*)  
A callback that is executed before a value is applied to a RowSet from one of the form's properties.

**Methods from `org.apache.struts.validator.ValidatorForm`**

`getPage`, `getResultValueMap`, `getValidatorResults`, `log`, `log`, `reset`,



`setPage, setValidatorResults, validate`

#### Methods from `org.apache.struts.action.ActionForm`

`getMultipartRequestHandler, getServlet, getServletWrapper, reset, reset, setMultipartRequestHandler, setServlet, validate, validate`

#### Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

## Constructor Detail

### RowSetForm

```
public RowSetForm()
```

Construct a RowSetForm.

## Method Detail

### applyInsertValuesToRowSet(RowSet) Method

```
public void applyInsertValuesToRowSet(RowSet rowSet)
    throws RowSetFormException
```

Populate the provided RowSet with the properties that have changed since the last time the RowSetForm was initialized or RowSetForm.clearChangedValues() was called. These values will be applied to the RowSet in the "insert" row. This method expects the RowSet to be in "default" state. Then, the values will be inserted into the RowSet and the RowSet's RowSet.insertRow() and RowSet.moveToCurrentRow() lifecycle methods are called.

#### Parameters

*rowSet*

the RowSet into which to insert the form's JavaBean properties

#### Exceptions

*RowSetFormException*

if an error occurs while inserting values into a RowSet from the changed properties.

---

## applyUpdateValuesToRowSet(RowSet) Method

```
public void applyUpdateValuesToRowSet(RowSet rowSet)
    throws RowSetFormException
```

Populate the provided RowSet with the properties that have changed since the last time the RowSetForm was initialized or RowSetForm.clearChangedValues() was called. These values will be applied to the RowSet in the "current" row. This method expects the RowSet to be in its "default" state. Then, the values will be set on the RowSet in order to update the values of the "current" row.

### Parameters

*rowSet*

the RowSet into which to insert the form's JavaBean properties

### Exceptions

*RowSetFormException*

if an error occurs while inserting values into a RowSet from the changed properties.

---

## applyValuesToForm(RowSet) Method

```
public void applyValuesToForm(RowSet rowSet)
    throws RowSetFormException
```

Populate the JavaBean properties of the form with the column values from the first row in the RowSet. This will populate the properties whose names match, in a case insensitive way, each column name in the RowSet's RowSetMetaData. The values that will populate the bean will be taken from the first row of the RowSet.

### Parameters

*rowSet*

the RowSet from which to read values that are set on the RowSetForm's JavaBean properties

### Exceptions

*RowSetFormException*

if an error occurs while applying values to the form from the RowSet

---

## clearChangedValues() Method

```
public void clearChangedValues()
```

Clear the list of changed values. This method can be used so that a RowSetForm instance may be reused multiple times for multiple RowSet instances.

---

## registerChange(String) Method

```
protected final void registerChange(String propertyName)
```

Register a change on a property with the name `propertyName`. This method should be called from any setter whose value changes should be tracked so that a `RowSet` is updated with the correctly changed values. Implementing a `JavaBean` setter that does not call this method will result in the changed value of the property never being updated on a `RowSet` if either `RowSetForm.applyUpdateValuesToRowSet(RowSet)` or `RowSetForm.applyInsertValuesToRowSet(RowSet)` are called

---

## writeForm(String, Object) Method

```
protected Object writeForm(String columnName,  
                             Object rowSetValue)
```

A callback that is executed before a value is applied to the form from a `RowSet`. When using an instance of a `RowSetForm` via the `RowSetForm.applyValuesToForm(RowSet)` method, this callback is invoked for each of the columns in the `RowSet` when the column's value is applied to the form.

Any changes to the type of a column's value done here should be paired with the inverse change in the `RowSetForm.writeRowSet(String, Object, Object)`. See `RowSetForm.writeRowSet(String, Object, Object)` for more information on how to use this feature.

Comparisons against the column name should be done using the `String.equalsIgnoreCase(String)` method as the capitalization of the column from the database and the name of the property in the form may be different.

### Parameters

*columnName*  
the name of the column that is being set

*rowSetValue*  
the value of the column in the `RowSet`

### Returns

*the object to set on the form*

---

## writeRowSet(String, Object, Object) Method

```
protected Object writeRowSet(String columnName,  
                             Object formValue,  
                             Object oldRowSetValue)
```

A callback that is executed before a value is applied to a `RowSet` from one of the form's properties. When using an instance of a `RowSetForm` via the `RowSetForm.applyInsertValuesToRowSet(RowSet)` or `RowSetForm.applyUpdateValuesToRowSet(RowSet)` methods, the properties of an extending

class are must be set on the RowSet. These are set based on a case-insensitive String comparison of the column names in the RowSet and the property names in the form. Each column that will be set on the RowSet is passed through this method using the value matching the column name from the form and that from the current row in the RowSet.

In this method, custom type conversion logic can be implemented to coerce the type of the form's value into one that the RowSet can handle. For example, the `Timestamp` class mandates a specific time format, but often a different format is used for entering dates in a web page. The value can be stored as a `String` in the form and converted into the `Timestamp` in this method by using a date formatter to switch between the two. Note, in order to set properties on the RowSet which are different types than the form, the `RowSetForm.writeForm(String, Object)` method must be implemented to convert from the database / RowSet specific type to the type in the form.

Comparisons against the column name should be done using the `String.equalsIgnoreCase(String)` method as the capitalization of the column from the database and the name of the property in the form may be different.

### Parameters

*columnName*

the name of the column that is being set

*formValue*

the value of this column's value from the form

*oldRowSetValue*

the value of this column in the RowSet

### Returns

*the value to set on the RowSet*

# RowSetFormException Class

*public class RowSetFormException*

extends Exception

An exception thrown when an error occurs applying values from a RowSet to a RowSetForm or from a RowSetForm to a RowSet.

---

## ***Hierarchy***

Object  
  Throwable  
    Exception  
      RowSetFormException

## ***All Implemented Interfaces***

Serializable

---

## **Constructor Summary**

### ***RowSetFormException()***

Construct a RowSetFormException.

### ***RowSetFormException(String message, Throwable cause)***

Construct a RowSetFormException with the given message and cause.

### ***RowSetFormException(String message)***

Construct a RowSetFormException with the given message.

### ***RowSetFormException(Throwable cause)***

Construct a RowSetFormException with the given cause.

## Method Summary

```
public
String getLocalizedMessage()
public
void setLocalizedMessage(String localizedMessage)
```

### Methods from `java.lang.Throwable`

```
fillInStackTrace, getCause, getMessage, getStackTrace, initCause,
printStackTrace, printStackTrace, printStackTrace, setStackTrace,
toString
```

### Methods from class `java.lang.Object`

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll,
toString, wait, wait, wait
```

## Constructor Detail

### RowSetFormException

```
public RowSetFormException()
```

Construct a RowSetFormException.

---

### RowSetFormException

```
public RowSetFormException(String message,
                           Throwable cause)
```

Construct a RowSetFormException with the given message and cause.

---

### RowSetFormException

```
public RowSetFormException(String message)
```

Construct a RowSetFormException with the given message.

---

### RowSetFormException

```
public RowSetFormException(Throwable cause)
```

Construct a RowSetFormException with the given cause.

## Method Detail

### **getLocalizedMessage() Method**

```
public String getLocalizedMessage()
```

#### *Overrides*

```
Throwable.getLocalizedMessage()
```

---

### **setLocalizedMessage(String) Method**

```
public void setLocalizedMessage(String localizedMessage)
```

# com.bea.wlw.netui.databinding.jsp Package

## Class Summary

FilterWindowService

The `FilterWindowService` is used to create a `FilterWindowService.FilterWindowData` object that contains state information that is used to render a filter window.

FilterWindowService.Filter

The `Filter` class contains information about filters that are applied to a grid/column pair for which the filter window is being rendered.

FilterWindowService.FilterWindowData

The `FilterWindowData` class contains information extracted from the current request and response which is used when rendering a filter window.



# FilterWindowService Class

*public final class FilterWindowService*

extends Object

The FilterWindowService is used to create a FilterWindowService.FilterWindowData object that contains state information that is used to render a filter window. The filter window is used to add parameters to a URL which, when submitted to the server, changes a query for data on the server. The filter window displays a type specific user interface depending on the type of the data being filtered. The FilterWindowService.isStringType(int), FilterWindowService.isDateType(int), FilterWindowService.isBooleanType(int) methods are used to determine the type of UI that will be rendered. The options available for filtering a specific column will also change depending on the column type.

---

## Hierarchy

Object  
FilterWindowService

---

## Nested Class Summary

public **FilterWindowService.Filter**  
static class

The Filter class contains information about filters that are applied to a grid/column pair for which the filter window is being rendered.

public static **FilterWindowService.FilterWindowData**  
final class

The FilterWindowData class contains information extracted from the current request and response which is used when rendering a filter window.

## Constructor Summary

*FilterWindowService()*

## Method Summary

```

        public static
FilterWindowService.FilterWindowData getFilterWindowData(HttpServletRequest
request, HttpServletResponse response)
    Factory method to create an instance of a
    FilterWindowService.FilterWindowData
    object that contains state from the request
    which is used when rendering a filter window.

        public static String[][]
getOptions()
    Get a list of options that are available for
    display in the filter drop downs of the filter
    window.

        public static String[][]
getOptions(Locale locale)
    Get a list of options that are available for
    display in the filter drop downs of the filter
    window.

        public static boolean
isBooleanType(int columnType)
    Check a given column type to see if it is a
    boolean column type.

        public static boolean
isDateType(int columnType)
    Check a given column type to see if it is a date
    column type.

        public static boolean
isStringType(int columnType)
    Check a given column type to see if it is a
    string column type.

```

### Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

## Constructor Detail

### FilterWindowService

```
public FilterWindowService()
```

## Method Detail

### getFilterWindowData(HttpServletRequest, HttpServletResponse) Method

```
public static FilterWindowService.FilterWindowData getFilterWindowData(HttpServletRequest request, HttpServletResponse response)
```

Factory method to create an instance of a `FilterWindowService.FilterWindowData` object that contains state from the request which is used when rendering a filter window.

#### Parameters

*request*  
the current request

*response*  
the current response

#### Returns

*a FilterWindowData object that represents necessary information that is parsed out of the URL for the filter window.*

---

### getOptions() Method

```
public static String[][] getOptions()
```

Get a list of options that are available for display in the filter drop downs of the filter window. Not all operations are available for all column types. The return value is a two dimensional array of Strings as:

```
(readable name of an operation, URL key of an operation)
```

#### Returns

*the list of filter operations. Using the default Locale for the JVM.*

---

### getOptions(Locale) Method

```
public static String[][] getOptions(Locale locale)
```

Get a list of options that are available for display in the filter drop downs of the filter window. Not all operations are available for all column types. The return value is a two dimensional array of Strings as:

```
(readable name of an operation, URL key of an operation)
```

## Returns

*the list of filter opations.*

---

## isBooleanType(int) Method

```
public static boolean isBooleanType(int columnType)
```

Check a given column type to see if it is a boolean column type. Boolean types include:

- 

### Parameters

*columnType*

the int representation of a column type

### Returns

*true if the column is a boolean type; false otherwise*

---

## isDateType(int) Method

```
public static boolean isDateType(int columnType)
```

Check a given column type to see if it is a date column type. Date types include:

- 

### Parameters

*columnType*

the int representation of a column type

### Returns

*true if the column is a date type; false otherwise*

---

## isStringType(int) Method

```
public static boolean isStringType(int columnType)
```

Check a given column type to see if it is a string column type. String types include:

- 

### Parameters

*columnType*

the int representation of a column type

## Returns

*true if the column is a string type; false otherwise*

# FilterWindowService.Filter Class

*public static class FilterWindowService.Filter*

extends Object

The Filter class contains information about filters that are applied to a grid/column pair for which the filter window is being rendered.

---

## Hierarchy

Object  
FilterWindowService.Filter

## Enclosing class

FilterWindowService

---

## Method Summary

public  
int **getOperation()**  
Get the filter operation for a filter.

public  
String **getValue()**  
Get the value of a filter

## Methods from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

## Method Detail

### getOperation() Method

public int getOperation()

Get the filter operation for a filter.

### Returns

*an value represnting a filter operation*

---

## **getValue() Method**

```
public String getValue()
```

Get the value of a filter

### **Returns**

*the String value of a filter*

---

## FilterWindowService.FilterWindowData Class

*public static final class FilterWindowService.FilterWindowData*

extends Object

The FilterWindowData class contains information extracted from the current request and response which is used when rendering a filter window. The information contained here is tightly coupled with all of the infrastructure used to render the filter window and submit new / changed filters to the server.

---

### Hierarchy

```
Object
  FilterWindowService.FilterWindowData
```

### Enclosing class

```
FilterWindowService
```

---

## Method Summary

```
public String
    getClearAllFiltersQueryString()
        Get a query parameters string that can reset all of the filters
        for all columns on the current grid.

public String
    getClearColumnFiltersQueryString()
        Get a query parameters string that can reset all of the filters
        for the current column / grid pair.

public String
    getColumnName()
        Get the name of the column to filter

public int
    getDefaultFilterOperation()
        Given the type of the column, get the default filter operation
        for the column

public String
    getFilterAction()
        Get a Page Flow action which the filter window executes if a
        change is made to the current filter set.

public String
    getFilterKey()
        Get the key used to set a filter on the URL.

public
```



```
FilterWindowService.Filter[] getFilters()
    Get an array of the filters currently applied to a grid for the
    current column

    public String
        getGridName()
        Get the name of the current grid on which a filter may be
        applied

    public int
        getJDBCColumnType()
        Get the JDBC type of the current column.

    public String[][]
        getOptions()
        Get a list of options that are available for display in the filter
        drop downs of the filter window.

    public String
        getPrefix()
        Get any prefix that should be applied to parameters that are
        set on the URL from the filter window.
```

#### Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `wait`, `wait`, `wait`

## Method Detail

### **getClearAllFiltersQueryString() Method**

```
public String getClearAllFiltersQueryString()
```

Get a query parameters string that can reset all of the filters for all columns on the current grid.

#### **Returns**

*the query parameters String that can be used to reset all of the filters on the current grid.*

---

### **getClearColumnFiltersQueryString() Method**

```
public String getClearColumnFiltersQueryString()
```

Get a query parameters string that can reset all of the filters for the current column / grid pair.

#### **Returns**

*the query parameters String that can be used to reset all of the filters on current column / grid pair.*

---

## getColumnName() Method

```
public String getColumnName()
```

Get the name of the column to filter

### Returns

*the String column name to filter*

---

## getDefaultFilterOperation() Method

```
public int getDefaultFilterOperation()
```

Given the type of the column, get the default filter operation for the column

### Returns

*the default filter operation for the currnet column*

---

## getFilterAction() Method

```
public String getFilterAction()
```

Get a Page Flow action which the filter window executes if a change is made to the current filter set.

### Returns

*a String action name*

---

## getFilterKey() Method

```
public String getFilterKey()
```

Get the key used to set a filter on the URL.

### Returns

*the String filter key*

---

## getFilters() Method

```
public FilterWindowService.Filter[] getFilters()
```

Get an array of the filters currently applied to a grid for the current column

**Returns**

*a array*

---

**getGridName() Method**

```
public String getGridName()
```

Get the name of the current grid on which a filter may be applied

**Returns**

*the name of a grid*

---

**getJDBCColumnType() Method**

```
public int getJDBCColumnType()
```

Get the JDBC type of the current column. These values are taken from the Types class.

**Returns**

*an int representing the type of a column*

---

**getOptions() Method**

```
public String[][] getOptions()
```

Get a list of options that are available for display in the filter drop downs of the filter window. Not all operations are available for all column types. The return value is a two dimensional array of Strings as:

```
(readable name of an operation, URL key of an operation)
```

**Returns**

*the list of filter operations.*

---

**getPrefix() Method**

```
public String getPrefix()
```

Get any prefix that should be applied to parameters that are set on the URL from the filter window.

## Returns

*the String prefix*

---

# com.bea.wlw.netui.pageflow Package

## Interface Summary

ActionResult	Encapsulation of the results returned by <code>PageFlowUtils.strutsLookup(ServletContext, ServletRequest, HttpServletResponse, String, String[])</code> .
AutoRegisterActionServlet.ModuleConfigLocator	Interface for specifying alternate locations for auto-registered Struts modules.
HttpRedirector	
PageFlowConstants	Constants related to Page Flows.
RequestParameterHandler	

## Class Summary

AutoRegisterActionServlet	ActionServlet that dynamically registers requested modules.
DynaFormData	Extension of <code>org.apache.struts.validator.DynaValidator</code> .
DynamicSubappActionServlet	ActionServlet that dynamically registers modules based on the subapp name. Generated by the Page Flow compiler.
FlowController	Base class for user-written flow controllers – <code>PageFlowController</code> .
FlowControllerFactory	Factory for creating <code>FlowControllers</code> – <code>userPageFlowControllerFactory</code> .
FormData	Base class for form-beans associated with action methods.
Forward	An object of this type is returned from an action method.
GlobalApp	Base class for user-defined global state and fallback controller.
PageFlowActionServlet	ActionServlet that dynamically registers modules based on the subapp name. Generated by the Page Flow compiler.
PageFlowActionServlet.DefaultModuleConfigLocator	Default <code>ModuleConfigLocator</code> that looks for Struts modules in <code>"/WEB-INF/pageflow-struts-generated/jpf-struts-config.xml"</code> .
PageFlowActionServlet.LegacyModuleConfigLocator	<code>ModuleConfigLocator</code> that looks for legacy Struts modules in <code>"/WEB-INF/jpf-struts-config-<i>&lt;module&gt;</i>".</code>
PageFlowContextListener	Performs various initialization at <code>ServletContext-init</code> time.
PageFlowController	Base class for user-defined state and controller logic.
PageFlowFileFilter	File filter that accepts any <code>PageFlowController</code> .
PageFlowRequestProcessor	The Page Flow extension of the Struts <code>RequestProcessor</code> that handles the Struts action servlet.
PageFlowRequestProcessor.DefaultHttpRedirector	
PageFlowRequestProcessor.ExceptionHandledAction	Used by <code>PageFlowRequestProcessor.ExceptionHandledActionMapping</code> to return an <code>HttpServletResponse, String</code> .
PageFlowRequestProcessor.ExceptionHandledActionMapping	Used by <code>PageFlowRequestProcessor.ExceptionHandledAction</code> to return an <code>HttpServletResponse, String</code> .
PageFlowRequestProcessor.GlobalAppActionMapping	Used by <code>PageFlowRequestProcessor.GlobalAppForwardAction</code> to return an <code>HttpServletResponse, String</code> .
PageFlowRequestProcessor.GlobalAppForwardAction	

	Used by <code>PageFlowRequestProcessor.Global</code> . <code>HttpServletResponse, String</code> ).
<code>PageFlowStack</code>	Stack for keeping track of a series of nested page flows.
<code>PageFlowUtils</code>	Utility methods related to Page Flow.
<code>PreviousActionInfo</code>	Stores information about a recent action execution with on <code>@jpf:forward</code> .
<code>PreviousPageInfo</code>	Stores information about a previously–displayed page.
<code>ProcessPopulate.ExpressionUpdateNode</code>	An inner class that represents the data that will be used

## Exception Summary

<code>ActionNotFoundException</code>	Exception that occurs when the user tries to execute an action that does not exist on the page flow.
<code>EmptyNestingStackException</code>	Exception that occurs when the user invokes an action in a nested page flow that is qualified with a <code>@jpf:forward</code> <code>return-action="action-name-in-calling-pageflow "</code> annotation, but there is no calling page flow.
<code>IllegalOutputFormException</code>	Base type for errors related to output forms on Forwards.
<code>IllegalOutputFormTypeException</code>	Exception that occurs when the first output form for a Forward resolves to a <code>@jpf:forward</code> whose <code>return-form</code> or <code>return-form-type</code> attribute demands a different form type.
<code>IllegalPageInputException</code>	Exception that occurs when a page input has been added to a Forward that resolves to a <code>@jpf:forward</code> marked <code>redirect="true"</code> .
<code>IllegalRedirectOutputFormException</code>	Exception that occurs when an output form has been added to a Forward that resolves to a <code>@jpf:forward</code> marked <code>redirect="true"</code> .
<code>InfiniteReturnToActionException</code>	Exception that occurs when the user invokes an action that is qualified with a <code>@jpf:forward</code> <code>return-to="previousAction"</code> annotation, but the previous action was the same as the current action (an infinite loop).
<code>LoginExpiredException</code>	Exception thrown when <code>NotLoggedInException</code> would be thrown, <i>and</i> when the current <code>HttpServletRequest</code> refers to a session that no longer exists.
<code>NoCurrentPageFlowException</code>	Action that occurs when the user invokes an action that is qualified with either a <code>@jpf:forward</code> <code>return-to="previousAction"</code> or <code>return-to="previousPage"</code> , but there is no current page flow.
<code>NoMatchingActionMethodException</code>	Exception that occurs when the current action method does not accept the type of form passed in the Forward to the action.
<code>NoPreviousActionException</code>	Exception that occurs when the user attempts to execute an action marked with the <code>@jpf:forward</code> <code>return-to="previousAction"</code> annotation, but there is no previously–run action.
<code>NoPreviousPageException</code>	Exception that occurs when the user attempts to execute an action marked with the <code>@jpf:forward</code> <code>return-to="previousPage"</code> annotation, but there is no previous page in the current page flow.

## Page Flow API Reference

### NotLoggedInException

Exception thrown when:

- An action method marked with `@jpf:action login-required="true"` is hit when there is no logged-in user, or
- An action method marked with `@jpf:action roles-allowed="roles"` is hit when there is no logged-in user.

### PageFlowException

Base class for PageFlow-related Exceptions.

### UnfulfilledRolesException

Exception that is thrown when an action method marked with `@jpf:action roles-allowed="roles"` is hit when there is a logged-in user who does not fulfil any of the given roles.

### UnresolvableForwardException

Exception thrown by `Forward` when a forward name does not resolve to a forward defined by `@jpf:forward`.

# ActionNotFoundException Class

*public class **ActionNotFoundException***

extends `PageFlowException`

Exception that occurs when the user tries to execute an action that does not exist on the page flow.

---

## **Hierarchy**

```
Object
  Throwable
    Exception
      RuntimeException
        PageFlowException
          ActionNotFoundException
```

## **All Implemented Interfaces**

```
Serializable
```

---

## **Constructor Summary**

***ActionNotFoundException**(String actionName, FlowController fc, ActionForm form)*

## **Method Summary**

```
protected
ActionForm getForm()
protected
Object[] getMessageArgs()
protected
String[] getMessageParts()
```

### **Methods from `com.bea.wlw.netui.pageflow.PageFlowException`**

```
getActionName, getFlowController, getFlowControllerURI,
getLocalizedMessage, getMessage, printError, sendError
```

### **Methods from `java.lang.Throwable`**



`fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString`

#### Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

## Constructor Detail

### ActionNotFoundException

```
public ActionNotFoundException(String actionName,
                               FlowController fc,
                               ActionForm form)
```

## Method Detail

### getForm() Method

```
protected ActionForm getForm()
```

---

### getMessageArgs() Method

```
protected Object[] getMessageArgs()
```

#### Overrides

```
PageFlowException.getMessageArgs()
```

---

### getMessageParts() Method

```
protected String[] getMessageParts()
```

#### Overrides

```
PageFlowException.getMessageParts()
```

# ActionResult Interface

*public interface **ActionResult***

Encapsulation of the results returned by `PageFlowUtils.strutsLookup(ServletContext, ServletRequest, HttpServletResponse, String, String[])`.

## Method Summary

```
public
    int getStatusCode()
public
    String getStatusMessage()
public
    String getURI()
public
    boolean hadCompileErrors()
public
    boolean isError()
public
    boolean isRedirect()
public
    void printCompileErrors(PrintWriter writer)
```

## Method Detail

### **getStatusCode() Method**

```
public int getStatusCode()
```

---

### **getStatusMessage() Method**

```
public String getStatusMessage()
```

---

### **getURI() Method**

```
public String getURI()
```

---

## **hadCompileErrors() Method**

```
public boolean hadCompileErrors()
```

---

## **isError() Method**

```
public boolean isError()
```

---

## **isRedirect() Method**

```
public boolean isRedirect()
```

---

## **printCompileErrors(PrintWriter) Method**

```
public void printCompileErrors(PrintWriter writer)
```

# AutoRegisterActionServlet Class

*public class **AutoRegisterActionServlet***

extends `ActionServlet`

`ActionServlet` that dynamically registers requested modules based on registered module configuration file locators. The user may specify `AutoRegisterActionServlet.ModuleConfigLocator` classes in `web.xml`, using the "moduleConfigLocators" init-parameter; this allows the user to define patterns for auto-registered Struts modules.

---

## *Hierarchy*

```

Object
  GenericServlet
    HttpServlet
      ActionServlet
        AutoRegisterActionServlet
  
```

## *All Implemented Interfaces*

`Serializable`, `Servlet`, `ServletConfig`

## *Direct Known Subclasses*

`PageFlowActionServlet`

---

## Nested Class Summary

```

public static interface AutoRegisterActionServlet.ModuleConfigLocator
    Interface for specifying alternate locations for auto-registered Struts modules.
  
```

## Field Summary

```

public static String MODULE_CONFIG_LOCATOR_CLASS_ATTR
    String
  
```

## Fields from `org.apache.struts.action.ActionServlet`

`config`, `configDigester`, `convertNull`, `dataSources`, `debug`, `internal`, `internalName`, `log`, `processor`, `registrations`, `servletMapping`, `servletName`

## Constructor Summary

*AutoRegisterActionServlet()*

## Method Summary

```

        public void
            destroy()

        protected
AutoRegisterActionServlet.ModuleConfigLocator[] getDefaultModuleConfigLocators()
            Get the base list of
            ModuleConfigLocators, to specify
            locations for auto-registered Struts
            modules.

        public String
            getInitParameter(String s)

        public Enumeration
            getInitParameterNames()

        protected ModuleConfig
            getModuleConfig(String
            modulePath, ServletRequest request,
            ServletResponse response)
            Get the Struts ModuleConfig for the
            given module path.

        protected
AutoRegisterActionServlet.ModuleConfigLocator[] getModuleConfigLocators()
            Get the current list of registered
            ModuleConfigLocators.

        public String
            getModuleConfPath(String
            modulePath)
            Get the webapp-relative path to the
            Struts module configuration file for a
            given module path, based on
            registered ModuleConfigLocators.

        public void
            init()
            Initialize the servlet.

        public void
            init(ServletConfig config)

        protected void
            process(HttpServletRequest
            request, HttpServletResponse

```

*response)*

This override of the base class `process()` registers a Struts module on the fly if the config file can be found in our standard place (named in our standard way), regardless of whether the module is configured in `web.xml`.

`protected synchronized ModuleConfig`

**`registerModule(String modulePath, String configFilePath)`**

Register a Struts module, initialized by the given configuration file.

### Methods from `org.apache.struts.action.ActionServlet`

`addServletMapping, destroyApplications, destroyConfigDigester, destroyInternal, destroyModules, doGet, doPost, findDataSource, findFormBean, findForward, findMapping, getApplicationConfig, getDebug, getInternal, getModuleConfig, getRequestProcessor, getResources, initApplicationConfig, initApplicationDataSources, initApplicationMessageResources, initApplicationPlugIns, initDataSources, initInternal, initModuleMessageResources, initModulePlugIns, initOther, initServlet, log,`

### Methods from `javax.servlet.http.HttpServlet`

`doDelete, doGet, doHead, doOptions, doPost, doPut, doTrace, getLastModified, service, service`

### Methods from `javax.servlet.GenericServlet`

`destroy, getServletConfig, getServletContext, getServletInfo, getServletName, init, log, log, service`

### Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

### Methods from interface `javax.servlet.Servlet`

`destroy, getServletConfig, getServletInfo, init, service`

### Methods from interface `javax.servlet.ServletConfig`

`getInitParameter, getInitParameterNames, getServletContext, getServletName`

## Field Detail

### MODULE\_CONFIG\_LOCATOR\_CLASS\_ATTR

```
public static String MODULE_CONFIG_LOCATOR_CLASS_ATTR
```

## Constructor Detail

### AutoRegisterActionServlet

```
public AutoRegisterActionServlet()
```

## Method Detail

### destroy() Method

```
public void destroy()
```

#### *Overrides*

```
ActionServlet.destroy()
```

---

### getDefaultModuleConfigLocators() Method

```
protected AutoRegisterActionServlet.ModuleConfigLocator[] getDefaultModuleConfigLocators()
```

Get the base list of ModuleConfigLocators, to specify locations for auto-registered Struts modules. By default, this list is empty; derived classes may override to provide locators, or the user may specify them using the "moduleConfigLocators" init-parameter. When an unrecognized Struts module is requested, each registered ModuleConfigLocator is queried for a possible path to the configuration file for the module. If the configuration file is found, the module is auto-registered against the file.

---

### getInitParameter(String) Method

```
public String getInitParameter(String s)
```

#### *Overrides*

```
GenericServlet.getInitParameter(String)
```

---

## getInitParameterNames() Method

```
public Enumeration getInitParameterNames()
```

### Overrides

```
GenericServlet.getInitParameterNames()
```

---

## getModuleConfig(String, ServletRequest, ServletResponse) Method

```
protected ModuleConfig getModuleConfig(String modulePath,
                                       ServletRequest request,
                                       ServletResponse response)
    throws IOException, ServletException
```

Get the Struts ModuleConfig for the given module path.

### Parameters

*modulePath*

the module path, from the request URI.

*request*

the current ServletRequest

*response*

the current HttpServletResponse

### Returns

*the Struts ModuleConfig that corresponds with modulePath*

### Exceptions

*IOException*

*ServletException*

---

## getModuleConfigLocators() Method

```
protected AutoRegisterActionServlet.ModuleConfigLocator[] getModuleConfigLocators()
```

Get the current list of registered ModuleConfigLocators.

### Returns

*an array of registered ModuleConfigLocators.*

### Related Topics

`AutoRegisterActionServlet.getDefaultModuleConfigLocators()`



## getModuleConfPath(String) Method

```
public String getModuleConfPath(String modulePath)
```

Get the webapp-relative path to the Struts module configuration file for a given module path, based on registered ModuleConfigLocators.

### Parameters

*modulePath*  
the Struts module path.

### Returns

*a String that is the path to the Struts configuration file, relative to the web application root, or null if no appropriate configuration file is found.*

### Related Topics

```
AutoRegisterActionServlet.getDefaultModuleConfigLocators()
```

---

## init() Method

```
public void init()  
    throws ServletException
```

Initialize the servlet.

### Overrides

ActionServlet.init()

### Exceptions

*ServletException*

---

## init(ServletConfig) Method

```
public void init(ServletConfig config)  
    throws ServletException
```

### Overrides

GenericServlet.init(ServletConfig)

### Exceptions

*ServletException*

---

## process(HttpServletRequest, HttpServletResponse) Method

```
protected void process(HttpServletRequest request,
                      HttpServletResponse response)
    throws IOException, ServletException
```

This override of the base class process() registers a Struts module on the fly if the config file can be found in our standard place (named in our standard way), regardless of whether the module is configured in web.xml.

### Overrides

ActionServlet.process(HttpServletRequest, HttpServletResponse)

### Exceptions

*IOException*

*ServletException*

---

## registerModule(String, String) Method

```
protected synchronized ModuleConfig registerModule(String modulePath,
                                                    String configFilePath)
    throws ServletException
```

Register a Struts module, initialized by the given configuration file.

### Parameters

*modulePath*

the module path, starting at the webapp root, e.g., "/info/help".

*configFilePath*

the path, starting at the webapp root, to the module configuration file (e.g., "/WEB-INF/my-generated-struts-config-info-help.xml").

### Returns

*the Struts ModuleConfig that was initialized.*

### Exceptions

*ServletException*

---

# AutoRegisterActionServlet.ModuleConfigLocator Interface

*public static interface **AutoRegisterActionServlet.ModuleConfigLocator***

Interface for specifying alternate locations for auto-registered Struts modules.

## Related Topics

`AutoRegisterActionServlet.ModuleConfigLocator.getDefaultModuleConfigLocators()`

---

## All Known Implementing Classes

`PageFlowActionServlet.DefaultModuleConfigLocator`

## Enclosing class

`AutoRegisterActionServlet`

---

## Method Summary

`public`  
`String` **`getModuleConfigPath`**(*String* `moduleName`)  
Get the webapp-relative path to a Struts module config file, based on the module name.

## Method Detail

### **`getModuleConfigPath(String)` Method**

`public String getModuleConfigPath(String moduleName)`

Get the webapp-relative path to a Struts module config file, based on the module name.

#### Parameters

*moduleName*  
the name of the Struts module, e.g., "someModule" or "some/other/module".

#### Returns

*the webapp-relative path the the Struts module config file.*

# DynaFormData Class

*public class DynaFormData*

extends DynaValidatorForm  
implements Map

Extension of org.apache.struts.validator.DynaValidatorForm that implements Map. Although org.apache.struts.validator.DynaValidatorForm cannot be used with <netui> tags, com.bea.wlw.netui.pageflow.DynaFormData may be used with <netui> tags.

---

## *Hierarchy*

Object  
  ActionForm  
    DynaActionForm  
      DynaValidatorForm  
        DynaFormData

## *All Implemented Interfaces*

DynaBean, Map, Serializable

---

## Field Summary

**Fields from org.apache.struts.validator.DynaValidatorForm**

page, validatorResults

**Fields from org.apache.struts.action.DynaActionForm**

dynaClass, dynaValues

**Fields from org.apache.struts.action.ActionForm**

multipartRequestHandler, servlet

## Constructor Summary

*DynaFormData()*

## Method Summary

```

    public
        void clear()
    public
boolean containsKey(Object key)
    public
boolean containsValue(Object value)
    public
        Set entrySet()
    public
        Object get(Object name)
    public
boolean isEmpty()
    public
        Set keySet()
    public
        Object put(Object key, Object value)
    public
        void putAll(Map map)
    public
        Object remove(Object key)
    public
        int size()
    public
Collection values()

```

### Methods from `org.apache.struts.validator.DynaValidatorForm`

```

getPage, getResultValueMap, getValidatorResults, log, log, reset,
setPage, setValidatorResults, validate

```

### Methods from `org.apache.struts.action.DynaActionForm`

```

contains, get, get, get, getDynaClass, getDynaProperty, getMap,
initialize, isDynaAssignable, remove, reset, reset, set, set, set,
toString

```

### Methods from `org.apache.struts.action.ActionForm`

```

getMultipartRequestHandler, getServlet, getServletWrapper, reset, reset,
setMultipartRequestHandler, setServlet, validate, validate

```

### Methods from class `java.lang.Object`

```

clone, equals, finalize, getClass, hashCode, notify, notifyAll,
toString, wait, wait, wait

```

### Methods from interface `org.apache.commons.beanutils.DynaBean`

`contains, get, get, get, getDynaClass, remove, set, set, set`

### Methods from interface `java.util.Map`

`clear, containsKey, containsValue, entrySet, equals, get, hashCode, isEmpty, keySet, put, putAll, remove, size, values`

## Constructor Detail

### DynaFormData

```
public DynaFormData()
```

## Method Detail

### clear() Method

```
public void clear()
```

---

### containsKey(Object) Method

```
public boolean containsKey(Object key)
```

---

### containsValue(Object) Method

```
public boolean containsValue(Object value)
```

---

### entrySet() Method

```
public Set entrySet()
```

---

### get(Object) Method

```
public Object get(Object name)
```

---

### isEmpty() Method

```
public boolean isEmpty()
```

---

## keySet() Method

```
public Set keySet()
```

---

## put(Object, Object) Method

```
public Object put(Object key,  
                  Object value)
```

---

## putAll(Map) Method

```
public void putAll(Map map)
```

---

## remove(Object) Method

```
public Object remove(Object key)
```

---

## size() Method

```
public int size()
```

---

## values() Method

```
public Collection values()
```

**DEPRECATED** Use `com.bea.wlw.netui.pageflow.PageFlowActionServlet` instead.

# DynamicSubappActionServlet Class

*public class **DynamicSubappActionServlet***

extends com.bea.wlw.netui.pageflow.PageFlowActionServlet

ActionServlet that dynamically registers modules based on naming/location conventions for Struts configuration files that are generated by the Page Flow compiler. These files are located in /WEB-INF/.pageflow-struts-generated, and are named jpf-struts-config-*module-name*.xml. The user may specify additional classes in web.xml, using the "moduleConfigLocators" init-parameter; this allows the user to define other patterns for auto-registered Struts modules.

---

## **Hierarchy**

```
Object
  GenericServlet
    HttpServlet
      ActionServlet
        com.bea.wlw.netui.pageflow.AutoRegisterActionServlet
        com.bea.wlw.netui.pageflow.PageFlowActionServlet
        DynamicSubappActionServlet
```

## **All Implemented Interfaces**

Serializable, Servlet, ServletConfig

---

## **Nested Class Summary**

### **Nested classes from**

**com.bea.wlw.netui.pageflow.com.bea.wlw.netui.pageflow.PageFlowActionServlet**

PageFlowActionServlet.DefaultModuleConfigLocator,  
PageFlowActionServlet.LegacyModuleConfigLocator

### **Nested classes from**

**com.bea.wlw.netui.pageflow.com.bea.wlw.netui.pageflow.AutoRegisterActionServlet**

AutoRegisterActionServlet.ModuleConfigLocator



## Field Summary

### Fields from

**com.bea.wlw.netui.pageflow.com.bea.wlw.netui.pageflow.AutoRegisterActionServlet**  
 MODULE\_CONFIG\_LOCATOR\_CLASS\_ATTR

### Fields from org.apache.struts.action.ActionServlet

config, configDigester, convertNull, dataSources, debug, internal, internalName, log, processor, registrations, servletMapping, servletName

## Constructor Summary

*DynamicSubappActionServlet()*

## Method Summary

### Methods from

**com.bea.wlw.netui.pageflow.com.bea.wlw.netui.pageflow.PageFlowActionServlet**  
 getDefaultModuleConfigLocators, getModuleConfPath, init, process,

### Methods from

**com.bea.wlw.netui.pageflow.com.bea.wlw.netui.pageflow.AutoRegisterActionServlet**  
 destroy, getDefaultModuleConfigLocators, getInitParameter, getInitParameterNames, getModuleConfig, getModuleConfigLocators, getModuleConfPath, init, init, process, registerModule,

### Methods from org.apache.struts.action.ActionServlet

addServletMapping, destroy, destroyApplications, destroyConfigDigester, destroyDataSources, destroyInternal, destroyModules, doGet, doPost, findDataSource, findFormBean, findForward, findMapping, getApplicationConfig, getDebug, getInternal, getModuleConfig, getRequestProcessor, getResources, init, initApplicationConfig, initApplicationDataSources, initApplicationMessageResources, initApplicationPlugIns, initConfigDigester, initDataSources, initInternal, initModuleConfig, initModuleDataSources, initModuleMessageResources, initModulePlugIns, initOther, initServlet, log, process

**Methods from `javax.servlet.http.HttpServlet`**

`doDelete`, `doGet`, `doHead`, `doOptions`, `doPost`, `doPut`, `doTrace`,  
`getLastModified`, `service`, `service`

**Methods from `javax.servlet.GenericServlet`**

`destroy`, `getInitParameter`, `getInitParameterNames`, `getServletConfig`,  
`getServletContext`, `getServletInfo`, `getServletName`, `init`, `init`, `log`, `log`,  
`service`

**Methods from class `java.lang.Object`**

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`,  
`toString`, `wait`, `wait`, `wait`

**Methods from interface `javax.servlet.Servlet`**

`destroy`, `getServletConfig`, `getServletInfo`, `init`, `service`

**Methods from interface `javax.servlet.ServletConfig`**

`getInitParameter`, `getInitParameterNames`, `getServletContext`,  
`getServletName`

## Constructor Detail

### DynamicSubappActionServlet

```
public DynamicSubappActionServlet()
```

# EmptyNestingStackException Class

*public class EmptyNestingStackException*

extends PageFlowException

Exception that occurs when the user invokes an action in a nested page flow that is qualified with a `@jpf:forward return-action="action-name-in-calling-pageflow "` annotation, but there is no calling page flow. This can happen in iterative development mode when you have modified files and caused the web application to be redeployed, or when the session expires.

---

## Hierarchy

```
Object
  Throwable
    Exception
      RuntimeException
        PageFlowException
          EmptyNestingStackException
```

## All Implemented Interfaces

```
Serializable
```

---

## Constructor Summary

*EmptyNestingStackException(String actionName, FlowController fc)*

## Method Summary

```
protected
  Object[] getMessageArgs()
protected
  String[] getMessageParts()
```

### Methods from com.bea.wlw.netui.pageflow.PageFlowException

```
getActionName, getFlowController, getFlowControllerURI,
getLocalizedMessage, getMessage, printError, sendError
```

### Methods from java.lang.Throwable

`fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString`

### Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

## Constructor Detail

### EmptyNestingStackException

```
public EmptyNestingStackException(String actionName,  
                                FlowController fc)
```

## Method Detail

### getMessageArgs() Method

```
protected Object[] getMessageArgs()
```

#### *Overrides*

```
PageFlowException.getMessageArgs()
```

---

### getMessageParts() Method

```
protected String[] getMessageParts()
```

#### *Overrides*

```
PageFlowException.getMessageParts()
```

# FlowController Class

*public abstract class FlowController*

extends Action  
implements ActionResolver, HttpSessionBindingListener, PageFlowConstants,  
Serializable

Base class for user-written flow controllers – PageFlowControllers and Global.app.

---

## **Hierarchy**

Object  
Action  
FlowController

## **All Implemented Interfaces**

EventListener, HttpSessionBindingListener, PageFlowConstants,  
Serializable

## **Direct Known Subclasses**

GlobalApp, PageFlowController

---

## Field Summary

### **Fields from org.apache.struts.action.Action**

ACTION\_SERVLET\_KEY, APPLICATION\_KEY, DATA\_SOURCE\_KEY, defaultLocale,  
ERROR\_KEY, EXCEPTION\_KEY, FORM\_BEANS\_KEY, FORWARDS\_KEY, LOCALE\_KEY,  
MAPPING\_KEY, MAPPINGS\_KEY, MESSAGE\_KEY, MESSAGES\_KEY, MULTIPART\_KEY,  
PLUG\_INS\_KEY, REQUEST\_PROCESSOR\_KEY, servlet, SERVLET\_KEY,  
TRANSACTION\_TOKEN\_KEY

### **Fields from interface com.bea.wlw.netui.pageflow .PageFlowConstants**

ACTION\_EXTENSION, ACTION\_EXTENSION\_LEN, ACTION\_OVERRIDE,  
BEGIN\_ACTION\_NAME, BEGIN\_ACTION\_PATH, DEFAULT\_FORWARD\_OVERFLOW\_COUNT,  
DEFAULT\_NESTING\_OVERFLOW\_COUNT, FORWARD\_OVERFLOW\_COUNT\_PARAM,  
GLOBALAPP\_CLASSNAME, GLOBALAPP\_MEMBER\_NAME,  
GLOBALAPP\_MODULE\_CONTEXT\_PATH, GLOBALAPP\_SOURCEFILE\_NAME,  
INTERNAL\_VAR\_PREFIX, ISNESTABLE\_VAR\_NAME, JPF\_EXTENSION,  
JPF\_MODULE\_CONFIG\_EXTENSION, JPF\_MODULE\_CONFIG\_GEN\_DIR,  
JPF\_MODULE\_CONFIG\_PREFIX, NESTING\_OVERFLOW\_COUNT\_PARAM,

PROCESS\_ROLES\_FAILED\_ATTR, WEBINF\_DIR

## Constructor Summary

### *FlowController()*

Default constructor.

## Method Summary

protected  
synchronized void **afterAction()**  
Callback that occurs after any user action method is invoked.

protected boolean  
**alwaysTrackPreviousAction()**  
When the current page flow does not use a @jpf:forward with a return-to="previousAction" attribute, the following methods always return null by default.

protected boolean  
**alwaysTrackPreviousPage()**  
When the current page flow does not use a @jpf:forward with either a return-to="currentPage" or a return-to="previousPage" attribute, the following methods always return null by default.

protected  
synchronized void **beforeAction()**  
Callback that occurs before any user action method is invoked.

public abstract  
void **ensureFailover**(*HttpServletRequest request*)  
Ensures that any changes to this FlowController will be replicated in a cluster (for failover), even if the replication scheme uses a change-detection algorithm that relies on HttpSession.setAttribute to be aware of changes.

public  
ActionForward **execute**(*ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response*)  
Perform decision logic to determine the next URI to be displayed.

protected Method  
**getActionMethod**(*String methodName, Class argType*)  
Get an action handler method of the given name/signature .

public String[]  
**getActions()**  
Get a list of the names of actions handled by methods in this

## Page Flow API Reference

PageFlowController.

```
protected String
    getCurrentActionName()
        Get the name of the current action being executed.

public abstract
    String getDisplayName()

protected final
    ActionMapping getMapping()
        Get the current Struts ActionMapping, which is information from the
        Struts–XML <action> tag that corresponds to the current page flow
        action being executed.

public ModuleConfig
    getModuleConfig(ServletContext context, HttpServletRequest
        request)
        Gets the Struts module configuration associated with this controller.

public abstract
    String getModulePath()
        Get the Struts module path for this controller.

protected final
    HttpServletRequest getRequest()
        Get the current HttpServletRequest.

protected final
    HttpServletResponse getResponse()
        Get the current HttpServletResponse.

    public
        ActionServlet getServlet()
            Get the current Struts ActionServlet.

protected final
    HttpSession getSession()
        Get the current user session.

    protected
        synchronized handleException(Throwable ex, ActionMapping mapping, String
            ActionForward actionName, ActionForm form, HttpServletRequest request,
            HttpServletResponse response)
            Handle the given exception – invoke user code if appropriate and return
            a destination URI.

    protected
        ActionForward invokeActionMethod(Method method, ActionForm arg)
            Invoke the given action handler method, passing it an argument if
            appropriate.

public boolean
    isAction(String name)
        Tell whether a given String is the name of an action handled by a
        method in this PageFlowController.

protected boolean
    isNestable()
        Tell whether this PageFlowController can be "nested", i.e., if it can be
        invoked from another page flow with the intention of returning to the
```

## Page Flow API Reference

original one.

```
public boolean
    isPageFlow()
        Tell whether this is a PageFlowController.

public void
    login(String username, String password)
        Log in the user, using "weak" username/password authentication.

public void
    logout(boolean invalidateSessions)
        Log out the current user.

protected void
    onCreate()
        Callback that is invoked when this controller instance is created.

protected void
    onDestroy()
        Callback that is invoked when this controller instance is "destroyed",
        i.e., removed from the user session.

protected void
    onDestroy(HttpSession session)
        Callback that is invoked when this controller instance is "destroyed",
        i.e., removed from the user session.

public String
    resolveAction(String actionName, ActionForm form)
        Call an action and return the result URI.

public static
    Locale retrieveUserLocale(HttpServletRequest request, String locale)

protected void
    sendError(String errText, HttpServletResponse response)
        Send a Page Flow error to the browser.

public void
    valueBound(HttpSessionBindingEvent event)
        Callback when this FlowController is added to the user session.

public void
    valueUnbound(HttpSessionBindingEvent event)
        Callback when this FlowController is removed from the user session.
```

### Methods from `org.apache.struts.action.Action`

`execute`, `generateToken`, `getDataSource`, `getDataSource`, `getLocale`,  
`getResources`, `getResources`, `getResources`, `isCancelled`, `isTokenValid`,  
`isTokenValid`, `perform`, `perform`, `resetToken`, `saveErrors`, `saveMessages`,  
`saveToken`, `setLocale`, `toHex`

### Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`,  
`toString`, `wait`, `wait`, `wait`



Methods from interface `javax.servlet.http.HttpSessionBindingListener`

`valueBound`, `valueUnbound`

## Constructor Detail

### FlowController

```
public FlowController()
```

Default constructor.

## Method Detail

### afterAction() Method

```
protected synchronized void afterAction()
    throws Exception
```

Callback that occurs after any user action method is invoked. `FlowController.getRequest()`, `FlowController.getResponse()`, `FlowController.getSession()`, and `FlowController.getMapping()` may all be used during this method. The action that was run can be discovered by calling `ActionMapping.getPath()` on the value returned from `FlowController.getMapping()`.

### Exceptions

*Exception*

---

### alwaysTrackPreviousAction() Method

```
protected boolean alwaysTrackPreviousAction()
```

When the current page flow does not use a `@jpf:forward` with a `return-to="previousAction"` attribute, the following methods always return `null` by default.

- `getPreviousActionInfo`
- `getPreviousActionURI`
- `getPreviousForm`

Override `alwaysTrackPreviousAction` (which always returns `false`) to enable these methods in all cases.

**Returns**

*true if the previous action should always be tracked, regardless of whether `return-to="previousAction"` is used.*

**Related Topics**

```
PageFlowController.getPreviousActionInfo()
PageFlowController.getPreviousActionURI()
PageFlowController.getPreviousForm()
```

---

**alwaysTrackPreviousPage() Method**

```
protected boolean alwaysTrackPreviousPage()
```

When the current page flow does not use a `@jpf:forward` with either a `return-to="currentPage"` or a `return-to="previousPage"` attribute, the following methods always return null by default.

- `getCurrentPageInfo`
- `getPreviousPageInfo`
- `getCurrentForwardPath`
- `getPreviousForwardPath`

Override `alwaysTrackPreviousPage` (which always returns false) to enable these methods in all cases.

**Returns**

*true if the previous page should always be tracked, regardless of whether `return-to="currentPage"` or `return-to="previousPage"` is used.*

**Related Topics**

```
PageFlowController.getCurrentPageInfo()
PageFlowController.getPreviousPageInfo()
PageFlowController.getCurrentForwardPath()
PageFlowController.getPreviousForwardPath()
```

---

**beforeAction() Method**

```
protected synchronized void beforeAction()
    throws Exception
```

Callback that occurs before any user action method is invoked. `FlowController.getRequest()`, `FlowController.getResponse()`, `FlowController.getSession()`, and `FlowController.getMapping()` may all be used during this method. The action to be run can be discovered by calling `ActionMapping.getPath()` on the value returned from `FlowController.getMapping()`.

## Exceptions

*Exception*

---

### ensureFailover(HttpServletRequest) Method

```
public abstract void ensureFailover(HttpServletRequest request)
```

Ensures that any changes to this FlowController will be replicated in a cluster (for failover), even if the replication scheme uses a change–detection algorithm that relies on HttpSession.setAttribute to be aware of changes.

#### Parameters

*request*  
the current HttpServletRequest

---

### execute(ActionMapping, ActionForm, HttpServletRequest, HttpServletResponse) Method

```
public ActionForward execute(ActionMapping mapping,
                             ActionForm form,
                             HttpServletRequest request,
                             HttpServletResponse response)
    throws Exception
```

Perform decision logic to determine the next URI to be displayed.

#### Overrides

Action.execute(ActionMapping, ActionForm, HttpServletRequest, HttpServletResponse)

#### Parameters

*mapping*  
the Struts ActionMapping for the current action being processed.

*form*  
the form–bean (if any) associated with the Struts action being processed. May be null.

*request*  
the current HttpServletRequest.

*response*  
the current HttpServletResponse.

#### Returns

*a Struts ActionForward object that specifies the next URI to be displayed.*

## Exceptions

*Exception*

if an Exception was thrown during user action-handling code.

---

## getActionMethod(String, Class) Method

```
protected Method getActionMethod(String methodName,
                                   Class argType)
```

Get an action handler method of the given name/signature .

### Parameters

*methodName*

the name of the action handler method to query.

*argType*

the type of the argument to the action handler method; if `null`, the method takes no arguments.

### Returns

*the desired Method, or null if it doesn't exist.*

---

## getActions() Method

```
public String[] getActions()
```

Get a list of the names of actions handled by methods in this PageFlowController.

### Returns

*a String array containing the names of actions handled by methods in this PageFlowController.*

---

## getCurrentActionName() Method

```
protected String getCurrentActionName()
```

Get the name of the current action being executed. This call is only valid during `FlowController.execute(ActionMapping, ActionForm, HttpServletRequest, HttpServletResponse)` (where any user action method is invoked), and during the lifecycle methods `FlowController.beforeAction()` and `FlowController.afterAction()`.

### Returns

*the name of the current action being executed.*

---

## getDisplayName() Method

```
public abstract String getDisplayName()
```

---

## getMapping() Method

```
protected final ActionMapping getMapping()
```

Get the current Struts ActionMapping, which is information from the Struts–XML <action> tag that corresponds to the current page flow action being executed. This call is only valid during `FlowController.execute(ActionMapping, ActionForm, HttpServletRequest, HttpServletResponse)` (where any user action method is invoked), and during the lifecycle methods `FlowController.beforeAction()` and `FlowController.afterAction()`.

### Returns

*the current Struts ActionMapping.*

---

## getModuleConfig(ServletContext, HttpServletRequest) Method

```
public ModuleConfig getModuleConfig(ServletContext context,  
                                   HttpServletRequest request)
```

Gets the Struts module configuration associated with this controller.

### Parameters

*context*  
the current ServletContext.

### Returns

*the Struts ModuleConfig for this controller.*

---

## getModulePath() Method

```
public abstract String getModulePath()
```

Get the Struts module path for this controller.

### Returns

*a String that is the Struts module path for this controller – the parent directory for a PageFlowController and "/-global" for Global.app.*

---

## getRequest() Method

```
protected final HttpServletRequest getRequest()
```

Get the current HttpServletRequest. This call is only valid during `FlowController.execute(ActionMapping, ActionForm, HttpServletRequest, HttpServletResponse)` (where any user action method is invoked), and during the lifecycle methods `FlowController.onCreate()`, `FlowController.beforeAction()`, `FlowController.afterAction()`.

### Returns

*the current HttpServletRequest.*

---

## getResponse() Method

```
protected final HttpServletResponse getResponse()
```

Get the current HttpServletResponse. This call is only valid during `FlowController.execute(ActionMapping, ActionForm, HttpServletRequest, HttpServletResponse)` (where any user action method is invoked), and during the lifecycle methods `FlowController.onCreate()`, `FlowController.beforeAction()`, `FlowController.afterAction()`.

### Returns

*the current HttpServletResponse.*

---

## getServlet() Method

```
public ActionServlet getServlet()
```

Get the current Struts ActionServlet.

### Overrides

`Action.getServlet()`

### Returns

*the ActionServlet.*

---

## getSession() Method

```
protected final HttpSession getSession()
```

Get the current user session. This call is only valid during `FlowController.execute(ActionMapping, ActionForm, HttpServletRequest,`

`HttpServletResponse`) (where any user action method is invoked), and during the lifecycle methods `FlowController.onCreate()`, `FlowController.onDestroy()`, `FlowController.beforeAction()`, `FlowController.afterAction()`.

### Returns

*the `HttpSession` for the current user session.*

---

## **handleException(Throwable, ActionMapping, String, ActionForm, HttpServletRequest, HttpServletResponse) Method**

```
protected synchronized ActionForward handleException(Throwable ex,
                                                    ActionMapping mapping,
                                                    String actionName,
                                                    ActionForm form,
                                                    HttpServletRequest request,
                                                    HttpServletResponse response)
    throws Exception
```

Handle the given exception – invoke user code if appropriate and return a destination URI.

### Parameters

*ex*

the Exception to handle.

*mapping*

the Struts action mapping for current Struts action being processed.

*actionName*

the name of the Struts action being processed.

*form*

the form-bean (if any) associated with the Struts action being processed. May be null.

*request*

the current `HttpServletRequest`.

*response*

the current `HttpServletResponse`.

### Returns

*a Struts `ActionForward` object that specifies the URI that should be displayed.*

### Exceptions

*Exception*

if another Exception is thrown during handling of *ex*.

---

**invokeActionMethod(Method, ActionForm) Method**

```
protected ActionForward invokeActionMethod(Method method,
                                           ActionForm arg)
    throws Exception
```

Invoke the given action handler method, passing it an argument if appropriate.

**Parameters**

*method*

the action handler method to invoke.

*arg*

the form-bean to pass; may be null.

**Returns**

*the ActionForward returned by the action handler method.*

**Exceptions**

*Exception*

if an Exception was raised in user code.

---

**isAction(String) Method**

```
public boolean isAction(String name)
```

Tell whether a given String is the name of an action handled by a method in this PageFlowController.

**Parameters**

*name*

the action-name to query.

**Returns**

*true if name is the name of an action handled by a method in this PageFlowController.*

---

**isNestable() Method**

```
protected boolean isNestable()
```

Tell whether this PageFlowController can be "nested", i.e., if it can be invoked from another page flow with the intention of returning to the original one. Page flows are declared to be nested using the `@jpf:controller nested="true"` annotation.



**Returns**

*true if this PageFlowController can be nested.*

---

**isPageFlow() Method**

```
public boolean isPageFlow()
```

Tell whether this is a PageFlowController.

**Returns**

*true if this is a PageFlowController.*

---

**login(String, String) Method**

```
public void login(String username,
                  String password)
    throws LoginException
```

Log in the user, using "weak" username/password authentication.

**Parameters**

*username*  
the user's login name

*password*  
the user's password

**Exceptions**

*LoginException*  
if the authentication failed

---

**logout(boolean) Method**

```
public void logout(boolean invalidateSessions)
```

Log out the current user. This causes the user to be removed from Servlet Session (on all single–signon webapps) as well as from the thread.

**Parameters**

*invalidateSessions*  
if true, the session is invalidated (on all single–signon webapps); otherwise the session and its data are left intact (except for authentication information used internally by the server). To invalidate the session in only the current webapp, set this parameter to `false` and call

---

```
FlowController.getSession().invalidate();
```

---

## onCreate() Method

```
protected void onCreate()
    throws Exception
```

Callback that is invoked when this controller instance is created. `FlowController.getRequest()`, `FlowController.getResponse()`, `FlowController.getSession()` may all be used during this method.

### Exceptions

*Exception*

---

## onDestroy() Method

### **DEPRECATED**

`com.bea.wlw.netui.pageflow.FlowController.onDestroy(javax.servlet.http.HttpSession)` should be used instead.

```
protected void onDestroy()
```

Callback that is invoked when this controller instance is "destroyed", i.e., removed from the user session. `FlowController.getRequest()`, `FlowController.getResponse()`, and `FlowController.getMapping()` may *not* be used during this method, since it may be called due to session termination outside of a request. `FlowController.getSession()` also may not be used, but the session is passed as an argument to `FlowController.onDestroy(HttpSession)`, which should be used in place of this method.

---

## onDestroy(HttpSession) Method

```
protected void onDestroy(HttpSession session)
```

Callback that is invoked when this controller instance is "destroyed", i.e., removed from the user session. `FlowController.getRequest()`, `FlowController.getResponse()`, and `FlowController.getMapping()` may *not* be used during this method, since it may be called due to session termination outside of a request. `FlowController.getSession()` also may not be used, but the session is passed as an argument.

Note that this method is *not synchronized*.

---

## resolveAction(String, ActionForm) Method

```
public String resolveAction(String actionName,
                           ActionForm form)
    throws Exception
```

Call an action and return the result URI.

### Parameters

*actionName*

the name of the action to run.

*form*

the form bean instance to pass to the action, or `null` if none should be passed.

### Returns

*the result webapp-relative URI, as a String.*

### Exceptions

*Exception*

if the action method throws an Exception.

---

## retrieveUserLocale(HttpServletRequest, String) Method

```
public static Locale retrieveUserLocale(HttpServletRequest request,
                                       String locale)
```

---

## sendError(String, HttpServletResponse) Method

```
protected void sendError(String errText,
                        HttpServletResponse response)
    throws IOException
```

Send a Page Flow error to the browser.

### Parameters

*errText*

the error message to display.

*response*

the current HttpServletResponse.

### Exceptions

*IOException*

---

## valueBound(HttpSessionBindingEvent) Method

```
public void valueBound(HttpSessionBindingEvent event)
```

Callback when this FlowController is added to the user session.

---

## **valueUnbound(HttpSessionBindingEvent) Method**

```
public void valueUnbound(HttpSessionBindingEvent event)
```

Callback when this FlowController is removed from the user session. Causes `FlowController.onDestroy()` to be called.

# FlowControllerFactory Class

*public class FlowControllerFactory*

extends Object

Factory for creating FlowControllers – user PageFlowControllers and Global.app.

---

## Hierarchy

Object  
FlowControllerFactory

---

## Constructor Summary

*FlowControllerFactory()*

## Method Summary

```
public static
    GlobalApp getGlobalApp(HttpServletRequest request,
                           HttpServletResponse response, ServletContext servletContext)
    Get or create the current user session's GlobalApp instance (from the
    Global.app file).

public static
PageFlowController getPageFlow(String jpfClassName, HttpServletRequest request,
                                 HttpServletResponse response, ServletContext servletContext)
    Get the PageFlowController for the given type.

public static
PageFlowController getPageFlowForRequest(HttpServletRequest request,
                                           HttpServletResponse response, ServletContext context)
    Get the PageFlowController instance that should be associated
    with the given request, based on the path of the request URI.

public static
FlowController getPageFlowForURI(HttpServletRequest request,
                                   HttpServletResponse response, String uri, ServletContext
                                   context)
    Get the PageFlowController instance that should be associated
    with the given URI.
```

Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

## Constructor Detail

### FlowControllerFactory

```
public FlowControllerFactory()
```

## Method Detail

### getGlobalApp(HttpServletRequest, HttpServletResponse, ServletContext) Method

```
public static GlobalApp getGlobalApp(HttpServletRequest request,
                                     HttpServletResponse response,
                                     ServletContext servletContext)
```

Get or create the current user session's GlobalApp instance (from the Global.app file).

#### Parameters

*request*  
the current HttpServletRequest.

#### Returns

*the current session's GlobalApp instance, or a new one (based on Global.app) if none is found. If Global.app does not exist in the current webapp, null is returned.*

---

### getPageFlow(String, HttpServletRequest, HttpServletResponse, ServletContext) Method

```
public static PageFlowController getPageFlow(String jpfClassName,
                                             HttpServletRequest request,
                                             HttpServletResponse response,
                                             ServletContext servletContext)
```

Get the PageFlowController for the given type. The PageFlowController stack (for nesting) will be cleared or pushed, and the new instance will be stored as the current PageFlowController.

#### Parameters

*jpfClassName*  
the name of the desired PageFlowController class.

*request*

the current `HttpServletRequest`.

## Returns

*the current user session's `PageFlowController` if it is of the requested type, or a newly-created one, or `null` if none could be found or created.*

---

## **getPageFlowForRequest(HttpServletRequest, HttpServletResponse, ServletContext) Method**

```
public static PageFlowController getPageFlowForRequest(HttpServletRequest request,
                                                         HttpServletResponse response,
                                                         ServletContext context)
```

Get the `PageFlowController` instance that should be associated with the given request, based on the path of the request URI. If it doesn't exist, create it. The `PageFlowController` stack (for nesting) will be cleared or pushed, and the new instance will be stored as the current `PageFlowController`.

## Parameters

*request*

the current `HttpServletRequest`.

*response*

the current `ServletContext`.

## Returns

*the `PageFlowController` for the request, or `null` if none was found and none could be created.*

---

## **getPageFlowForURI(HttpServletRequest, HttpServletResponse, String, ServletContext) Method**

```
public static FlowController getPageFlowForURI(HttpServletRequest request,
                                                 HttpServletResponse response,
                                                 String uri,
                                                 ServletContext context)
```

Get the `PageFlowController` instance that should be associated with the given URI. If it doesn't exist, create it. The `PageFlowController` stack (for nesting) will be cleared or pushed, and the new instance will be stored as the current `PageFlowController`.

## Parameters

*request*

the current `HttpServletRequest`.

*response*

## Page Flow API Reference

the current `HttpServletResponse`.

*uri*

the webapp-relative URI for the page flow.

*context*

the current `ServletContext`.

### **Returns**

*the `PageFlowController` for the request, or `null` if none was found and none could be created.*



# FormData Class

*public class FormData*

extends ValidatorForm

Base class for form-beans associated with action methods in PageFlowControllers.

---

## *Hierarchy*

Object  
  ActionForm  
    ValidatorForm  
      FormData

## *All Implemented Interfaces*

Serializable

## *Direct Known Subclasses*

RowSetForm

---

## Field Summary

**Fields from org.apache.struts.validator.ValidatorForm**

page, validatorResults

**Fields from org.apache.struts.action.ActionForm**

multipartRequestHandler, servlet

## Constructor Summary

*FormData()*

Default  
constructor.

## Method Summary

### Methods from `org.apache.struts.validator.ValidatorForm`

`getPage`, `getResultValueMap`, `getValidatorResults`, `log`, `log`, `reset`,  
`setPage`, `setValidatorResults`, `validate`

### Methods from `org.apache.struts.action.ActionForm`

`getMultipartRequestHandler`, `getServlet`, `getServletWrapper`, `reset`, `reset`,  
`setMultipartRequestHandler`, `setServlet`, `validate`, `validate`

### Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`,  
`toString`, `wait`, `wait`, `wait`

## Constructor Detail

### FormData

```
public FormData()
```

Default constructor.

# Forward Class

*public class Forward*

extends ActionForward

An object of this type is returned from an action methods in a PageFlowController to determine the next URI to be displayed. It is constructed on the name of a forward defined by the @jpf:forward tag, and resolves to the URI specified in that forward.

---

## Hierarchy

```
Object
  ForwardConfig
    ActionForward
      Forward
```

## All Implemented Interfaces

```
Serializable
```

---

## Field Summary

```
public
static RETURN_TO_ACTION
final int      int

public
static RETURN_TO_CURRENT_PAGE
final int      int

public
static RETURN_TO_NONE
final int      int

public
static RETURN_TO_PAGE
final int      int

public
static RETURN_TO_PREVIOUS_ACTION
final int      int

public
static RETURN_TO_PREVIOUS_PAGE
final int      int
```

## Fields from org.apache.struts.config.ForwardConfig

configured, contextRelative, name, path, redirect

## Constructor Summary

***Forward(String forwardName, String pageInputName, Object pageInputValue)***

Constructor which accepts the name of a forward defined by the @jpf:forward tag.

***Forward(String forwardName, ActionForm outputForm)***

Constructor which accepts the name of a forward defined by the @jpf:forward tag.

***Forward(String forwardName)***

Constructor which accepts the name of a forward defined by the @jpf:forward tag.

***Forward(URI path, boolean redirect)***

Constructs a Forward that returns the given URI for `Forward.getPath()`.

***Forward(URI path)***

Constructs a Forward that returns the given URI for `Forward.getPath()`.

***Forward(URL path)***

Constructs a Forward that returns the given URL for `Forward.getPath()`.

## Method Summary

```
public final
    void addOutputForm(ActionForm form)
        Add a form bean that will be made available in the request (or user session, as
        appropriate) if this Forward is returned by an action method in a
        PageFlowController.

public void
    addPageInput(String name, Object value)
        Adds a page input that will be made available in the request, through
        PageFlowUtils.getPageInput(String, ServletRequest)..

public void
    addQueryParam(String name, String value)
        Add a query parameter to the URI returned by Forward.getPath().
```

## Page Flow API Reference

```
public final addQueryParam(String name)
    void          Add a query parameter with no value to the URI returned by
                  Forward.getPath().

    public
    boolean doesResolve()
        Tell whether Forward.getPath() will be successful, i.e., whether one of the
        following two conditions is met:
            ◇ the name around which this object was constructed resolves to a path
              defined by a @jpf:forward, or
            ◇ this object was constructed around an explicit path, by Forward. or
              Forward..

    protected
    ActionForward findForward(String forwardName)
        Resolves the forward with the given name, from the stored ActionMapping if
        possible, or from the stored alternate ModuleConfig as a last resort.

    public
    boolean forwardsToPageFlow()
        Tell whether the URI returned by Forward.getPath() is for a page flow.

    public final
    ActionForm[] getOutputForms()
        Get all form-beans attached to this forward through
        Forward.addOutputForm(ActionForm) or Forward..

    public Map
        getPageInputs()
            Get all page inputs that have been set on this Forward.

    public
    String getPath()
        Get the URI path associated with this object.

    public
    String getQueryString()
        Get the query string that will be appended to the URI returned by
        Forward.getPath().

    public int
        getReturnToType()
            Get the type of return, if this is a return-to type.

    public
    String getReturnToTypeAsString()
        Get the type of return as a String, if this is a return-to type.

    public void
        initialize(ActionMapping mapping, FlowController flowController)
            Set the current ActionMapping and associated FlowController.

    public
    boolean isNestedReturn()
        Tell whether returning this forward from an action method will cause a return
        from a nested PageFlowController.

    public
    boolean isRedirect()
        Tell whether the URI resolved by this Forward should be redirected to.
```

public **isReturnToAction()**  
 boolean      Tell whether returning this forward from an action method will cause the previous action to be re-run.

public  
 boolean **isReturnToPage()**  
             Tell whether returning this forward from an action method will cause a previous page to be displayed.

public void  
             **setPath(String path)**  
             Set the path to be returned by `Forward.getPath()`.

public void  
             **setQueryString(String queryString)**  
             Set the query string that will be appended to the URI returned by `Forward.getPath()`.

public void  
             **setRedirect(boolean redirect)**  
             Set whether the URI resolved by this Forward should be redirected to.

### Methods from `org.apache.struts.config.ForwardConfig`

`freeze`, `getContextRelative`, `getName`, `getRedirect`, `setContextRelative`, `setName`, `toString`

### Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

## Field Detail

### RETURN\_TO\_ACTION

```
public static final int RETURN_TO_ACTION
```

---

### RETURN\_TO\_CURRENT\_PAGE

```
public static final int RETURN_TO_CURRENT_PAGE
```

---

### RETURN\_TO\_NONE

```
public static final int RETURN_TO_NONE
```

---

### RETURN\_TO\_PAGE

```
public static final int RETURN_TO_PAGE
```

---

## RETURN\_TO\_PREVIOUS\_ACTION

```
public static final int RETURN_TO_PREVIOUS_ACTION
```

---

## RETURN\_TO\_PREVIOUS\_PAGE

```
public static final int RETURN_TO_PREVIOUS_PAGE
```

## Constructor Detail

### Forward

```
public Forward(String forwardName,
               String pageInputName,
               Object pageInputValue)
```

Constructor which accepts the name of a forward defined by the `@jpf:forward` tag. The values returned from `Forward.getPath()`, `Forward.getRedirect()` and `Forward.contextRelative` are resolved from this forward. Also accepts a named page input to make available in the request, through `PageFlowUtils.getPageInput(String, ServletRequest)`.

---

### Forward

```
public Forward(String forwardName,
               ActionForm outputForm)
```

Constructor which accepts the name of a forward defined by the `@jpf:forward` tag. The values returned from `Forward.getPath()`, `Forward.getRedirect()` and `Forward.contextRelative` are resolved from this forward. Also accepts a form bean to make available in the request (or user session, as appropriate).

---

### Forward

```
public Forward(String forwardName)
```

Constructor which accepts the name of a forward defined by the `@jpf:forward` tag. The values returned from `Forward.getPath()`, `Forward.getRedirect()` and `Forward.contextRelative` are resolved from this forward.

---

### Forward

```
public Forward(URI path,
               boolean redirect)
```

Constructs a Forward that returns the given URI for `Forward.getPath()`.

---

## Forward

```
public Forward(URI path)
```

Constructs a Forward that returns the given URI for `Forward.getPath()`. By default the Forward will cause server forward (not a browser redirect); to change this, use `Forward.setRedirect(boolean)`.

---

## Forward

```
public Forward(URL path)
```

Constructs a Forward that returns the given URL for `Forward.getPath()`. Because the URL path is absolute by nature, this Forward will cause a browser redirect.

## Method Detail

### addOutputForm(ActionForm) Method

```
public final void addOutputForm(ActionForm form)
```

Add a form bean that will be made available in the request (or user session, as appropriate) if this Forward is returned by an action method in a `PageFlowController`. Specifically, each form bean is stored as a request attribute with a name determined by `PageFlowUtils.getFormBeanName(ActionForm, HttpServletRequest)`.

#### Parameters

*form*

the form bean instance to add.

---

### addPageInput(String, Object) Method

```
public void addPageInput(String name,
                        Object value)
```

Adds a page input that will be made available in the request, through `PageFlowUtils.getPageInput(String, ServletRequest)`.

#### Parameters

*name*

the name of the page input.



*value*

the page input value.

---

## addQueryParam(String, String) Method

```
public void addQueryParam(String name,  
                          String value)
```

Add a query parameter to the URI returned by `Forward.getPath()`.

### Parameters

*name*

the name of the query parameter.

*value*

the value of the query parameter, or `null` if there is no value.

---

## addQueryParam(String) Method

```
public final void addQueryParam(String name)
```

Add a query parameter with no value to the URI returned by `Forward.getPath()`.

### Parameters

*name*

the name of the query parameter.

---

## doesResolve() Method

```
public boolean doesResolve()
```

Tell whether `Forward.getPath()` will be successful, i.e., whether one of the following two conditions is met:

- the name around which this object was constructed resolves to a path defined by a `@jpf:forward`, or
- this object was constructed around an explicit path, by `Forward.` or `Forward..`

### Returns

*true if this forward does resolve to a URI path.*

---

## findForward(String) Method

```
protected ActionForward findForward(String forwardName)
```

Resolves the forward with the given name, from the stored ActionMapping if possible, or from the stored alternate ModuleConfig as a last resort.

### Parameters

*forwardName*

the name of the forward to resolve.

### Returns

*the resolved ActionForward, or null if none is found.*

---

## forwardsToPageFlow() Method

```
public boolean forwardsToPageFlow()
```

Tell whether the URI returned by `Forward.getPath()` is for a page flow.

### Returns

*true if the URI returned by `Forward.getPath()` is for a page flow, i.e., if it ends in ".jpf".*

---

## getOutputForms() Method

```
public final ActionForm[] getOutputForms()
```

Get all form-beans attached to this forward through `Forward.addOutputForm(ActionForm)` or `Forward..`

### Returns

*an array of ActionForm instances that are attached to this forward.*

---

## getPageInputs() Method

```
public Map getPageInputs()
```

Get all page inputs that have been set on this Forward.

### Returns

*a Map of name/value pairs representing all page inputs.*

**Related Topics**

`Forward.addPageInput(String, Object)`

---

**getPath() Method**

```
public String getPath()
```

Get the URI path associated with this object. Resolve it from the name of a forward (@jpf:forward) if necessary.

**Overrides**

`ForwardConfig.getPath()`

**Returns**

*a String that is the URI path.*

**Related Topics**

`Forward.`  
`Forward.`  
`Forward.`  
`Forward.`  
`Forward.setPath(String)`

---

**getQueryString() Method**

```
public String getQueryString()
```

Get the query string that will be appended to the URI returned by `Forward.getPath()`.

**Returns**

*the query string that will be appended to the URI, or null if there is no query string.*

---

**getReturnToType() Method**

```
public int getReturnToType()
```

Get the type of return, if this is a return-to type.

**Returns**

*one of the following values: `Forward.RETURN_TO_CURRENT_PAGE`,  
`Forward.RETURN_TO_PREVIOUS_PAGE`, `Forward.RETURN_TO_PAGE`,  
`Forward.RETURN_TO_PREVIOUS_ACTION`, `Forward.RETURN_TO_ACTION`, or*

*Forward.RETURN\_TO\_NONE if this Forward is not a return-to type.*

### Related Topics

`Forward.isReturnToAction()`  
`Forward.isReturnToPage()`

---

## getReturnToTypeAsString() Method

```
public String getReturnToTypeAsString()
```

Get the type of return as a String, if this is a return-to type.

### Returns

*one of the following values: currentPage, previousPage, page, (deprecated), previousAction, action (deprecated), or null if this is not a return-to type.*

### Related Topics

`Forward.isReturnToAction()`  
`Forward.isReturnToPage()`

---

## initialize(ActionMapping, FlowController) Method

```
public void initialize(ActionMapping mapping,  
                      FlowController flowController)
```

Set the current ActionMapping and associated FlowController. Normally, this method is called by the framework, but you can use it to initialize the Forward object in order to call `Forward.getPath()`.

### Parameters

*mapping*

the current ActionMapping; this can be obtained from `FlowController.GetMapping()`.

*flowController*

the object in which to look for referenced return-form members.

---

## isNestedReturn() Method

```
public boolean isNestedReturn()
```

Tell whether returning this forward from an action method will cause a return from a nested PageFlowController.

## Returns

*true if this forward will cause a return from nesting.*

---

## isRedirect() Method

```
public boolean isRedirect()
```

Tell whether the URI resolved by this Forward should be redirected to.

## Returns

*true if the controller will send a browser redirect to the URI for this Forward; false if it will do a server forward to the URI.*

---

## isReturnToAction() Method

```
public boolean isReturnToAction()
```

Tell whether returning this forward from an action method will cause the previous action to be re-run.

## Returns

*true if returning this forward from an action method will cause the previous action to be re-run, i.e., whether the URI returned by `Forward.getPath()` will end in "previous-action-name.do".*

---

## isReturnToPage() Method

```
public boolean isReturnToPage()
```

Tell whether returning this forward from an action method will cause a previous page to be displayed.

## Returns

*true if returning this forward from an action method will cause a previous page to be displayed.*

---

## setPath(String) Method

```
public void setPath(String path)
```

Set the path to be returned by `Forward.getPath()`. This overrides any path or forward name set in a constructor.

## Overrides

`ForwardConfig.setPath(String)`

## Parameters

*path*

the URI to be returned by `Forward.getPath()`.

---

## setQueryString(String) Method

```
public void setQueryString(String queryString)
```

Set the query string that will be appended to the URI returned by `Forward.getPath()`.

## Parameters

*queryString*

the query string that will be appended to the URI. If this string does not start with ' ? ', then this character will be prepended; if the string is `null`, the query string will be removed.

---

## setRedirect(boolean) Method

```
public void setRedirect(boolean redirect)
```

Set whether the URI resolved by this `Forward` should be redirected to.

## Overrides

`ForwardConfig.setRedirect(boolean)`

## Parameters

*redirect*

if `true`, the controller will send a browser redirect to the URI for this `Forward`; otherwise, it will do a server forward to the URI.

# GlobalApp Class

*public class* **GlobalApp**

extends FlowController  
implements PageFlowConstants

Base class for user-defined global state and fallback controller logic. A specific instance of this class (based on the Global.app source file in /WEB-INF/src/global) is created and stored in the user session before any PageFlowController is created, and unless it is explicitly removed from the session, it is not freed until the session is invalidated.

Actions that are not handled by the current PageFlowController "bubble" up to the current GlobalApp instance, which gets a chance to handle them in user-defined action methods. Uncaught exceptions also bubble through the GlobalApp instance, which has a chance to catch them.

Data in the current GlobalApp instance can be accessed by databinding tags using the globalApp scope.

## Related Topics

PageFlowController

---

## Hierarchy

Object  
  Action  
    FlowController  
      GlobalApp

## All Implemented Interfaces

EventListener, HttpSessionBindingListener, PageFlowConstants,  
Serializable

---

## Field Summary

### Fields from org.apache.struts.action.Action

ACTION\_SERVLET\_KEY, APPLICATION\_KEY, DATA\_SOURCE\_KEY, defaultLocale,  
ERROR\_KEY, EXCEPTION\_KEY, FORM\_BEANS\_KEY, FORWARDS\_KEY, LOCALE\_KEY,  
MAPPING\_KEY, MAPPINGS\_KEY, MESSAGE\_KEY, MESSAGES\_KEY, MULTIPART\_KEY,  
PLUG\_INS\_KEY, REQUEST\_PROCESSOR\_KEY, servlet, SERVLET\_KEY,  
TRANSACTION\_TOKEN\_KEY

### Fields from interface com.bea.wlw.netui.pageflow.PageFlowConstants

ACTION\_EXTENSION, ACTION\_EXTENSION\_LEN, ACTION\_OVERRIDE,  
BEGIN\_ACTION\_NAME, BEGIN\_ACTION\_PATH, DEFAULT\_FORWARD\_OVERFLOW\_COUNT,  
DEFAULT\_NESTING\_OVERFLOW\_COUNT, FORWARD\_OVERFLOW\_COUNT\_PARAM,  
GLOBALAPP\_CLASSNAME, GLOBALAPP\_MEMBER\_NAME,  
GLOBALAPP\_MODULE\_CONTEXT\_PATH, GLOBALAPP\_SOURCEFILE\_NAME,  
INTERNAL\_VAR\_PREFIX, ISNESTABLE\_VAR\_NAME, JPF\_EXTENSION,  
JPF\_MODULE\_CONFIG\_EXTENSION, JPF\_MODULE\_CONFIG\_GEN\_DIR,  
JPF\_MODULE\_CONFIG\_PREFIX, NESTING\_OVERFLOW\_COUNT\_PARAM,  
PROCESS\_ROLES\_FAILED\_ATTR, WEBINF\_DIR

## Constructor Summary

*GlobalApp()*

## Method Summary

```
public
    void ensureFailover(HttpServletRequest request)
        Ensures that any changes to this FlowController will be replicated in a cluster (for failover),
        even if the replication scheme uses a change-detection algorithm that relies on
        HttpSession.setAttribute to be aware of changes.

public
    String getDisplayName()

public
    String getModulePath()
        Get the Struts module path for Global.app.
```

### Methods from **com.bea.wlw.netui.pageflow.FlowController**

afterAction, alwaysTrackPreviousAction, alwaysTrackPreviousPage,  
beforeAction, execute, getActionMethod, getActions,  
getCurrentActionName, getMapping, getModuleConfig, getRequest,  
getResponse, getServlet, getSession, handleException,  
invokeActionMethod, isAction, isNestable, isPageFlow, login, logout,  
onCreate, onDestroy, onDestroy, resolveAction, retrieveUserLocale,  
sendError, valueBound, valueUnbound

### Methods from **org.apache.struts.action.Action**

execute, execute, generateToken, getDataSource, getDataSource,  
getLocale, getResources, getResources, getResources, getServlet,  
isCancelled, isTokenValid, isTokenValid, perform, perform, resetToken,  
saveErrors, saveMessages, saveToken, setLocale, setServlet, toHex



### Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

### Methods from interface `javax.servlet.http.HttpSessionBindingListener`

`valueBound`, `valueUnbound`

## Constructor Detail

### GlobalApp

```
public GlobalApp()
```

## Method Detail

### ensureFailover(HttpServletRequest) Method

```
public void ensureFailover(HttpServletRequest request)
```

Ensures that any changes to this FlowController will be replicated in a cluster (for failover), even if the replication scheme uses a change–detection algorithm that relies on `HttpSession.setAttribute` to be aware of changes.

#### Overrides

```
FlowController.ensureFailover(HttpServletRequest)
```

#### Parameters

*request*

the current `HttpServletRequest`

---

### getDisplayName() Method

```
public String getDisplayName()
```

#### Overrides

```
FlowController.getDisplayName()
```

---

### getModulePath() Method

```
public String getModulePath()
```

Get the Struts module path for `Global.app`.

***Overrides***

`FlowController.getModulePath()`

**Returns**

*"/-global".*

---

# HttpRedirector Interface

*public interface **HttpRedirector***

---

## *All Known Implementing Classes*

PageFlowRequestProcessor.DefaultHttpRedirector

---

## Method Summary

```
public
    void forward(String uri, HttpServletRequest request, HttpServletResponse response,
                  ServletContext servletContext)

public
    void redirect(String uri, HttpServletRequest request, HttpServletResponse
                  response, ServletContext servletContext)
```

## Method Detail

### **forward(String, HttpServletRequest, HttpServletResponse, ServletContext)** Method

```
public void forward(String uri,
                    HttpServletRequest request,
                    HttpServletResponse response,
                    ServletContext servletContext)
    throws IOException, ServletException
```

#### Exceptions

*IOException*  
*ServletException*

---

### **redirect(String, HttpServletRequest, HttpServletResponse, ServletContext)** Method

```
public void redirect(String uri,
                    HttpServletRequest request,
                    HttpServletResponse response,
                    ServletContext servletContext)
    throws IOException, ServletException
```

#### Exceptions

*IOException*  
*ServletException*



# IllegalOutputFormException Class

*public abstract class IllegalOutputFormException*

extends PageFlowException

Base type for errors related to output forms on Forwards.

## Related Topics

Forward.addOutputForm(ActionForm)

---

## Hierarchy

```
Object
  Throwable
    Exception
      RuntimeException
        PageFlowException
          IllegalOutputFormException
```

## All Implemented Interfaces

Serializable

## Direct Known Subclasses

IllegalOutputFormTypeException, IllegalRedirectOutputFormException

---

## Constructor Summary

*IllegalOutputFormException(String forwardName, String actionName,  
FlowController flowController, String outputFormType)*

## Method Summary

```
public
String getForwardName()
    Get the name of the relevant Forward.
```

```
public
String getOutputFormType()
    Get the type name of the relevant output form.
```

```
public setForwardName(String forwardName)
    void          Set the name of the relevant Forward.
public
    void setOutputFormType(String outputFormType)
        Set the type name of the relevant output form.
```

#### Methods from `com.bea.wlw.netui.pageflow.PageFlowException`

```
getActionName, getFlowController, getFlowControllerURI,
getLocalizedMessage, getMessage, getMessageArgs, getMessageParts,
printError, sendError
```

#### Methods from `java.lang.Throwable`

```
fillInStackTrace, getCause, getLocalizedMessage, getMessage,
getStackTrace, initCause, printStackTrace, printStackTrace,
printStackTrace, setStackTrace, toString
```

#### Methods from class `java.lang.Object`

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll,
toString, wait, wait, wait
```

## Constructor Detail

### `IllegalOutputFormException`

```
public IllegalOutputFormException(String forwardName,
                                     String actionName,
                                     FlowController flowController,
                                     String outputFormType)
```

## Method Detail

### `getForwardName()` Method

```
public String getForwardName()
```

Get the name of the relevant Forward.

#### Returns

*a String that is the name of the relevant Forward.*

---

## getOutputFormType() Method

```
public String getOutputFormType()
```

Get the type name of the relevant output form.

### Returns

*a String that is the type name of the relevant output form.*

---

## setForwardName(String) Method

```
public void setForwardName(String forwardName)
```

Set the name of the relevant Forward.

### Parameters

*forwardName*  
a String that is the name of the relevant Forward.

---

## setOutputFormType(String) Method

```
public void setOutputFormType(String outputFormType)
```

Set the type name of the relevant output form.

### Parameters

*outputFormType*  
a String that is the type name of the relevant output form.

# IllegalOutputFormTypeException Class

*public class **IllegalOutputFormTypeException***

extends `IllegalOutputFormException`

Exception that occurs when the first output form for a Forward resolves to a `@jpf:forward` whose `return-form` or `return-form-type` attribute demands a different form type.

## Related Topics

`Forward.addOutputForm(ActionForm)`

---

## Hierarchy

```
Object
  Throwable
    Exception
      RuntimeException
        PageFlowException
          IllegalOutputFormException
            IllegalOutputFormTypeException
```

## All Implemented Interfaces

```
Serializable
```

---

## Constructor Summary

***IllegalOutputFormTypeException**(String forwardName, String actionName, FlowController flowController, String outputFormType, String requiredType)*

## Method Summary

```
protected
Object[] getMessageArgs()
public
String[] getMessageParts()
public
String getRequiredType()
    Get the name of the required form type.
```



**Methods from `com.bea.wlw.netui.pageflow.IllegalOutputFormException`**

`getForwardName`, `getOutputFormType`, `setForwardName`, `setOutputFormType`

**Methods from `com.bea.wlw.netui.pageflow.PageFlowException`**

`getActionName`, `getFlowController`, `getFlowControllerURI`,  
`getLocalizedMessage`, `getMessage`, `printError`, `sendError`

**Methods from `java.lang.Throwable`**

`fillInStackTrace`, `getCause`, `getLocalizedMessage`, `getMessage`,  
`getStackTrace`, `initCause`, `printStackTrace`, `printStackTrace`,  
`printStackTrace`, `setStackTrace`, `toString`

**Methods from class `java.lang.Object`**

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`,  
`toString`, `wait`, `wait`, `wait`

## Constructor Detail

### `IllegalOutputFormTypeException`

```
public IllegalOutputFormTypeException(String forwardName,  
                                     String actionName,  
                                     FlowController flowController,  
                                     String outputFormType,  
                                     String requiredType)
```

## Method Detail

### `getMessageArgs()` Method

```
protected Object[] getMessageArgs()
```

*Overrides*

```
PageFlowException.getMessageArgs()
```

---

### `getMessageParts()` Method

```
public String[] getMessageParts()
```

*Overrides*

```
PageFlowException.getMessageParts()
```

---

## **getRequiredType() Method**

```
public String getRequiredType()
```

Get the name of the required form type.

### **Returns**

*a String that is the name of the required form type.*

# IllegalPageInputException Class

*public class **IllegalPageInputException***

extends PageFlowException

Exception that occurs when a page input has been added to a Forward that resolves to a @jpf:forward marked redirect="true". Page inputs may not be used on redirect forwards.

---

## **Hierarchy**

```
Object
  Throwable
    Exception
      RuntimeException
        PageFlowException
          IllegalPageInputException
```

## **All Implemented Interfaces**

```
Serializable
```

---

## **Constructor Summary**

***IllegalPageInputException***(String forwardName, String actionName, FlowController flowController, String pageInputName)

Constructor.

## **Method Summary**

```
public
String getForwardName()
    Get the name of the relevant Forward.

protected
Object[] getMessageArgs()
    public
String[] getMessageParts()
    public
String getPageInputName()
    Get the name of the relevant page input.
```

```
public setForwardName(String forwardName)
    void          Set the name of the relevant Forward.
public
    void setPageInputName(String pageInputName)
        Set the name of the relevant page input.
```

#### Methods from `com.bea.wlw.netui.pageflow.PageFlowException`

`getActionName`, `getFlowController`, `getFlowControllerURI`,  
`getLocalizedMessage`, `getMessage`, `printError`, `sendError`

#### Methods from `java.lang.Throwable`

`fillInStackTrace`, `getCause`, `getLocalizedMessage`, `getMessage`,  
`getStackTrace`, `initCause`, `printStackTrace`, `printStackTrace`,  
`printStackTrace`, `setStackTrace`, `toString`

#### Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`,  
`toString`, `wait`, `wait`, `wait`

## Constructor Detail

### `IllegalPageInputException`

```
public IllegalPageInputException(String forwardName,
                                String actionName,
                                FlowController flowController,
                                String pageInputName)
```

Constructor.

## Method Detail

### `getForwardName()` Method

```
public String getForwardName()
```

Get the name of the relevant Forward.

#### Returns

*a String that is the name of the relevant Forward.*

---

## getMessageArgs() Method

```
protected Object[] getMessageArgs()
```

### *Overrides*

```
PageFlowException.getMessageArgs()
```

---

## getMessageParts() Method

```
public String[] getMessageParts()
```

### *Overrides*

```
PageFlowException.getMessageParts()
```

---

## getPageInputName() Method

```
public String getPageInputName()
```

Get the name of the relevant page input.

### **Returns**

*a String that is the name of the relevant page input.*

---

## setForwardName(String) Method

```
public void setForwardName(String forwardName)
```

Set the name of the relevant Forward.

### **Parameters**

*forwardName*

a String that is the name of the relevant Forward.

---

## setPageInputName(String) Method

```
public void setPageInputName(String pageInputName)
```

Set the name of the relevant page input.

### **Parameters**

*pageInputName*

a String that is the name of the relevant page input.



# IllegalRedirectOutputFormException Class

*public class **IllegalRedirectOutputFormException***

extends `IllegalOutputFormException`

Exception that occurs when an output form has been added to a Forward that resolves to a `@jpf:forward` marked `redirect="true"`. Output forms may not be used on redirect forwards.

## Related Topics

`Forward.addOutputForm(ActionForm)`

---

## Hierarchy

```
Object
  Throwable
    Exception
      RuntimeException
        PageFlowException
          IllegalOutputFormException
            IllegalRedirectOutputFormException
```

## All Implemented Interfaces

`Serializable`

---

## Constructor Summary

***IllegalRedirectOutputFormException**(String forwardName, String actionName, FlowController flowController, String outputFormType)*

## Method Summary

```
protected
  Object[] getMessageArgs()
  public
  String[] getMessageParts()
```

Methods from `com.bea.wlw.netui.pageflow.IllegalOutputFormException`

`getForwardName`, `getOutputFormType`, `setForwardName`, `setOutputFormType`

#### Methods from `com.bea.wlw.netui.pageflow.PageFlowException`

`getActionName`, `getFlowController`, `getFlowControllerURI`,  
`getLocalizedMessage`, `getMessage`, `printError`, `sendError`

#### Methods from `java.lang.Throwable`

`fillInStackTrace`, `getCause`, `getLocalizedMessage`, `getMessage`,  
`getStackTrace`, `initCause`, `printStackTrace`, `printStackTrace`,  
`printStackTrace`, `setStackTrace`, `toString`

#### Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`,  
`toString`, `wait`, `wait`, `wait`

## Constructor Detail

### `IllegalRedirectOutputFormException`

```
public IllegalRedirectOutputFormException(String forwardName,  
                                         String actionName,  
                                         FlowController flowController,  
                                         String outputFormType)
```

## Method Detail

### `getMessageArgs()` Method

```
protected Object[] getMessageArgs()
```

#### *Overrides*

```
PageFlowException.getMessageArgs()
```

---

### `getMessageParts()` Method

```
public String[] getMessageParts()
```

#### *Overrides*

```
PageFlowException.getMessageParts()
```



# InfiniteReturnToActionException Class

*public class **InfiniteReturnToActionException***

extends PageFlowException

Exception that occurs when the user invokes an action that is qualified with a @jpf:forward return-to="previousAction" annotation, but the previous action was the same as the current action (an infinite loop).

---

## **Hierarchy**

```
Object
  Throwable
    Exception
      RuntimeException
        PageFlowException
          InfiniteReturnToActionException
```

## **All Implemented Interfaces**

Serializable

---

## **Constructor Summary**

***InfiniteReturnToActionException**(String actionName, FlowController fc)*

## **Method Summary**

```
protected
  Object[] getMessageArgs()
protected
  String[] getMessageParts()
```

### **Methods from com.bea.wlw.netui.pageflow.PageFlowException**

getActionName, getFlowController, getFlowControllerURI,  
getLocalizedMessage, getMessage, printError, sendError

### **Methods from java.lang.Throwable**

`fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString`

#### Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

## Constructor Detail

### InfiniteReturnToActionException

```
public InfiniteReturnToActionException(String actionName,  
                                       FlowController fc)
```

## Method Detail

### getMessageArgs() Method

```
protected Object[] getMessageArgs()
```

#### *Overrides*

```
PageFlowException.getMessageArgs()
```

---

### getMessageParts() Method

```
protected String[] getMessageParts()
```

#### *Overrides*

```
PageFlowException.getMessageParts()
```

# LoginExpiredException Class

*public class LoginExpiredException*

extends NotLoggedInException

Exception thrown when NotLoggedInException would be thrown, *and* when the current HttpServletRequest refers to a session that no longer exists.

---

## Hierarchy

```
Object
  Throwable
    Exception
      RuntimeException
        PageFlowException
          NotLoggedInException
            LoginExpiredException
```

## All Implemented Interfaces

Serializable

---

## Constructor Summary

*LoginExpiredException(String actionName, FlowController fc)*

## Method Summary

```
public
String[] getMessageParts()
```

Methods from **com.bea.wlw.netui.pageflow.NotLoggedInException**

getMessageArgs,

Methods from **com.bea.wlw.netui.pageflow.PageFlowException**

```
getActionName, getFlowController, getFlowControllerURI,
getLocalizedMessage, getMessage, getMessageArgs, getMessageParts,
printError, sendError
```

#### Methods from `java.lang.Throwable`

`fillInStackTrace`, `getCause`, `getLocalizedMessage`, `getMessage`,  
`getStackTrace`, `initCause`, `printStackTrace`, `printStackTrace`,  
`printStackTrace`, `setStackTrace`, `toString`

#### Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`,  
`toString`, `wait`, `wait`, `wait`

## Constructor Detail

### LoginExpiredException

```
public LoginExpiredException(String actionName,  
                             FlowController fc)
```

## Method Detail

### getMessageParts() Method

```
public String[] getMessageParts()
```

#### *Overrides*

```
NotLoggedInException.getMessageParts()
```

# NoCurrentPageFlowException Class

*public class NoCurrentPageFlowException*

extends `ReturnToException`

Action that occurs when the user invokes an action that is qualified with either a `@jpf:forward` `return-to="previousAction"` or `return-to="previousPage"`, but there is no current page flow. This happens in iterative development mode when you have modified files and caused the web application to be redeployed, or when the session expires.

---

## ***Hierarchy***

```
Object
  Throwable
    Exception
      RuntimeException
        PageFlowException
          ReturnToException
            NoCurrentPageFlowException
```

## ***All Implemented Interfaces***

```
Serializable
```

---

## **Constructor Summary**

*NoCurrentPageFlowException(String actionName, Forward fwd)*

## **Method Summary**

```
protected
  String[] getMessageParts()
```

Methods from `com.bea.wlw.netui.pageflow.internal.ReturnToException`

`getMessageArgs`, `getReturnToType`

Methods from `com.bea.wlw.netui.pageflow.PageFlowException`

`getActionName`, `getFlowController`, `getFlowControllerURI`,  
`getLocalizedMessage`, `getMessage`, `getMessageArgs`, `getMessageParts`,

`printError, sendError`

#### Methods from `java.lang.Throwable`

`fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString`

#### Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

## Constructor Detail

### NoCurrentPageFlowException

```
public NoCurrentPageFlowException(String actionName,  
                                Forward fwd)
```

## Method Detail

### getMessageParts() Method

```
protected String[] getMessageParts()
```

#### *Overrides*

`ReturnToException.getMessageParts()`

# NoMatchingActionMethodException Class

*public class NoMatchingActionMethodException*

extends PageFlowException

Exception that occurs when the current action method does not accept the type of form passed in the Forward to the action. This may happen when the user returned to the calling page flow, from a nested page flow, with a specified form (@jpf:forward return-form="form-name" or return-form-type="form-type", but no action in the calling page flow accepts that form type.

---

## Hierarchy

```

Object
  Throwable
    Exception
      RuntimeException
        PageFlowException
          NoMatchingActionMethodException
  
```

## All Implemented Interfaces

```

Serializable
  
```

---

## Constructor Summary

*NoMatchingActionMethodException(String actionName, ActionForm form, FlowController fc)*

## Method Summary

```

public
String getFormClassName()
protected
Object[] getMessageArgs()
protected
String[] getMessageParts()
  
```

### Methods from com.bea.wlw.netui.pageflow.PageFlowException

```

getActionName, getFlowController, getFlowControllerURI,
getLocalizedMessage, getMessage, printError, sendError
  
```

### Methods from `java.lang.Throwable`

`fillInStackTrace`, `getCause`, `getLocalizedMessage`, `getMessage`,  
`getStackTrace`, `initCause`, `printStackTrace`, `printStackTrace`,  
`printStackTrace`, `setStackTrace`, `toString`

### Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`,  
`toString`, `wait`, `wait`, `wait`

## Constructor Detail

### NoMatchingActionMethodException

```
public NoMatchingActionMethodException(String actionName,  
                                     ActionForm form,  
                                     FlowController fc)
```

## Method Detail

### getFormClassName() Method

```
public String getFormClassName()
```

---

### getMessageArgs() Method

```
protected Object[] getMessageArgs()
```

#### *Overrides*

```
PageFlowException.getMessageArgs()
```

---

### getMessageParts() Method

```
protected String[] getMessageParts()
```

#### *Overrides*

```
PageFlowException.getMessageParts()
```



# NoPreviousActionException Class

*public class NoPreviousActionException*

extends RuntimeException

Exception that occurs when the user attempts to execute an action marked with the @jpf:forward return-to="previousAction" annotation, but there is no previously-run action. For example, this could happen if the begin action returns a @jpf:forward marked return-to="previousAction".

---

## Hierarchy

```
Object
  Throwable
    Exception
      RuntimeException
        PageFlowException
          ReturnToException
            NoPreviousActionException
```

## All Implemented Interfaces

Serializable

---

## Constructor Summary

*NoPreviousActionException(String actionName, Forward fwd, FlowController fc)*

## Method Summary

protected  
String[] *getMessageParts()*

Methods from **com.bea.wlw.netui.pageflow.internal.ReturnToException**

getMessageArgs, getReturnToType

Methods from **com.bea.wlw.netui.pageflow.PageFlowException**

getActionName, getFlowController, getFlowControllerURI,  
getLocalizedMessage, getMessage, getMessageArgs, getMessageParts,  
printError, sendError

#### Methods from `java.lang.Throwable`

`fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString`

#### Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

## Constructor Detail

### NoPreviousActionException

```
public NoPreviousActionException(String actionName,  
                                Forward fwd,  
                                FlowController fc)
```

## Method Detail

### getMessageParts() Method

```
protected String[] getMessageParts()
```

#### *Overrides*

`ReturnToException.getMessageParts()`

# NoPreviousPageException Class

*public class NoPreviousPageException*

extends `ReturnToException`

Exception that occurs when the user attempts to execute an action marked with the `@jpf:forward return-to="previousPage"` annotation, but there is no previous page in the current page flow. This could happen if the begin action returns a `@jpf:forward` marked `return-to="currentPage"`.

---

## *Hierarchy*

```
Object
  Throwable
    Exception
      RuntimeException
        PageFlowException
          ReturnToException
            NoPreviousPageException
```

## *All Implemented Interfaces*

`Serializable`

---

## Constructor Summary

*NoPreviousPageException(String actionName, Forward fwd, FlowController fc)*

## Method Summary

protected  
String[] *getMessageParts()*

Methods from `com.bea.wlw.netui.pageflow.internal.ReturnToException`

`getMessageArgs`, `getReturnToType`

Methods from `com.bea.wlw.netui.pageflow.PageFlowException`

`getActionName`, `getFlowController`, `getFlowControllerURI`,  
`getLocalizedMessage`, `getMessage`, `getMessageArgs`, `getMessageParts`,  
`printError`, `sendError`

#### Methods from `java.lang.Throwable`

`fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString`

#### Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

## Constructor Detail

### NoPreviousPageException

```
public NoPreviousPageException(String actionName,  
                               Forward fwd,  
                               FlowController fc)
```

## Method Detail

### getMessageParts() Method

```
protected String[] getMessageParts()
```

#### *Overrides*

`ReturnToException.getMessageParts()`

# NotLoggedInException Class

*public class NotLoggedInException*

extends PageFlowException

Exception thrown when:

- An action method marked with @jpf:action login-required="true" is hit when there is no logged-in user, or
- An action method marked with @jpf:action roles-allowed="roles" is hit when there is no logged-in user.

If the requested session-ID is different than the current session-ID, the LoginExpiredException will be thrown instead of the NotLoggedInException.

---

## *Hierarchy*

```
Object
  Throwable
    Exception
      RuntimeException
        PageFlowException
          NotLoggedInException
```

## *All Implemented Interfaces*

Serializable

## *Direct Known Subclasses*

LoginExpiredException

---

## Constructor Summary

*NotLoggedInException(String actionName, FlowController fc)*

Construct with no error message.

## Method Summary

protected  
Object[] **getMessageArgs()**

```
public
String[] getMessageParts()
```

#### Methods from `com.bea.wlw.netui.pageflow.PageFlowException`

```
getActionName, getFlowController, getFlowControllerURI,
getLocalizedMessage, getMessage, printError, sendError
```

#### Methods from `java.lang.Throwable`

```
fillInStackTrace, getCause, getLocalizedMessage, getMessage,
getStackTrace, initCause, printStackTrace, printStackTrace,
printStackTrace, setStackTrace, toString
```

#### Methods from class `java.lang.Object`

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll,
toString, wait, wait, wait
```

## Constructor Detail

### NotLoggedInException

```
public NotLoggedInException(String actionName,
                           FlowController fc)
```

Construct with no error message.

## Method Detail

### getMessageArgs() Method

```
protected Object[] getMessageArgs()
```

#### *Overrides*

```
PageFlowException.getMessageArgs()
```

---

### getMessageParts() Method

```
public String[] getMessageParts()
```

#### *Overrides*

```
PageFlowException.getMessageParts()
```

# PageFlowActionServlet Class

*public class PageFlowActionServlet*

extends `AutoRegisterActionServlet`

ActionServlet that dynamically registers modules based on naming/location conventions for Struts configuration files that are generated by the Page Flow compiler. These files are located in `/WEB-INF/.pageflow-struts-generated`, and are named `jpf-struts-config-module-name.xml`. The user may specify additional `PageFlowActionServlet.DefaultModuleConfigLocator` classes in `web.xml`, using the "moduleConfigLocators" init-parameter; this allows the user to define other patterns for auto-registered Struts modules.

---

## Hierarchy

```
Object
  GenericServlet
    HttpServlet
      ActionServlet
        AutoRegisterActionServlet
          PageFlowActionServlet
```

## All Implemented Interfaces

`Serializable`, `Servlet`, `ServletConfig`

## Direct Known Subclasses

`DynamicSubappActionServlet`

---

## Nested Class Summary

```
public PageFlowActionServlet.DefaultModuleConfigLocator
static class Default ModuleConfigLocator that looks for Struts module configuration files
               according to the pattern
               "/WEB-INF/.pageflow-struts-generated/jpf-struts-config-<module>".

protected PageFlowActionServlet.LegacyModuleConfigLocator
static class ModuleConfigLocator that looks for legacy Struts module configuration files
               according to the pattern "/WEB-INF/jpf-struts-config-<module>".
```

Nested classes from `com.bea.wlw.netui.pageflow.AutoRegisterActionServlet`

`AutoRegisterActionServlet.ModuleConfigLocator`

## Field Summary

Fields from `com.bea.wlw.netui.pageflow.AutoRegisterActionServlet`

`MODULE_CONFIG_LOCATOR_CLASS_ATTR`

Fields from `org.apache.struts.action.ActionServlet`

`config`, `configDigester`, `convertNull`, `dataSources`, `debug`, `internal`,  
`internalName`, `log`, `processor`, `registrations`, `servletMapping`, `servletName`

## Constructor Summary

*PageFlowActionServlet()*

## Method Summary

<pre> protected AutoRegisterActionServlet.ModuleConfigLocator[] </pre>	<pre> <b><i>getDefaultModuleConfigLocators()</i></b> </pre>	<p>Get the base list of ModuleConfigLocators, to specify locations for auto-registered Struts modules.</p>
<pre> public String </pre>	<pre> <b><i>getModuleConfPath(String modulePath)</i></b> </pre>	<p>Get the webapp-relative path to the Struts module configuration file for a given module path.</p>
<pre> public void </pre>	<pre> <b><i>init(ServletConfig config)</i></b> </pre>	
<pre> protected void </pre>	<pre> <b><i>process(HttpServletRequest request, HttpServletResponse response)</i></b> </pre>	<p>This override of the base class process() registers a Struts module on the fly if the config file can be found in our standard place (named in our standard way), regardless of whether the module is configured in web.xml.</p>



**Methods from `com.bea.wlw.netui.pageflow.AutoRegisterActionServlet`**

`destroy`, `getInitParameter`, `getInitParameterNames`, `getModuleConfig`,  
`getModuleConfigLocators`, `init`, `registerModule`,

**Methods from `org.apache.struts.action.ActionServlet`**

`destroy`, `destroyApplications`, `destroyConfigDigester`, `destroyDataSources`,  
`destroyInternal`, `destroyModules`, `doGet`, `doPost`, `findDataSource`,  
`findFormBean`, `findForward`, `findMapping`, `getApplicationConfig`, `getDebug`,  
`getInternal`, `getModuleConfig`, `getRequestProcessor`, `getResources`, `init`,  
`initApplicationConfig`, `initApplicationDataSources`,  
`initApplicationMessageResources`, `initApplicationPlugIns`,  
`initConfigDigester`, `initDataSources`, `initInternal`, `initModuleConfig`,  
`initModuleDataSources`, `initModuleMessageResources`, `initModulePlugIns`,  
`initOther`, `initServlet`, `log`, `process`

**Methods from `javax.servlet.http.HttpServlet`**

`doDelete`, `doGet`, `doHead`, `doOptions`, `doPost`, `doPut`, `doTrace`,  
`getLastModified`, `service`, `service`

**Methods from `javax.servlet.GenericServlet`**

`destroy`, `getInitParameter`, `getInitParameterNames`, `getServletConfig`,  
`getServletContext`, `getServletInfo`, `getServletName`, `init`, `init`, `log`, `log`,  
`service`

**Methods from class `java.lang.Object`**

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`,  
`toString`, `wait`, `wait`, `wait`

**Methods from interface `javax.servlet.Servlet`**

`destroy`, `getServletConfig`, `getServletInfo`, `init`, `service`

**Methods from interface `javax.servlet.ServletConfig`**

`getInitParameter`, `getInitParameterNames`, `getServletContext`,  
`getServletName`

## Constructor Detail

### PageFlowActionServlet

```
public PageFlowActionServlet()
```

## Method Detail

### getDefaultModuleConfigLocators() Method

```
protected AutoRegisterActionServlet.ModuleConfigLocator[] getDefaultModuleConfigLocators()
```

Get the base list of ModuleConfigLocators, to specify locations for auto-registered Struts modules. By default, this ActionServlet auto-registers Struts modules whose configuration files are located at `"/WEB-INF/.pageflow-struts-generated/jpf-struts-config-<module>".` Overriding this method allows alternate locations to be specified. When an unrecognized Struts module is requested, each registered ModuleConfigLocator is queried for a possible path to the configuration file for the module. If the configuration file is found, the module is auto-registered against the file.

#### Overrides

```
AutoRegisterActionServlet.getDefaultModuleConfigLocators()
```

---

### getModuleConfPath(String) Method

```
public String getModuleConfPath(String modulePath)
```

Get the webapp-relative path to the Struts module configuration file for a given module path. By default, this is `"/WEB-INF/.pageflow-struts-generated/jpf-struts-config-<module>".` but alternate locations can be specified by adding PageFlowActionServlet.DefaultModuleConfigLocators.

#### Overrides

```
AutoRegisterActionServlet.getModuleConfPath(String)
```

#### Parameters

*modulePath*  
the Struts module path.

#### Returns

*a String that is the path to the Struts configuration file, relative to the web application root.*

#### Related Topics

```
PageFlowActionServlet.getDefaultModuleConfigLocators()
```

---

### init(ServletConfig) Method

```
public void init(ServletConfig config)
    throws ServletException
```

#### Overrides

```
AutoRegisterActionServlet.init(ServletConfig)
```

## Exceptions

*ServletException*

---

## process(HttpServletRequest, HttpServletResponse) Method

```
protected void process(HttpServletRequest request,  
                        HttpServletResponse response)  
    throws IOException, ServletException
```

*Description copied from* `AutoRegisterActionServlet.process(HttpServletRequest, HttpServletResponse)`

This override of the base class `process()` registers a Struts module on the fly if the config file can be found in our standard place (named in our standard way), regardless of whether the module is configured in `web.xml`.

### Overrides

```
AutoRegisterActionServlet.process(HttpServletRequest,  
HttpServletResponse)
```

## Exceptions

*IOException*

*ServletException*

---

# PageFlowActionServlet.DefaultModuleConfigLocator Class

*public static class PageFlowActionServlet.DefaultModuleConfigLocator*

extends Object

implements AutoRegisterActionServlet.ModuleConfigLocator, Serializable

Default ModuleConfigLocator that looks for Struts module configuration files according to the pattern "/WEB-INF/.pageflow-struts-generated/jpf-struts-config-*<module>*". An instance of this class is registered by default.

## Related Topics

PageFlowActionServlet.DefaultModuleConfigLocator.getDefaultModuleConfigLocators

---

## Hierarchy

Object

PageFlowActionServlet.DefaultModuleConfigLocator

## All Implemented Interfaces

AutoRegisterActionServlet.ModuleConfigLocator, Serializable

## Direct Known Subclasses

PageFlowActionServlet.LegacyModuleConfigLocator

## Enclosing class

PageFlowActionServlet

---

## Constructor Summary

*PageFlowActionServlet.DefaultModuleConfigLocator()*

## Method Summary

protected

String *getGenDir()*

```
public getModuleConfigPath(String moduleName)  
String          Get the webapp-relative path to a Struts module config file, based on the module  
                name.
```

### Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`,  
`toString`, `wait`, `wait`, `wait`

### Methods from

**interface `com.bea.wlw.netui.pageflow.AutoRegisterActionServlet.ModuleConfigLocator`**

`getModuleConfigPath`

## Constructor Detail

### **PageFlowActionServlet.DefaultModuleConfigLocator**

```
public PageFlowActionServlet.DefaultModuleConfigLocator()
```

## Method Detail

### **getGenDir() Method**

```
protected String getGenDir()
```

---

### **getModuleConfigPath(String) Method**

```
public String getModuleConfigPath(String moduleName)
```

*Description copied from*

`AutoRegisterActionServlet.ModuleConfigLocator.getModuleConfigPath(String)`  
Get the webapp-relative path to a Struts module config file, based on the module name.

### **Parameters**

*moduleName*

the name of the Struts module, e.g., "someModule" or "some/other/module".

### **Returns**

*the webapp-relative path to the Struts module config file.*

# PageFlowActionServlet.LegacyModuleConfigLocator Class

*protected static class* **PageFlowActionServlet.LegacyModuleConfigLocator**

extends `PageFlowActionServlet.DefaultModuleConfigLocator`

ModuleConfigLocator that looks for legacy Struts module configuration files according to the pattern `"/WEB-INF/jpf-struts-config-<module>"`. An instance of this class is registered by default.

## Related Topics

`PageFlowActionServlet.LegacyModuleConfigLocator.getDefaultModuleConfigLocators()`

---

## Hierarchy

Object

PageFlowActionServlet.DefaultModuleConfigLocator

PageFlowActionServlet.LegacyModuleConfigLocator

## All Implemented Interfaces

AutoRegisterActionServlet.ModuleConfigLocator, Serializable

## Enclosing class

PageFlowActionServlet

---

## Constructor Summary

*PageFlowActionServlet.LegacyModuleConfigLocator()*

## Method Summary

protected  
String *getGenDir()*

## Methods from

`com.bea.wlw.netui.pageflow.PageFlowActionServlet.DefaultModuleConfigLocator`

`getModuleConfigPath`

### Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`,  
`toString`, `wait`, `wait`, `wait`

### Methods from

**interface `com.bea.wlw.netui.pageflow.AutoRegisterActionServlet.ModuleConfigLocator`**

`getModuleConfigPath`

## Constructor Detail

### **PageFlowActionServlet.LegacyModuleConfigLocator**

`protected PageFlowActionServlet.LegacyModuleConfigLocator()`

## Method Detail

### **getGenDir() Method**

`protected String getGenDir()`

#### *Overrides*

`PageFlowActionServlet.DefaultModuleConfigLocator.getGenDir()`

# PageFlowConstants Interface

*public interface PageFlowConstants*

Constants related to Page Flows.

---

## All Known Implementing Classes

FlowController, PageFlowRequestProcessor, PageFlowUtils, GlobalApp

---

## Field Summary

public static final String	<b><i>ACTION_EXTENSION</i></b>	The extension for Struts action URIs used with page flows.
public static final int	<b><i>ACTION_EXTENSION_LEN</i></b>	int
public static final String	<b><i>ACTION_OVERRIDE</i></b>	The prefix for a button or imageButton that has an action.
public static final String	<b><i>BEGIN_ACTION_NAME</i></b>	The name of the special action in a user-defined PageFlowController (.jpf) that is processed when the URI for the .jpf is accessed.
public static final String	<b><i>BEGIN_ACTION_PATH</i></b>	String
public static final int	<b><i>DEFAULT_FORWARD_OVERFLOW_COUNT</i></b>	int
public static final int	<b><i>DEFAULT_NESTING_OVERFLOW_COUNT</i></b>	int
public static final String	<b><i>FORWARD_OVERFLOW_COUNT_PARAM</i></b>	String
public static final String	<b><i>GLOBALAPP_CLASSNAME</i></b>	The name of the GlobalApp class defined by the user for a web application.
public static final String	<b><i>GLOBALAPP_MEMBER_NAME</i></b>	String
public static		



```

final String GLOBALAPP_MODULE_CONTEXT_PATH
        String

        public
        static GLOBALAPP_SOURCEFILE_NAME
final String
        The name of the source file for the GlobalApp class defined by the user for a web
        application.

        public
        static INTERNAL_VAR_PREFIX
final String
        String

        public
        static ISNESTABLE_VAR_NAME
final String
        String

        public
        static JPF_EXTENSION
final String
        The file extension for user-defined PageFlowController source files.

        public
        static JPF_MODULE_CONFIG_EXTENSION
final String
        The file extensionfor Struts configuration files generated by the Page Flow compiler.

        public
        static JPF_MODULE_CONFIG_GEN_DIR
final String
        The webapp-relative directory for Struts module configuration files generated by the
        page flow compiler.

        public
        static JPF_MODULE_CONFIG_PREFIX
final String
        The filename prefix for Struts module configuration files generated by the page flow
        compiler.

        public
        static NESTING_OVERFLOW_COUNT_PARAM
final String
        String

        public
        static PROCESS_ROLES_FAILED_ATTR
final String
        String

        public
        static WEBINF_DIR
final String
        String

```

## Field Detail

### ACTION\_EXTENSION

```
public static final String ACTION_EXTENSION
```

The extension for Struts action URIs used with page flows.

---

## **ACTION\_EXTENSION\_LEN**

```
public static final int ACTION_EXTENSION_LEN
```

---

## **ACTION\_OVERRIDE**

```
public static final String ACTION_OVERRIDE
```

The prefix for a button or imageButton that has an action.

---

## **BEGIN\_ACTION\_NAME**

```
public static final String BEGIN_ACTION_NAME
```

The name of the special action in a user-defined PageFlowController (.jpf) that is processed when the URI for the .jpf is accessed.

---

## **BEGIN\_ACTION\_PATH**

```
public static final String BEGIN_ACTION_PATH
```

---

## **DEFAULT\_FORWARD\_OVERFLOW\_COUNT**

```
public static final int DEFAULT_FORWARD_OVERFLOW_COUNT
```

---

## **DEFAULT\_NESTING\_OVERFLOW\_COUNT**

```
public static final int DEFAULT_NESTING_OVERFLOW_COUNT
```

---

## **FORWARD\_OVERFLOW\_COUNT\_PARAM**

```
public static final String FORWARD_OVERFLOW_COUNT_PARAM
```

---

## **GLOBALAPP\_CLASSNAME**

```
public static final String GLOBALAPP_CLASSNAME
```

The name of the GlobalApp class defined by the user for a web application.

---

## GLOBALAPP\_MEMBER\_NAME

```
public static final String GLOBALAPP_MEMBER_NAME
```

---

## GLOBALAPP\_MODULE\_CONTEXT\_PATH

```
public static final String GLOBALAPP_MODULE_CONTEXT_PATH
```

---

## GLOBALAPP\_SOURCEFILE\_NAME

```
public static final String GLOBALAPP_SOURCEFILE_NAME
```

The name of the source file for the GlobalApp class defined by the user for a web application.

---

## INTERNAL\_VAR\_PREFIX

```
public static final String INTERNAL_VAR_PREFIX
```

---

## ISNESTABLE\_VAR\_NAME

```
public static final String ISNESTABLE_VAR_NAME
```

---

## JPF\_EXTENSION

```
public static final String JPF_EXTENSION
```

The file extension for user-defined PageFlowController source files.

---

## JPF\_MODULE\_CONFIG\_EXTENSION

```
public static final String JPF_MODULE_CONFIG_EXTENSION
```

The file extension for Struts configuration files generated by the Page Flow compiler.

---

## JPF\_MODULE\_CONFIG\_GEN\_DIR

```
public static final String JPF_MODULE_CONFIG_GEN_DIR
```

The webapp-relative directory for Struts module configuration files generated by the page flow compiler.

---

## **JPF\_MODULE\_CONFIG\_PREFIX**

```
public static final String JPF_MODULE_CONFIG_PREFIX
```

The filename prefix for Struts module configuration files generated by the page flow compiler.

---

## **NESTING\_OVERFLOW\_COUNT\_PARAM**

```
public static final String NESTING_OVERFLOW_COUNT_PARAM
```

---

## **PROCESS\_ROLES\_FAILED\_ATTR**

```
public static final String PROCESS_ROLES_FAILED_ATTR
```

---

## **WEBINF\_DIR**

```
public static final String WEBINF_DIR
```

# PageFlowContextListener Class

*public class PageFlowContextListener*

extends Object  
implements ServletContextListener

Performs various initialization at ServletContext–init time.

---

## **Hierarchy**

Object  
PageFlowContextListener

## **All Implemented Interfaces**

EventListener, ServletContextListener

---

## **Constructor Summary**

*PageFlowContextListener()*

## **Method Summary**

public  
void **contextDestroyed**(ServletContextEvent event)  
public  
void **contextInitialized**(ServletContextEvent event)

### **Methods from class java.lang.Object**

clone, equals, finalize, getClass, hashCode, notify, notifyAll,  
toString, wait, wait, wait

### **Methods from interface javax.servlet.ServletContextListener**

contextDestroyed, contextInitialized

## Constructor Detail

### PageFlowContextListener

```
public PageFlowContextListener()
```

## Method Detail

### contextDestroyed(ServletContextEvent) Method

```
public void contextDestroyed(ServletContextEvent event)
```

---

### contextInitialized(ServletContextEvent) Method

```
public void contextInitialized(ServletContextEvent event)
```

# PageFlowController Class

*public abstract class PageFlowController*

extends FlowController

Base class for user-defined state and controller logic associated with a particular web application directory. The source code for this class lives in the directory with a ".jpf" extension, and is automatically recompiled by the server whenever necessary. When a request arrives for any ".jpf", ".jsp", or ".do" in the directory, an instance of the PageFlowController class is set as the *current page flow*, and remains in the session until a different one becomes active.

The user's PageFlowController class handles *actions* that are most commonly raised by user interaction with JSP files. The actions are handled by *action methods* that perform arbitrary logic and determine the next URI to be displayed.

Actions that are not handled by the current PageFlowController "bubble" up to the current GlobalApp instance.

Data in the current PageFlowController instance can be accessed by databinding tags using the pageFlow scope.

## Related Topics

GlobalApp

---

## Hierarchy

```
Object
  Action
    FlowController
      PageFlowController
```

## All Implemented Interfaces

```
EventListener, HttpSessionBindingListener, PageFlowConstants,
Serializable
```

---

## Field Summary

### Fields from org.apache.struts.action.Action

```
ACTION_SERVLET_KEY, APPLICATION_KEY, DATA_SOURCE_KEY, defaultLocale,
ERROR_KEY, EXCEPTION_KEY, FORM_BEANS_KEY, FORWARDS_KEY, LOCALE_KEY,
MAPPING_KEY, MAPPINGS_KEY, MESSAGE_KEY, MESSAGES_KEY, MULTIPART_KEY,
```

PLUG\_INS\_KEY, REQUEST\_PROCESSOR\_KEY, servlet, SERVLET\_KEY,  
TRANSACTION\_TOKEN\_KEY

### Fields from interface `com.bea.wlw.netui.pageflow.PageFlowConstants`

ACTION\_EXTENSION, ACTION\_EXTENSION\_LEN, ACTION\_OVERRIDE,  
BEGIN\_ACTION\_NAME, BEGIN\_ACTION\_PATH, DEFAULT\_FORWARD\_OVERFLOW\_COUNT,  
DEFAULT\_NESTING\_OVERFLOW\_COUNT, FORWARD\_OVERFLOW\_COUNT\_PARAM,  
GLOBALAPP\_CLASSNAME, GLOBALAPP\_MEMBER\_NAME,  
GLOBALAPP\_MODULE\_CONTEXT\_PATH, GLOBALAPP\_SOURCEFILE\_NAME,  
INTERNAL\_VAR\_PREFIX, ISNESTABLE\_VAR\_NAME, JPF\_EXTENSION,  
JPF\_MODULE\_CONFIG\_EXTENSION, JPF\_MODULE\_CONFIG\_GEN\_DIR,  
JPF\_MODULE\_CONFIG\_PREFIX, NESTING\_OVERFLOW\_COUNT\_PARAM,  
PROCESS\_ROLES\_FAILED\_ATTR, WEBINF\_DIR

## Constructor Summary

### *PageFlowController()*

Default  
constructor.

## Method Summary

public void

***ensureFailover***(*HttpServletRequest request*)

Ensures that any changes to this FlowController will be replicated in a cluster (for failover), even if the replication scheme uses a change-detection algorithm that relies on HttpSession.setAttribute to be aware of changes.

protected String

***getCurrentForwardPath***()

Get the webapp-relative URI for the most recent page (in this page flow) shown to the user.

public final

PreviousPageInfo ***getCurrentPageInfo***()

Get information about the most recent page (in this page flow) shown to the user.

public String

***getDisplayName***()

public String

***getModulePath***()

Get the Struts module path for this page flow.

public final

PreviousActionInfo ***getPreviousActionInfo***()



## Page Flow API Reference

Get information about the most recent action run in this page flow.

protected String

***getPreviousActionURI()***

Get the URI for the most recent action in this PageFlowController.

protected

ActionForm ***getPreviousForm()***

Get the form instance from the most recent action execution in this PageFlowController.

protected String

***getPreviousForwardPath()***

Get the webapp-relative URI for the previous page (in this page flow) shown to the user.

public final

PreviousPageInfo ***getPreviousPageInfo()***

Get information about the previous page (in this page flow) shown to the user.

protected String

***getTaxonomy()***

Get the "resource taxonomy": a period-separated list that starts with the current web application name, continues through all of this PageFlowController's parent directories, and ends with this PageFlowController's class name.

public String

***getURI()***

Get the URI for addressing this PageFlowController.

protected boolean

***isNestable()***

Tell whether this PageFlowController can be "nested", i.e., if it can be invoked from another page flow with the intention of returning to the original one.

public boolean

***isPageFlow()***

Tell whether this is a PageFlowController.

public void

***valueUnbound(HttpSessionBindingEvent event)***

Callback when this page flow is removed from the user session.

### Methods from **com.bea.wlw.netui.pageflow.FlowController**

afterAction, alwaysTrackPreviousAction, alwaysTrackPreviousPage, beforeAction, execute, getActionMethod, getActions, getCurrentActionName, getMapping, getModuleConfig, getRequest, getResponse, getServlet, getSession, handleException, invokeActionMethod, isAction, login, logout, onCreate, onDestroy, onDestroy, resolveAction, retrieveUserLocale, sendError, valueBound,

### Methods from **org.apache.struts.action.Action**

`execute, execute, generateToken, getDataSource, getDataSource, getLocale, getResources, getResources, getResources, getServlet, isCancelled, isTokenValid, isTokenValid, perform, perform, resetToken, saveErrors, saveMessages, saveToken, setLocale, setServlet, toHex`

### Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

### Methods from interface `javax.servlet.http.HttpSessionBindingListener`

`valueBound, valueUnbound`

## Constructor Detail

### PageFlowController

```
public PageFlowController()
```

Default constructor.

## Method Detail

### ensureFailover(HttpServletRequest) Method

```
public void ensureFailover(HttpServletRequest request)
```

Ensures that any changes to this FlowController will be replicated in a cluster (for failover), even if the replication scheme uses a change-detection algorithm that relies on `HttpSession.setAttribute` to be aware of changes.

#### Overrides

```
FlowController.ensureFailover(HttpServletRequest)
```

#### Parameters

*request*

the current `HttpServletRequest`

---

### getCurrentForwardPath() Method

```
protected String getCurrentForwardPath()
```

Get the webapp-relative URI for the most recent page (in this page flow) shown to the user.

*Note: if the current page flow does not use a @jpf:forward with a return-to="previousPage" or return-to="currentPage" attribute, then this method will always return null by default. To enable it in this situation, add the following method to the page flow:*

```
protected boolean alwaysTrackPreviousPage()  
{  
    return true;  
}
```

### Returns

*a String that is the URI path for the most recent page shown to the user.*

### Related Topics

```
PageFlowController.getPreviousPageInfo()  
PageFlowController.getCurrentPageInfo()  
PageFlowController.getPreviousActionInfo()  
PageFlowController.getPreviousActionURI()  
PageFlowController.getPreviousForm()  
PageFlowController.getPreviousForwardPath()
```

---

## getCurrentPageInfo() Method

```
public final PreviousPageInfo getCurrentPageInfo()
```

Get information about the most recent page (in this page flow) shown to the user.

*Note: if the current page flow does not use a @jpf:forward with a return-to="previousPage" or return-to="currentPage" attribute, then this method will always return null by default. To enable it in this situation, add the following method to the page flow:*

```
protected boolean alwaysTrackPreviousPage()  
{  
    return true;  
}
```

### Returns

*a PreviousPageInfo with information about the most recent page shown to the user.*

### Related Topics

```
PageFlowController.getPreviousPageInfo()  
PageFlowController.getPreviousActionInfo()  
PageFlowController.getPreviousActionURI()  
PageFlowController.getPreviousForm()  
PageFlowController.getPreviousForwardPath()  
PageFlowController.getCurrentForwardPath()
```

---

## getDisplayName() Method

```
public String getDisplayName()
```

### Overrides

```
FlowController.getDisplayName()
```

---

## getModulePath() Method

```
public String getModulePath()
```

Get the Struts module path for this page flow.

### Overrides

```
FlowController.getModulePath()
```

### Returns

*a String that is the Struts module path for this controller, and which is also the directory path from the web application root to this PageFlowController (not including the action filename).*

---

## getPreviousActionInfo() Method

```
public final PreviousActionInfo getPreviousActionInfo()
```

Get information about the most recent action run in this page flow.

*Note: if the current page flow does not use a @jpf:forward with a return-to="previousAction" attribute, then this method will always return null by default. To enable it in this situation, add the following method to the page flow:*

```
protected boolean alwaysTrackPreviousAction()
{
    return true;
}
```

### Returns

*a PreviousActionInfo with information about the most recent action run in this page flow.*

### Related Topics

```
PageFlowController.getPreviousPageInfo()
PageFlowController.getCurrentPageInfo()
PageFlowController.getPreviousActionURI()
PageFlowController.getPreviousForm()
PageFlowController.getPreviousForwardPath()
PageFlowController.getCurrentForwardPath()
```

## getPreviousActionURI() Method

```
protected String getPreviousActionURI()
```

Get the URI for the most recent action in this PageFlowController.

*Note: if the current page flow does not use a @jpf:forward with a return-to="previousAction" attribute, then this method will always return null by default. To enable it in this situation, add the following method to the page flow:*

```
protected boolean alwaysTrackPreviousAction()  
{  
    return true;  
}
```

### Returns

*a String that is the most recent URI.*

### Related Topics

```
PageFlowController.getPreviousPageInfo()  
PageFlowController.getCurrentPageInfo()  
PageFlowController.getPreviousActionInfo()  
PageFlowController.getPreviousForm()  
PageFlowController.getPreviousForwardPath()  
PageFlowController.getCurrentForwardPath()
```

---

## getPreviousForm() Method

```
protected ActionForm getPreviousForm()
```

Get the form instance from the most recent action execution in this PageFlowController.

*Note: if the current page flow does not use a @jpf:forward with a return-to="previousAction" attribute, then this method will always return null by default. To enable it in this situation, add the following method to the page flow:*

```
protected boolean alwaysTrackPreviousAction()  
{  
    return true;  
}
```

### Returns

*the ActionForm instance from the most recent action execution, or null if there was no ActionForm instance.*

**Related Topics**

```

PageFlowController.getPreviousPageInfo()
PageFlowController.getCurrentPageInfo()
PageFlowController.getPreviousActionInfo()
PageFlowController.getPreviousActionURI()
PageFlowController.getPreviousForwardPath()
PageFlowController.getCurrentForwardPath()

```

---

**getPreviousForwardPath() Method**

```
protected String getPreviousForwardPath()
```

Get the webapp-relative URI for the previous page (in this page flow) shown to the user. The previous page is the one shown before the most recent page.

*Note: if the current page flow does not use a @jpf:forward with a return-to="previousPage" or return-to="currentPage" attribute, then this method will always return null by default. To enable it in this situation, add the following method to the page flow:*

```

protected boolean alwaysTrackPreviousPage()
{
    return true;
}

```

**Returns**

*a String that is the URI path for the previous page shown to the user.*

**Related Topics**

```

PageFlowController.getPreviousPageInfo()
PageFlowController.getCurrentPageInfo()
PageFlowController.getPreviousActionInfo()
PageFlowController.getPreviousActionURI()
PageFlowController.getPreviousForm()
PageFlowController.getCurrentForwardPath()

```

---

**getPreviousPageInfo() Method**

```
public final PreviousPageInfo getPreviousPageInfo()
```

Get information about the previous page (in this page flow) shown to the user. The previous page is the one shown before the most recent page.

*Note: if the current page flow does not use a @jpf:forward with a return-to="previousPage" or return-to="currentPage" attribute, then this method will always return null by default. To enable it in this situation, add the following method to the page flow:*

```
protected boolean alwaysTrackPreviousPage()  
{  
    return true;  
}
```

### Returns

*a PreviousPageInfo with information about the previous page shown to the user.*

### Related Topics

```
PageFlowController.getCurrentPageInfo()  
PageFlowController.getPreviousActionInfo()  
PageFlowController.getPreviousActionURI()  
PageFlowController.getPreviousForm()  
PageFlowController.getPreviousForwardPath()  
PageFlowController.getCurrentForwardPath()
```

---

## getTaxonomy() Method

```
protected String getTaxonomy()
```

Get the "resource taxonomy": a period-separated list that starts with the current web application name, continues through all of this PageFlowController's parent directories, and ends with this PageFlowController's class name.

---

## getURI() Method

```
public String getURI()
```

Get the URI for addressing this PageFlowController.

### Returns

*a String that is the URI which will execute the begin action on this PageFlowController.*

---

## isNestable() Method

```
protected boolean isNestable()
```

Tell whether this PageFlowController can be "nested", i.e., if it can be invoked from another page flow with the intention of returning to the original one. Page flows are declared to be nested using the `@jpf:controller nested="true"` annotation.

### Overrides

```
FlowController.isNestable()
```

## Returns

*true if this PageFlowController can be nested.*

---

## isPageFlow() Method

```
public boolean isPageFlow()
```

Tell whether this is a PageFlowController.

### Overrides

```
FlowController.isPageFlow()
```

## Returns

*true.*

---

## valueUnbound(HttpSessionBindingEvent) Method

```
public void valueUnbound(HttpSessionBindingEvent event)
```

Callback when this page flow is removed from the user session. Causes `PageFlowController.onDestroy()` to be called.

### Overrides

```
FlowController.valueUnbound(HttpSessionBindingEvent)
```



# PageFlowException Class

*public abstract class PageFlowException*

extends RuntimeException

Base class for PageFlow-related Exceptions.

---

## **Hierarchy**

```
Object
  Throwable
    Exception
      RuntimeException
        PageFlowException
```

## **All Implemented Interfaces**

Serializable

## **Direct Known Subclasses**

ActionNotFoundException, EmptyNestingStackException,  
IllegalOutputFormException, IllegalPageInputException,  
InfiniteReturnToActionException, NoMatchingActionMethodException,  
NotLoggedInException, ReturnToException, UnfulfilledRolesException,  
UnresolvableForwardException

---

## **Constructor Summary**

*PageFlowException(String actionName, FlowController fc)*

## **Method Summary**

```
public String
    getActionName()
        Get the name of the action associated with this exception.

    public
FlowController getFlowController()
        Get the related FlowController.

public String
```

***getFlowControllerURI()***

Get the name of the related FlowController.

public String

***getLocalizedMessage()***

public String

***getMessage()***

protected

abstract

Object[] ***getMessageArgs()***

protected

abstract

String[] ***getMessageParts()***

public void

***printError(PrintWriter writer)***

Print a formatted message.

public void

***sendError(HttpServletRequest request, HttpServletResponse response)***

Handle the error by writing a message to the response.

**Methods from java.lang.Throwable**

fillInStackTrace, getCause, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

**Methods from class java.lang.Object**

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructor Detail

### PageFlowException

```
public PageFlowException(String actionName,
                        FlowController fc)
```

## Method Detail

### getActionName() Method

```
public String getActionName()
```

Get the name of the action associated with this exception.

## Returns

*a String that is the name of the action associated with this exception.*

---

## getFlowController() Method

```
public FlowController getFlowController()
```

Get the related FlowController.

## Returns

*the FlowController associated with this exception.*

---

## getFlowControllerURI() Method

```
public String getFlowControllerURI()
```

Get the name of the related FlowController.

## Returns

*the class name of the FlowController associated with this exception.*

---

## getLocalizedMessage() Method

```
public String getLocalizedMessage()
```

### *Overrides*

```
Throwable.getLocalizedMessage()
```

---

## getMessage() Method

```
public String getMessage()
```

### *Overrides*

```
Throwable.getMessage()
```

---

## getMessageArgs() Method

```
protected abstract Object[] getMessageArgs()
```

---

## getMessageParts() Method

```
protected abstract String[] getMessageParts()
```

---

## printError(PrintWriter) Method

```
public void printError(PrintWriter writer)
```

Print a formatted message.

### Parameters

*writer*

a writer to which to print the formatted message.

---

## sendError(HttpServletRequest, HttpServletResponse) Method

```
public void sendError(HttpServletRequest request,  
                     HttpServletResponse response)  
    throws IOException
```

Handle the error by writing a message to the response.

### Parameters

*request*

the current HttpServletRequest

*response*

the current HttpServletResponse

### Exceptions

*IOException*

# PageFlowFileFilter Class

*public class PageFlowFileFilter*

extends Object  
implements FilenameFilter

File filter that accepts any PageFlowController source file (\*.jpf).

---

## **Hierarchy**

Object  
PageFlowFileFilter

## **All Implemented Interfaces**

FilenameFilter

---

## **Constructor Summary**

*PageFlowFileFilter()*

## **Method Summary**

public  
boolean **accept**(File dir, String name)

### **Methods from class java.lang.Object**

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### **Methods from interface java.io.FilenameFilter**

accept

## **Constructor Detail**

## PageFlowFileFilter

```
public PageFlowFileFilter()
```

## Method Detail

### accept(File, String) Method

```
public boolean accept(File dir,  
                     String name)
```

# PageFlowRequestProcessor Class

*public class PageFlowRequestProcessor*

extends RequestProcessor  
implements PageFlowConstants, Serializable

The Page Flow extension of the Struts RequestProcessor, which contains callbacks that are invoked during processing of a request to the Struts action servlet. This class is registered as the *controller* for all Struts modules derived from page flows.

---

## Hierarchy

```

Object
  RequestProcessor
    PageFlowRequestProcessor
  
```

## All Implemented Interfaces

PageFlowConstants, Serializable

---

## Nested Class Summary

```

protected class PageFlowRequestProcessor.DefaultHttpRedirector
    public PageFlowRequestProcessor.ExceptionHandledAction
static class      Used by
    PageFlowRequestProcessor.ExceptionHandledAction.processMapping(Http
    HttpServletResponse, String).
    protected PageFlowRequestProcessor.ExceptionHandledActionMapping
static class      Used by
    PageFlowRequestProcessor.ExceptionHandledActionMapping.processMapp
    HttpServletResponse, String).
    protected PageFlowRequestProcessor.GlobalAppActionMapping
static class      Used by
    PageFlowRequestProcessor.GlobalAppActionMapping.processMapping(Http
    HttpServletResponse, String).
    public PageFlowRequestProcessor.GlobalAppForwardAction
static class      Used by
    PageFlowRequestProcessor.GlobalAppForwardAction.processMapping(Http
    HttpServletResponse, String).
  
```

## Field Summary

### Fields from `org.apache.struts.action.RequestProcessor`

`actions`, `appConfig`, `INCLUDE_PATH_INFO`, `INCLUDE_SERVLET_PATH`, `log`, `moduleConfig`, `servlet`

### Fields from interface `com.bea.wlw.netui.pageflow.PageFlowConstants`

`ACTION_EXTENSION`, `ACTION_EXTENSION_LEN`, `ACTION_OVERRIDE`, `BEGIN_ACTION_NAME`, `BEGIN_ACTION_PATH`, `DEFAULT_FORWARD_OVERFLOW_COUNT`, `DEFAULT_NESTING_OVERFLOW_COUNT`, `FORWARD_OVERFLOW_COUNT_PARAM`, `GLOBALAPP_CLASSNAME`, `GLOBALAPP_MEMBER_NAME`, `GLOBALAPP_MODULE_CONTEXT_PATH`, `GLOBALAPP_SOURCEFILE_NAME`, `INTERNAL_VAR_PREFIX`, `ISNESTABLE_VAR_NAME`, `JPF_EXTENSION`, `JPF_MODULE_CONFIG_EXTENSION`, `JPF_MODULE_CONFIG_GEN_DIR`, `JPF_MODULE_CONFIG_PREFIX`, `NESTING_OVERFLOW_COUNT_PARAM`, `PROCESS_ROLES_FAILED_ATTR`, `WEBINF_DIR`

## Constructor Summary

*`PageFlowRequestProcessor()`*

## Method Summary

```
protected void
    changeScheme(String webappRelativeURI, String scheme, int port,
        HttpServletRequest request, HttpServletResponse response)
protected void
    doForward(String uri, HttpServletRequest request,
        HttpServletResponse response)
    protected
    ActionMapping getBeginMapping()
protected final
    int getForwardedRequestCount(HttpServletRequest request)
    public void
        init(ActionServlet servlet, ModuleConfig config)
protected final
    boolean isForwardedRequest(HttpServletRequest request)
    public void
        process(HttpServletRequest request, HttpServletResponse
            response)
```



## Page Flow API Reference

```
protected Action processActionCreate(HttpServletRequest request,
                                       HttpServletResponse response, ActionMapping actionMapping)

    protected
    ActionForm processActionForm(HttpServletRequest request,
                                  HttpServletResponse response, ActionMapping mapping)

    public
    ActionForward processException(HttpServletRequest request,
                                    HttpServletResponse response, Exception ex, ActionForm form,
                                    ActionMapping mapping)

protected void
    processForwardConfig(HttpServletRequest request,
                        HttpServletResponse response, ForwardConfig forward)
    This override of the base method ensures that absolute URIs don't get the
    context path prepended, and handles forwards to special things like
    return-to="currentPage".

    public
    ActionMapping processMapping(HttpServletRequest request,
                                  HttpServletResponse response, String path)

    protected
    HttpServletRequest processMultipart(HttpServletRequest request)
    If this is a multipart request, wrap it with a special wrapper.

protected void
    processNoCache(HttpServletRequest request,
                  HttpServletResponse response)
    Set the no-cache headers for all responses.

protected void
    processPopulate(HttpServletRequest request,
                   HttpServletResponse response, ActionForm form, ActionMapping
                   mapping)

protected boolean
    processRoles(HttpServletRequest request,
                 HttpServletResponse response, ActionMapping mapping)

protected boolean
    shouldDoSecureForwards()

protected void
    superForward(String uri, HttpServletRequest request,
                 HttpServletResponse response)
```

### Methods from `org.apache.struts.action.RequestProcessor`

`destroy`, `doInclude`, `getDebug`, `getInternal`, `getServletContext`,  
`internalModuleRelativeForward`, `internalModuleRelativeInclude`, `log`, `log`,  
`processActionForward`, `processActionPerform`, `processContent`,  
`processForward`, `processInclude`, `processLocale`, `processPath`,  
`processPreprocess`, `processValidate`

### Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

## Constructor Detail

### PageFlowRequestProcessor

```
public PageFlowRequestProcessor()
```

## Method Detail

### changeScheme(String, String, int, HttpServletRequest, HttpServletResponse) Method

```
protected void changeScheme(String webappRelativeURI,
                             String scheme,
                             int port,
                             HttpServletRequest request,
                             HttpServletResponse response)
    throws URISyntaxException, IOException, ServletException
```

#### Exceptions

*URISyntaxException*  
*IOException*  
*ServletException*

---

### doForward(String, HttpServletRequest, HttpServletResponse) Method

```
protected void doForward(String uri,
                          HttpServletRequest request,
                          HttpServletResponse response)
    throws IOException, ServletException
```

#### Overrides

RequestProcessor.doForward(String, HttpServletRequest, HttpServletResponse)

#### Exceptions

*IOException*  
*ServletException*

---

## getBeginMapping() Method

```
protected ActionMapping getBeginMapping()
```

---

## getForwardedRequestCount(HttpServletRequest request) Method

```
protected final int getForwardedRequestCount(HttpServletRequest request)
```

---

## init(ActionServlet, ModuleConfig) Method

```
public void init(ActionServlet servlet,
                 ModuleConfig config)
    throws ServletException
```

### Overrides

```
RequestProcessor.init(ActionServlet, ModuleConfig)
```

### Exceptions

*ServletException*

---

## isForwardedRequest(HttpServletRequest request) Method

```
protected final boolean isForwardedRequest(HttpServletRequest request)
```

---

## process(HttpServletRequest request, HttpServletResponse response) Method

```
public void process(HttpServletRequest request,
                    HttpServletResponse response)
    throws IOException, ServletException
```

### Overrides

```
RequestProcessor.process(HttpServletRequest request, HttpServletResponse response)
```

### Exceptions

*IOException*

*ServletException*

---

## processActionCreate(HttpServletRequest request, HttpServletResponse response, ActionMapping actionMapping) Method

```
protected Action processActionCreate(HttpServletRequest request,
                                    HttpServletResponse response,
                                    ActionMapping actionMapping)
    throws IOException
```

**Overrides**

RequestProcessor.processActionCreate(HttpServletRequest request, HttpServletResponse response, ActionMapping mapping)

**Exceptions**

*IOException*

---

**processActionForm(HttpServletRequest, HttpServletResponse, ActionMapping) Method**

```
protected ActionForm processActionForm(HttpServletRequest request,
                                         HttpServletResponse response,
                                         ActionMapping mapping)
```

**Overrides**

RequestProcessor.processActionForm(HttpServletRequest request, HttpServletResponse response, ActionMapping mapping)

---

**processException(HttpServletRequest, HttpServletResponse, Exception, ActionForm, ActionMapping) Method**

```
public ActionForward processException(HttpServletRequest request,
                                       HttpServletResponse response,
                                       Exception ex,
                                       ActionForm form,
                                       ActionMapping mapping)
    throws IOException, ServletException
```

**Overrides**

RequestProcessor.processException(HttpServletRequest request, HttpServletResponse response, Exception, ActionForm, ActionMapping mapping)

**Exceptions**

*IOException*

*ServletException*

---

**processForwardConfig(HttpServletRequest, HttpServletResponse, ForwardConfig) Method**

```
protected void processForwardConfig(HttpServletRequest request,
                                       HttpServletResponse response,
                                       ForwardConfig forward)
    throws IOException, ServletException
```

This override of the base method ensures that absolute URIs don't get the context path prepended, and handles forwards to special things like return-to="currentPage".

**Overrides**

RequestProcessor.processForwardConfig(HttpServletRequest request, HttpServletResponse response, ForwardConfig)

**Exceptions***IOException**ServletException*

---

**processMapping(HttpServletRequest, HttpServletResponse, String) Method**

```
public ActionMapping processMapping(HttpServletRequest request,
                                   HttpServletResponse response,
                                   String path)
    throws IOException
```

**Overrides**

```
RequestProcessor.processMapping(HttpServletRequest, HttpServletResponse, String)
```

**Exceptions***IOException*

---

**processMultipart(HttpServletRequest) Method**

```
protected HttpServletRequest processMultipart(HttpServletRequest request)
```

If this is a multipart request, wrap it with a special wrapper. Otherwise, return the request unchanged.

**Overrides**

```
RequestProcessor.processMultipart(HttpServletRequest)
```

**Parameters***request*

The HttpServletRequest we are processing

---

**processNoCache(HttpServletRequest, HttpServletResponse) Method**

```
protected void processNoCache(HttpServletRequest request,
                               HttpServletResponse response)
```

Set the no-cache headers for all responses. This overrides the base Struts behavior to prevent caching entirely in iterative dev mode.

**Overrides**

```
RequestProcessor.processNoCache(HttpServletRequest, HttpServletResponse)
```

---

**processPopulate(HttpServletRequest, HttpServletResponse, ActionForm, ActionMapping) Method**

```
protected void processPopulate(HttpServletRequest request,
                                HttpServletResponse response,
                                ActionForm form,
                                ActionMapping mapping)
    throws ServletException
```

**Overrides**

```
RequestProcessor.processPopulate(HttpServletRequest, HttpServletResponse, ActionForm,
ActionMapping)
```

**Exceptions**

*ServletException*

---

**processRoles(HttpServletRequest, HttpServletResponse, ActionMapping) Method**

```
protected boolean processRoles(HttpServletRequest request,
                                HttpServletResponse response,
                                ActionMapping mapping)
    throws IOException, ServletException
```

**Overrides**

```
RequestProcessor.processRoles(HttpServletRequest, HttpServletResponse, ActionMapping)
```

**Exceptions**

*IOException*  
*ServletException*

---

**shouldDoSecureForwards() Method**

```
protected boolean shouldDoSecureForwards()
```

---

**superForward(String, HttpServletRequest, HttpServletResponse) Method**

```
protected void superForward(String uri,
                              HttpServletRequest request,
                              HttpServletResponse response)
    throws IOException, ServletException
```

**Exceptions**

*IOException*  
*ServletException*



# PageFlowRequestProcessor.DefaultHttpRedirector Class

*protected class PageFlowRequestProcessor.DefaultHttpRedirector*

extends Object  
implements HttpRedirector, Serializable

---

## Hierarchy

Object  
PageFlowRequestProcessor.DefaultHttpRedirector

## All Implemented Interfaces

HttpRedirector, Serializable

## Enclosing class

PageFlowRequestProcessor

---

## Constructor Summary

*PageFlowRequestProcessor.DefaultHttpRedirector()*

## Method Summary

```
public  
    void forward(String uri, HttpServletRequest request, HttpServletResponse response,  
                ServletContext servletContext)  
public  
    void redirect(String uri, HttpServletRequest request, HttpServletResponse  
                response, ServletContext servletContext)
```

## Methods from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Methods from interface com.bea.wlw.netui.pageflow.HttpRedirector



`forward, redirect`

## Constructor Detail

### PageFlowRequestProcessor.DefaultHttpRedirector

`protected PageFlowRequestProcessor.DefaultHttpRedirector()`

## Method Detail

### **forward(String, HttpServletRequest, HttpServletResponse, ServletContext)** **Method**

```
public void forward(String uri,  
                    HttpServletRequest request,  
                    HttpServletResponse response,  
                    ServletContext servletContext)  
    throws IOException, ServletException
```

#### Exceptions

*IOException*  
*ServletException*

---

### **redirect(String, HttpServletRequest, HttpServletResponse, ServletContext)** **Method**

```
public void redirect(String uri,  
                    HttpServletRequest request,  
                    HttpServletResponse response,  
                    ServletContext servletContext)  
    throws IOException, ServletException
```

#### Exceptions

*IOException*  
*ServletException*

# PageFlowRequestProcessor.ExceptionHandledAction Class

*public static class PageFlowRequestProcessor.ExceptionHandledAction*

extends Action

Used by

PageFlowRequestProcessor.ExceptionHandledAction.processMapping(HttpServletRequestRequest, HttpServletResponse, String). This action simply returns the ActionForward stored in the ExceptionHandledActionMapping that's passed in.

---

## Hierarchy

Object

Action

PageFlowRequestProcessor.ExceptionHandledAction

## Enclosing class

PageFlowRequestProcessor

---

## Field Summary

### Fields from org.apache.struts.action.Action

ACTION\_SERVLET\_KEY, APPLICATION\_KEY, DATA\_SOURCE\_KEY, defaultLocale, ERROR\_KEY, EXCEPTION\_KEY, FORM\_BEANS\_KEY, FORWARDS\_KEY, LOCALE\_KEY, MAPPING\_KEY, MAPPINGS\_KEY, MESSAGE\_KEY, MESSAGES\_KEY, MULTIPART\_KEY, PLUG\_INS\_KEY, REQUEST\_PROCESSOR\_KEY, servlet, SERVLET\_KEY, TRANSACTION\_TOKEN\_KEY

## Constructor Summary

*PageFlowRequestProcessor.ExceptionHandledAction()*

## Method Summary

```
public
ActionForward execute(ActionMapping mapping, ActionForm form,
                      HttpServletRequest request, HttpServletResponse response)
```

### Methods from `org.apache.struts.action.Action`

`execute`, `generateToken`, `getDataSource`, `getDataSource`, `getLocale`,  
`getResources`, `getResources`, `getResources`, `getServlet`, `isCancelled`,  
`isTokenValid`, `isTokenValid`, `perform`, `perform`, `resetToken`, `saveErrors`,  
`saveMessages`, `saveToken`, `setLocale`, `setServlet`, `toHex`

### Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`,  
`toString`, `wait`, `wait`, `wait`

## Constructor Detail

### `PageFlowRequestProcessor.ExceptionHandledAction`

```
public PageFlowRequestProcessor.ExceptionHandledAction()
```

## Method Detail

### `execute(ActionMapping, ActionForm, HttpServletRequest, HttpServletResponse)` Method

```
public ActionForward execute(ActionMapping mapping,
                             ActionForm form,
                             HttpServletRequest request,
                             HttpServletResponse response)
```

#### *Overrides*

```
Action.execute(ActionMapping, ActionForm, HttpServletRequest, HttpServletResponse)
```

# PageFlowRequestProcessor.ExceptionHandledActionMapping Class

*protected static class* **PageFlowRequestProcessor.ExceptionHandledActionMapping**

extends `ActionMapping`

Used by

`PageFlowRequestProcessor.ExceptionHandledActionMapping.processMapping(HttpServletRequest, HttpServletResponse, String)`. Its main job is to return `String` as the type.

---

## *Hierarchy*

```
Object
├── ActionConfig
│   ├── ActionMapping
│       └── PageFlowRequestProcessor.ExceptionHandledActionMapping
```

## *All Implemented Interfaces*

`Serializable`

## *Enclosing class*

`PageFlowRequestProcessor`

---

## Field Summary

### Fields from `org.apache.struts.config.ActionConfig`

`attribute`, `configured`, `exceptions`, `forward`, `forwards`, `include`, `input`, `moduleConfig`, `multipartClass`, `name`, `parameter`, `path`, `prefix`, `roleNames`, `roles`, `scope`, `suffix`, `type`, `unknown`, `validate`

## Constructor Summary

*PageFlowRequestProcessor.ExceptionHandledActionMapping*(*String* *actionPath*, *ActionForward* *fwd*)

## Method Summary

```

    public
ActionForward getActionForward()
    public
String getType()
    public
boolean getValidate()

```

### Methods from `org.apache.struts.action.ActionMapping`

```

findException, findForward, findForwards, getInputForward

```

### Methods from `org.apache.struts.config.ActionConfig`

```

addExceptionConfig, addForwardConfig, findExceptionConfig,
findExceptionConfigs, findForwardConfig, findForwardConfigs, freeze,
getApplicationConfig, getAttribute, getForward, getInclude, getInput,
getModuleConfig, getMultipartClass, getName, getParameter, getPath,
getPrefix, getRoleNames, getRoles, getScope, getSuffix, getUnknown,
removeExceptionConfig, removeForwardConfig, setApplicationConfig,
setAttribute, setForward, setInclude, setInput, setModuleConfig,
setMultipartClass, setName, setParameter, setPath, setPrefix, setRoles,
setScope, setSuffix, setType, setUnknown, setValidate, toString

```

### Methods from class `java.lang.Object`

```

clone, equals, finalize, getClass, hashCode, notify, notifyAll,
toString, wait, wait, wait

```

## Constructor Detail

### PageFlowRequestProcessor.ExceptionHandledActionMapping

```

public PageFlowRequestProcessor.ExceptionHandledActionMapping(String actionPath,
                                                             ActionForward fwd)

```

## Method Detail

### getActionForward() Method

```

public ActionForward getActionForward()

```

---

## **getType() Method**

```
public String getType()
```

### *Overrides*

```
ActionConfig.getType()
```

---

## **getValidate() Method**

```
public boolean getValidate()
```

### *Overrides*

```
ActionConfig.getValidate()
```

# PageFlowRequestProcessor.GlobalAppActionMapping Class

*protected static class PageFlowRequestProcessor.GlobalAppActionMapping*

extends ActionMapping

Used by

PageFlowRequestProcessor.GlobalAppActionMapping.processMapping(HttpServletRequestRequest, HttpServletResponse, String). Its main job is to return as the type.

---

## *Hierarchy*

Object  
  ActionConfig  
    ActionMapping  
      PageFlowRequestProcessor.GlobalAppActionMapping

## *All Implemented Interfaces*

Serializable

## *Enclosing class*

PageFlowRequestProcessor

---

## Field Summary

### Fields from org.apache.struts.config.ActionConfig

attribute, configured, exceptions, forward, forwards, include, input, moduleConfig, multipartClass, name, parameter, path, prefix, roleNames, roles, scope, suffix, type, unknown, validate

## Constructor Summary

*PageFlowRequestProcessor.GlobalAppActionMapping(String actionPath)*

## Method Summary

```
public
String getType()

public
boolean getValidate()
```

### Methods from `org.apache.struts.action.ActionMapping`

`findException`, `findForward`, `findForwards`, `getInputForward`

### Methods from `org.apache.struts.config.ActionConfig`

`addExceptionConfig`, `addForwardConfig`, `findExceptionConfig`, `findExceptionConfigs`, `findForwardConfig`, `findForwardConfigs`, `freeze`, `getApplicationConfig`, `getAttribute`, `getForward`, `getInclude`, `getInput`, `getModuleConfig`, `getMultipartClass`, `getName`, `getParameter`, `getPath`, `getPrefix`, `getRoleNames`, `getRoles`, `getScope`, `getSuffix`, `getUnknown`, `removeExceptionConfig`, `removeForwardConfig`, `setApplicationConfig`, `setAttribute`, `setForward`, `setInclude`, `setInput`, `setModuleConfig`, `setMultipartClass`, `setName`, `setParameter`, `setPath`, `setPrefix`, `setRoles`, `setScope`, `setSuffix`, `setType`, `setUnknown`, `setValidate`, `toString`

### Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

## Constructor Detail

### PageFlowRequestProcessor.GlobalAppActionMapping

```
public PageFlowRequestProcessor.GlobalAppActionMapping(String actionPath)
```

## Method Detail

### `getType()` Method

```
public String getType()
```

#### *Overrides*

```
ActionConfig.getType()
```

---



## **getValidate() Method**

```
public boolean getValidate()
```

### *Overrides*

```
    ActionConfig.getValidate()
```

# PageFlowRequestProcessor.GlobalAppForwardAction Class

*public static class PageFlowRequestProcessor.GlobalAppForwardAction*

extends Action

Used by

PageFlowRequestProcessor.GlobalAppForwardAction.processMapping(HttpServletRequestRequest, HttpServletResponse, String). This action simply forwards to the stored action-path, but within the Struts module associated with Global.app.

---

## Hierarchy

Object

Action

PageFlowRequestProcessor.GlobalAppForwardAction

## Enclosing class

PageFlowRequestProcessor

---

## Field Summary

### Fields from org.apache.struts.action.Action

ACTION\_SERVLET\_KEY, APPLICATION\_KEY, DATA\_SOURCE\_KEY, defaultLocale, ERROR\_KEY, EXCEPTION\_KEY, FORM\_BEANS\_KEY, FORWARDS\_KEY, LOCALE\_KEY, MAPPING\_KEY, MAPPINGS\_KEY, MESSAGE\_KEY, MESSAGES\_KEY, MULTIPART\_KEY, PLUG\_INS\_KEY, REQUEST\_PROCESSOR\_KEY, servlet, SERVLET\_KEY, TRANSACTION\_TOKEN\_KEY

## Constructor Summary

*PageFlowRequestProcessor.GlobalAppForwardAction()*

## Method Summary

```
public
ActionForward execute(ActionMapping mapping, ActionForm form,
                      HttpServletRequest request, HttpServletResponse response)
```

### Methods from `org.apache.struts.action.Action`

`execute`, `generateToken`, `getDataSource`, `getDataSource`, `getLocale`,  
`getResources`, `getResources`, `getResources`, `getServlet`, `isCancelled`,  
`isTokenValid`, `isTokenValid`, `perform`, `perform`, `resetToken`, `saveErrors`,  
`saveMessages`, `saveToken`, `setLocale`, `setServlet`, `toHex`

### Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`,  
`toString`, `wait`, `wait`, `wait`

## Constructor Detail

### PageFlowRequestProcessor.GlobalAppForwardAction

```
public PageFlowRequestProcessor.GlobalAppForwardAction()
```

## Method Detail

### `execute(ActionMapping, ActionForm, HttpServletRequest, HttpServletResponse)` Method

```
public ActionForward execute(ActionMapping mapping,
                             ActionForm form,
                             HttpServletRequest request,
                             HttpServletResponse response)
```

#### Overrides

```
Action.execute(ActionMapping, ActionForm, HttpServletRequest, HttpServletResponse)
```

# PageFlowStack Class

*public class PageFlowStack*

extends Stack  
implements HttpSessionBindingListener

Stack for keeping track of a series of nested page flows. When a nested page flow is entered, the previous page flow is pushed onto this stack, which is kept in the user session.

---

## ***Hierarchy***

```
Object
  AbstractCollection
    AbstractList
      Vector
        Stack
          PageFlowStack
```

## ***All Implemented Interfaces***

Cloneable, Collection, EventListener, HttpSessionBindingListener, List, RandomAccess, Serializable

---

## **Field Summary**

### **Fields from java.util.Vector**

capacityIncrement, elementCount, elementData

### **Fields from java.util.AbstractList**

modCount

## **Constructor Summary**

*PageFlowStack()*

## Method Summary

```
public
    int search(Class targetClass)
        Returns the 1-based position where a PageFlowController with the same class is on this
        stack.

public
    void valueBound(HttpSessionBindingEvent event)

public
    void valueUnbound(HttpSessionBindingEvent event)
```

### Methods from `java.util.Stack`

`empty`, `peek`, `pop`, `push`, `search`

### Methods from `java.util.Vector`

`add`, `add`, `addAll`, `addAll`, `addElement`, `capacity`, `clear`, `clone`, `contains`, `containsAll`, `copyInto`, `elementAt`, `elements`, `ensureCapacity`, `equals`, `firstElement`, `get`, `hashCode`, `indexOf`, `indexOf`, `insertElementAt`, `isEmpty`, `lastElement`, `lastIndexOf`, `lastIndexOf`, `remove`, `remove`, `removeAll`, `removeAllElements`, `removeElement`, `removeElementAt`, `removeRange`, `retainAll`, `set`, `setElementAt`, `setSize`, `size`, `subList`, `toArray`, `toArray`, `toString`, `trimToSize`

### Methods from `java.util.AbstractList`

`add`, `add`, `addAll`, `clear`, `equals`, `get`, `hashCode`, `indexOf`, `iterator`, `lastIndexOf`, `listIterator`, `listIterator`, `remove`, `removeRange`, `set`, `subList`

### Methods from `java.util.AbstractCollection`

`add`, `addAll`, `clear`, `contains`, `containsAll`, `isEmpty`, `iterator`, `remove`, `removeAll`, `retainAll`, `size`, `toArray`, `toArray`, `toString`

### Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

### Methods from interface `java.util.Collection`

`add`, `addAll`, `clear`, `contains`, `containsAll`, `equals`, `hashCode`, `isEmpty`, `iterator`, `remove`, `removeAll`, `retainAll`, `size`, `toArray`, `toArray`

### Methods from interface `javax.servlet.http.HttpSessionBindingListener`

`valueBound`, `valueUnbound`

### Methods from interface `java.util.List`

`add`, `add`, `addAll`, `addAll`, `clear`, `contains`, `containsAll`, `equals`, `get`, `hashCode`, `indexOf`, `isEmpty`, `iterator`, `lastIndexOf`, `listIterator`, `listIterator`, `remove`, `remove`, `removeAll`, `retainAll`, `set`, `size`, `subList`, `toArray`, `toArray`

## Constructor Detail

### PageFlowStack

```
public PageFlowStack()
```

## Method Detail

### `search(, Class)` Method

```
public int search( ,  
                  Class targetClass)
```

Returns the 1-based position where a `PageFlowController` with the same class is on this stack. This overloads the method `Stack.search(Object o)`

#### Returns

*the 1-based position from the top of the stack where the object is located; the return value -1 indicates that the object is not on the stack.*

---

### `valueBound(HttpSessionBindingEvent)` Method

```
public void valueBound(HttpSessionBindingEvent event)
```

---

### `valueUnbound(HttpSessionBindingEvent)` Method

```
public void valueUnbound(HttpSessionBindingEvent event)
```

# PageFlowUtils Class

*public class PageFlowUtils*

extends Object  
implements PageFlowConstants

str Utility methods related to Page Flow.

---

## **Hierarchy**

Object  
PageFlowUtils

## **All Implemented Interfaces**

PageFlowConstants

---

## **Constructor Summary**

*PageFlowUtils()*

## **Method Summary**

```
public static void
    addPageInput(String name, Object value, ServletRequest request)
        Set a named page input, which corresponds to an input declared by the
        pageInput JSP tag.

public static void
    addValidationError(String propertyName, String messageKey,
        Object[] messageArgs, ServletRequest request)
        Add a validation error that will be shown with the Errors and Error tags.

public static void
    addValidationError(String propertyName, String messageKey,
        Object messageArg, ServletRequest request)
        Add a validation error that will be shown with the Errors and Error tags.

public static void
    addValidationError(String propertyName, String messageKey,
        ServletRequest request)
        Add a validation error that will be shown with the Errors and Error tags.

public static void
```

## Page Flow API Reference

```
destroyPageFlowStack(HttpServletRequest request)
    Destroys the stack of PageFlowControllers that have invoked
    nested page flows.

public static void
dumpRequest(HttpServletRequest request, PrintStream output)
    Print parameters and attributes in the given request.

public static void
dumpServletContext(ServletContext context, PrintStream output)
    Print attributes in the given ServletContext.

    public static
        final ensureCurrentPageFlow(HttpServletRequest request,
PageFlowController HttpServletRequest response, ServletContext context)
            Get or create the current PageFlowController for the given request.

    public static
        GlobalApp ensureGlobalApp(HttpServletRequest request,
HttpServletRequest response)
            Get or create the current GlobalApp instance.

    public static
        GlobalApp ensureGlobalApp(HttpServletRequest request,
HttpServletRequest response, ServletContext servletContext)
            Get or create the current GlobalApp instance.

    public static
        String getActionURI(ServletRequest request)
            Get the most recent action URI that was processed by
            FlowController.execute(ActionMapping, ActionForm,
            HttpServletRequest, HttpServletResponse).

    public static
        String getBeginActionURI(String requestURI)
            Get a URI for the "begin" action in the PageFlowController associated
            with the given request URI.

    public static
        ActionResolver getCurrentActionResolver(HttpServletRequest request)
            Get the current ActionResolver (PageFlowController).

    public static
        final getCurrentPageFlow(HttpServletRequest request)
PageFlowController
            Get the current PageFlowController.

    public static
        String getFileExtension(String filename)
            Get the file extension from a file name.

    public static
        String getFormBeanName(ActionForm formInstance, HttpServletRequest
request)
            Get the name for the type of a ActionForm instance.

    public static
        String getFormBeanName(Class formBeanClass, HttpServletRequest
request)
            Get the name for an ActionForm type.
```



## Page Flow API Reference

```
public static getGlobalApp(HttpServletRequest request)
    GlobalApp          Get the current GlobalApp instance.

public static
    String getJpfClassName(String uri)
        Get the class name of a PageFlowController, given the URI to it.

public static
    String getModuleConfPath(String modulePath)

public static
    String getModulePath(HttpServletRequest request, String requestURI)
        Get the Struts module path for a URI.

public static
    String getModulePath(HttpServletRequest request)
        Get the Struts module path for the current request URI.

public static
    String getModulePathForRelativeURI(String uri)
        Get the Struts module path for a URI that is relative to the web
        application root.

    public static
PageFlowController getNestingPageFlow(HttpServletRequest request)
        Get the PageFlowController that is nesting the current one.

    public static
        final Stack getPageFlowStack(HttpServletRequest request)
            Get the stack of nested pageflows for the current user session.

    public static
        String getPageFlowURI(String className)
            Get the URI for a PageFlowController, given its class name.

    public static
        Object getPageInput(String name, ServletRequest request)
            Get a named page input that was registered in the current request.

    public static
        final String getRelativeURI(HttpServletRequest request,
            PageFlowController relativeTo)
            Get the request URI, relative to the URI of the given PageFlowController.

    public static
        final String getRelativeURI(HttpServletRequest request, String uri,
            PageFlowController relativeTo)
            Get a URI relative to the URI of the given PageFlowController.

    public static
        boolean isAbsoluteURI(String uri)
            Tell whether a given URI is absolute, i.e., whether it contains a
            scheme-part (e.g., "http:").

    public static
        Boolean isSecureResource(String uri, ServletContext context,
            HttpServletRequest request)
            Tell whether a web application resource requires a secure transport
            protocol.

    public static
```

## Page Flow API Reference

Boolean **isSecureResource**(*String uri, ServletContext context*)

public static void  
    **preventCache**(*HttpServletResponse response*)  
        Set response headers to prevent caching of the response by the browser.

public static void  
    **setCurrentActionResolver**(*ActionResolver resolver, HttpServletRequest request*)  
        Set the current ActionResolver (PageFlowController) in the user session.

public static void  
    **setOutputForm**(*ActionMapping mapping, ActionForm form, HttpServletRequest request, boolean overwrite*)  
        Make a form bean available as an attributet in the request/session (as appropriate).

public static void  
    **setOutputForms**(*ActionMapping mapping, Forward fwd, HttpServletRequest request, boolean overwrite*)  
        Make any form beans in the given Forward object available as attributets in the request/session (as appropriate).

public static void  
    **setOutputForms**(*ActionMapping mapping, Forward fwd, HttpServletRequest request*)  
        Make any form beans in the given Forward object available as attributets in the request/session (as appropriate).

public static void  
    **setOutputForms**(*ActionMapping mapping, ActionForm[] outputForms, HttpServletRequest request*)  
        Make a set of form beans available as attributets in the request/session (as appropriate).

public static void  
    **setOutputForms**(*ActionMapping mapping, ActionForm[] outputForms, HttpServletRequest request, boolean overwrite*)  
        Make a set of form beans available as attributets in the request/session (as appropriate).

public static  
    ActionResult **strutsLookup**(*ServletContext context, ServletRequest request, HttpServletResponse response, String actionOverride, String[] autoResolveExtensions*)  
        Resolve the given action to a URI by running an entire request–processing cycle on the given ScopedRequest and ScopedResponse.

### Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

## Constructor Detail

### PageFlowUtils

```
public PageFlowUtils()
```

## Method Detail

### addPageInput(String, Object, ServletRequest) Method

```
public static void addPageInput(String name,
                                Object value,
                                ServletRequest request)
```

Sett a named page input, which corresponds to an input declared by the `pageInput` JSP tag. The actual value can be read from a JSP using the "pageInput " databinding context.

#### Parameters

*name*

the name of hte page input.

*value*

the value of the page input.

*request*

the current ServletRequest.

---

### addValidationError(String, String, Object[], ServletRequest) Method

```
public static void addValidationError(String propertyName,
                                      String messageKey,
                                      Object[] messageArgs,
                                      ServletRequest request)
```

Add a validation error that will be shown with the Errors and Error tags.

#### Parameters

*propertyName*

the name of the property with which to associate this error.

*messageKey*

the message-resources key for the error message.

*messageArgs*

an array of arguments for the error message.

*request*

the current ServletRequest.

---

## **addValidationError(String, String, Object, ServletRequest) Method**

```
public static void addValidationError(String propertyName,
                                     String messageKey,
                                     Object messageArg,
                                     ServletRequest request)
```

Add a validation error that will be shown with the Errors and Error tags.

### **Parameters**

*propertyName*

the name of the property with which to associate this error.

*messageKey*

the message-resources key for the error message.

*messageArg*

an argument for the error message.

*request*

the current ServletRequest.

---

## **addValidationError(String, String, ServletRequest) Method**

```
public static void addValidationError(String propertyName,
                                     String messageKey,
                                     ServletRequest request)
```

Add a validation error that will be shown with the Errors and Error tags.

### **Parameters**

*propertyName*

the name of the property with which to associate this error.

*messageKey*

the message-resources key for the error message.

*request*

the current ServletRequest.

---

**destroyPageFlowStack(HttpServletRequest request) Method**

```
public static void destroyPageFlowStack(HttpServletRequest request)
```

Destroys the stack of PageFlowControllers that have invoked nested page flows.

**Parameters**

*request*

the current HttpServletRequest.

---

**dumpRequest(HttpServletRequest request, PrintStream output) Method**

```
public static void dumpRequest(HttpServletRequest request,
                               PrintStream output)
```

Print parameters and attributes in the given request.

**Parameters**

*request*

the current HttpServletRequest.

*output*

a PrintStream to which to output request parameters and request/session attributes; if null, System.err is used.

---

**dumpServletContext(ServletContext, PrintStream) Method**

```
public static void dumpServletContext(ServletContext context,
                                       PrintStream output)
```

Print attributes in the given ServletContext.

**Parameters**

*context*

the current ServletContext.

*output*

a PrintStream to which to output ServletContext attributes; if null, System.err is used.

---

**ensureCurrentPageFlow(HttpServletRequest request, HttpServletResponse response, ServletContext context) Method**

```
public static final PageFlowController ensureCurrentPageFlow(HttpServletRequest request,
                                                             HttpServletResponse response,
                                                             ServletContext context)
```

Get or create the current PageFlowController for the given request.

### Parameters

*request*

the current HttpServletRequest.

*response*

the current HttpServletResponse.

*context*

the current ServletContext.

### Returns

*the current PageFlowController in the user session, or if there is none, create one appropriate to the given request if possible. Failing that, return null.*

---

## ensureGlobalApp(HttpServletRequest, HttpServletResponse) Method

```
public static GlobalApp ensureGlobalApp(HttpServletRequest request,
                                         HttpServletResponse response)
```

Get or create the current GlobalApp instance.

### Parameters

*request*

the current HttpServletRequest.

*response*

the current HttpServletResponse

### Returns

*the current GlobalApp from the user session, or a newly-instantiated one (based on the user's Global.app file) if none was in the session. Failing that, return null.*

---

## ensureGlobalApp(HttpServletRequest, HttpServletResponse, ServletContext) Method

```
public static GlobalApp ensureGlobalApp(HttpServletRequest request,
                                         HttpServletResponse response,
                                         ServletContext servletContext)
```

Get or create the current GlobalApp instance.

**Parameters***request*the current `HttpServletRequest`.*response*the current `HttpServletResponse`**Returns**

*the current `GlobalApp` from the user session, or a newly-instantiated one (based on the user's `Global.app` file) if none was in the session. Failing that, return `null`.*

---

**getActionURI(ServletRequest) Method**

```
public static String getActionURI(ServletRequest request)
```

Get the most recent action URI that was processed by `FlowController.execute(ActionMapping, ActionForm, HttpServletRequest, HttpServletResponse)`.

**Parameters***request*the current `ServletRequest`.**Returns**

*a String that is the most recent action URI. This is only valid during a request that has been forwarded from the action URI.*

---

**getBeginActionURI(String) Method**

```
public static String getBeginActionURI(String requestURI)
```

Get a URI for the "begin" action in the `PageFlowController` associated with the given request URI.

**Returns**

*a String that is the URI for the "begin" action in the `PageFlowController` associated with the given request URI.*

---

**getCurrentActionResolver(HttpServletRequest) Method**

```
public static ActionResolver getCurrentActionResolver(HttpServletRequest request)
```

Get the current `ActionResolver` (`PageFlowController`).

## Returns

*the current ActionResolver from the user session, or null if there is none.*

---

## getCurrentPageFlow(HttpServletRequest) Method

```
public static final PageFlowController getCurrentPageFlow(HttpServletRequest request)
```

Get the current PageFlowController.

## Parameters

*request*

the current HttpServletRequest.

## Returns

*the current PageFlowController from the user session, or null if there is none.*

---

## getFileExtension(String) Method

```
public static String getFileExtension(String filename)
```

Get the file extension from a file name.

## Parameters

*filename*

the file name.

## Returns

*the file extension (everything after the last '.'), or the empty string if there is no file extension.*

---

## getFormBeanName(ActionForm, HttpServletRequest) Method

```
public static String getFormBeanName(ActionForm formInstance,  
                                     HttpServletRequest request)
```

Get the name for the type of a ActionForm instance. Use a name looked up from the current Struts module, or, if none is found, create one.

## Parameters

*formInstance*

the ActionForm instance whose type will determine the name.

*request*



the current `HttpServletRequest`, which contains a reference to the current Struts module.

## Returns

*the name found in the Struts module, or, if none is found, a name that is either:*

- ◆ *a camel-cased version of the base class name (minus any package or outer-class qualifiers, or, if that name is already taken,*
  - ◆ *the full class name, with '.' and '\$' replaced by '\_'.*
- 

## getFormBeanName(Class, HttpServletRequest) Method

```
public static String getFormBeanName(Class formBeanClass,  
                                     HttpServletRequest request)
```

Get the name for an `ActionForm` type. Use a name looked up from the current Struts module, or, if none is found, create one.

## Parameters

*formBeanClass*

the `ActionForm`-derived class whose type will determine the name.

*request*

the current `HttpServletRequest`, which contains a reference to the current Struts module.

## Returns

*the name found in the Struts module, or, if none is found, a name that is either:*

- ◆ *a camel-cased version of the base class name (minus any package or outer-class qualifiers, or, if that name is already taken,*
  - ◆ *the full class name, with '.' and '\$' replaced by '\_'.*
- 

## getGlobalApp(HttpServletRequest) Method

```
public static GlobalApp getGlobalApp(HttpServletRequest request)
```

Get the current `GlobalApp` instance.

## Parameters

*request*

the current `HttpServletRequest`.

## Returns

*the current `GlobalApp` from the user session, or null if none exists.*

---

## getJpfClassName(String) Method

```
public static String getJpfClassName(String uri)
```

Get the class name of a PageFlowController, given the URI to it.

### Parameters

*uri*

the URI to the PageFlowController, which should be relative to the web application root (i.e., it should not include the context path).

---

## getModuleConfPath(String) Method

**DEPRECATED** Use

`com.bea.wlw.netui.pageflow.PageFlowActionServlet.getModuleConfPath(java.lang.String)` instead. Get the path to the Struts module configuration file (e.g., `"/WEB-INF/.pageflow-struts-generated/jpf-struts-config-someModule"`) for a given module path (e.g., `"someModule"`), according to the PageFlow convention.

```
public static String getModuleConfPath(String modulePath)
```

### Parameters

*modulePath*

the Struts module path.

### Returns

*a String that is the path to the Struts configuration file, relative to the web application root.*

---

## getModulePath(HttpServletRequest, String) Method

```
public static String getModulePath(HttpServletRequest request,
                                   String requestURI)
```

Get the Struts module path for a URI. This is the parent directory, relative to the web application root, of the file referenced by the URI.

### Parameters

*request*

the current HttpServletRequest.

*requestURI*

the URI for which to get the Struts module path.

---

## getModulePath(HttpServletRequest) Method

```
public static String getModulePath(HttpServletRequest request)
```

Get the Struts module path for the current request URI. This is the parent directory, relative to the web application root, of the file referenced by the request URI.

### Parameters

*request*

the current HttpServletRequest.

---

## getModulePathForRelativeURI(String) Method

```
public static String getModulePathForRelativeURI(String uri)
```

Get the Struts module path for a URI that is relative to the web application root.

### Parameters

*uri*

the URI for which to get the module path.

---

## getNestingPageFlow(HttpServletRequest) Method

```
public static PageFlowController getNestingPageFlow(HttpServletRequest request)
```

Get the PageFlowController that is nesting the current one.

### Parameters

*request*

the current HttpServletRequest.

### Returns

*the nesting PageFlowController, or null if the current one is not being nested.*

---

## getPageFlowStack(HttpServletRequest) Method

```
public static final Stack getPageFlowStack(HttpServletRequest request)
```

Get the stack of nested pageflows for the current user session. Create and store an empty stack if none exists.

## Parameters

*request*

the current `HttpServletRequest`

## Returns

*a `PageFlowStack` of nested pageflows for the current user session.*

---

## getPageFlowURI(String) Method

```
public static String getPageFlowURI(String className)
```

Get the URI for a `PageFlowController`, given its class name.

## Parameters

*className*

the name of the `PageFlowController` class.

## Returns

*a String that is the URI for the `PageFlowController`, relative to the web application root (i.e., not including the context path).*

---

## getPageInput(String, ServletRequest) Method

```
public static Object getPageInput(String name,  
                                 ServletRequest request)
```

Get a named page input that was registered in the current request.

## Parameters

*name*

the name of the page input.

*request*

the current `ServletRequest`

## Related Topics

`PageFlowUtils.addPageInput(String, Object, ServletRequest)`

---

## getRelativeURI(HttpServletRequest, PageFlowController) Method

```
public static final String getRelativeURI(HttpServletRequest request,  
                                          PageFlowController relativeTo)
```

Get the request URI, relative to the URI of the given PageFlowController.

### Parameters

*request*

the current HttpServletRequest.

*relativeTo*

a PageFlowController to which the returned URI should be relative, or null if the returned URI should be relative to the webapp root.

---

## getRelativeURI(HttpServletRequest, String, PageFlowController) Method

```
public static final String getRelativeURI(HttpServletRequest request,
                                         String uri,
                                         PageFlowController relativeTo)
```

Get a URI relative to the URI of the given PageFlowController.

### Parameters

*request*

the current HttpServletRequest.

*uri*

the URI which should be made relative.

*relativeTo*

a PageFlowController to which the returned URI should be relative, or null if the returned URI should be relative to the webapp root.

---

## isAbsoluteURI(String) Method

```
public static boolean isAbsoluteURI(String uri)
```

Tell whether a given URI is absolute, i.e., whether it contains a scheme-part (e.g., "http:").

### Parameters

*uri*

the URI to test.

### Returns

*true if the given URI is absolute.*

---

**isSecureResource(String, ServletContext, HttpServletRequest) Method**

```
public static Boolean isSecureResource(String uri,
                                     ServletContext context,
                                     HttpServletRequest request)
```

Tell whether a web application resource requires a secure transport protocol. This is determined from web.xml; for example, the following block specifies that all resources under /login require a secure transport protocol.

```
<security-constraint>
  <web-resource-collection>
    <web-resource-name>Secure PageFlow - begin</web-resource-name>
    <url-pattern>/login/*</url-pattern>
  </web-resource-collection>
  <user-data-constraint>
    <transport-guarantee>CONFIDENTIAL</transport-guarantee>
  </user-data-constraint>
</security-constraint>
```

**Parameters**

*uri*

a webapp-relative URI for a resource. There must not be query parameters or a scheme on the URI.

*context*

the current request.

**Returns**

*Boolean.TRUE* if a transport-guarantee of *CONFIDENTIAL* or *INTEGRAL* is associated with the given resource; *Boolean.FALSE* a transport-guarantee of *NONE* is associated with the given resource; or *null* if there is no transport-guarantee associated with the given resource.

---

**isSecureResource(String, ServletContext) Method**

**DEPRECATED** Use `isSecureResource( String uri, ServletContext context, HttpServletRequest request )` instead. This version will not work in a portable pageflow environment.

```
public static Boolean isSecureResource(String uri,
                                     ServletContext context)
```

**Parameters**

*uri*

a webapp-relative URI for a resource. There must not be query parameters or a scheme on the URI.

*context*

the ServletContext

**Returns**

*Boolean.TRUE* if a transport-guarantee of *CONFIDENTIAL* or *INTEGRAL* is associated with the given resource; *Boolean.FALSE* a transport-guarantee of *NONE* is associated with the given resource; or *null* if there is no transport-guarantee associated with the given resource.

---

**preventCache(HttpServletResponse) Method**

```
public static void preventCache(HttpServletResponse response)
```

Set response headers to prevent caching of the response by the browser.

**Parameters**

*response*  
the current HttpServletResponse

---

**setCurrentActionResolver(ActionResolver, HttpServletRequest) Method**

```
public static void setCurrentActionResolver(ActionResolver resolver,
                                           HttpServletRequest request)
```

Set the current ActionResolver (PageFlowController) in the user session.

**Parameters**

*resolver*  
the ActionResolver to set as the current one in the user session.

*request*  
the current HttpServletRequest.

---

**setOutputForm(ActionMapping, ActionForm, HttpServletRequest, boolean) Method**

```
public static void setOutputForm(ActionMapping mapping,
                                 ActionForm form,
                                 HttpServletRequest request,
                                 boolean overwrite)
```

Make a form bean available as an attribute in the request/session (as appropriate).

**Parameters**

*mapping*  
the ActionMapping for the current Struts action being processed.

*form*

an `ActionForm` instance to be made available in the request/session (as appropriate).

*request*

if `false` a form from `fwd` will only be set in the request if there is no existing form with the same name.

*overwrite*

the current `HttpServletRequest`.

---

### **setOutputForms(ActionMapping, Forward, HttpServletRequest, boolean) Method**

```
public static void setOutputForms(ActionMapping mapping,
                                  Forward fwd,
                                  HttpServletRequest request,
                                  boolean overwrite)
```

Make any form beans in the given `Forward` object available as attributes in the request/session (as appropriate).

#### **Parameters**

*mapping*

the `ActionMapping` for the current Struts action being processed.

*fwd*

the `Forward` object that contains the `ActionForm` instances to be made available in the request/session (as appropriate).

*request*

the current `HttpServletRequest`.

*overwrite*

if `false` a form from `fwd` will only be set in the request if there is no existing form with the same name.

---

### **setOutputForms(ActionMapping, Forward, HttpServletRequest) Method**

```
public static void setOutputForms(ActionMapping mapping,
                                  Forward fwd,
                                  HttpServletRequest request)
```

Make any form beans in the given `Forward` object available as attributes in the request/session (as appropriate).

#### **Parameters**

*mapping*

the `ActionMapping` for the current Struts action being processed.



*fwd*

the Forward object that contains the ActionForm instances to be made available in the request/session (as appropriate).

*request*

the current HttpServletRequest.

## **setOutputForms(ActionMapping, ActionForm[], HttpServletRequest) Method**

```
public static void setOutputForms(ActionMapping mapping,
                                ActionForm[] outputForms,
                                HttpServletRequest request)
```

Make a set of form beans available as attributets in the request/session (as appropriate).

### **Parameters**

*mapping*

the ActionMapping for the current Struts action being processed.

*outputForms*

an array of ActionForm instances to be made available in the request/session (as appropriate).

*request*

the current HttpServletRequest.

## **setOutputForms(ActionMapping, ActionForm[], HttpServletRequest, boolean) Method**

```
public static void setOutputForms(ActionMapping mapping,
                                ActionForm[] outputForms,
                                HttpServletRequest request,
                                boolean overwrite)
```

Make a set of form beans available as attributets in the request/session (as appropriate).

### **Parameters**

*mapping*

the ActionMapping for the current Struts action being processed.

*outputForms*

an array of ActionForm instances to be made available in the request/session (as appropriate).

*request*

if false a form from fwd will only be set in the request if there is no existing form with the same name.

*overwrite*

the current `HttpServletRequest`.

---

## **strutsLookup(ServletContext, ServletRequest, HttpServletResponse, String, String[]) Method**

```
public static ActionResult strutsLookup(ServletContext context,  
                                       ServletRequest request,  
                                       HttpServletResponse response,  
                                       String actionOverride,  
                                       String[] autoResolveExtensions)  
  
    throws Exception
```

Resolve the given action to a URI by running an entire request–processing cycle on the given `ScopedRequest` and `ScopedResponse`.

### **Parameters**

*context*

the current `ServletContext`

*request*

the `ServletRequest`, which must be a `ScopedRequest`.

*response*

the `ServletResponse`, which must be a `ScopedResponse`.

*actionOverride*

if not `null`, this action–name is used to construct an action URI which is set as the request URI.

*autoResolveExtensions*

a list of URI extensions (e.g., ".do", ".jpf") that will be auto–resolved, i.e., on which this method will be recursively called. If `null`, the default extensions ".do" and ".jpf" will be used.

### **Exceptions**

*Exception*

# PreviousActionInfo Class

*public class PreviousActionInfo*

extends Object  
implements Serializable

Stores information about a recent action execution within a pageflow — used with Used with `return-to="previousAction"` on `@jpf:forward`.

---

## *Hierarchy*

Object  
PreviousActionInfo

## *All Implemented Interfaces*

Serializable

---

## Constructor Summary

*PreviousActionInfo(ActionForm form, String actionURI)*

Constructor which accepts an ActionForm and action URI.

## Method Summary

```
public
String getActionURI()
    Get the URI that was used to execute the
    action.

public
ActionForm getForm()
    Get the form that was passed to the action.

public
void setActionURI(String actionURI)
    Set the URI that was used to execute the action.

public
void setForm(ActionForm form)
    Set the form that was passed to the action.
```

### Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

## Constructor Detail

### PreviousActionInfo

```
public PreviousActionInfo(ActionForm form,  
                          String actionURI)
```

Constructor which accepts an ActionForm and action URI.

## Method Detail

### getActionURI() Method

```
public String getActionURI()
```

Get the URI that was used to execute the action.

#### Returns

*the String URI that was used to execute the action.*

---

### getForm() Method

```
public ActionForm getForm()
```

Get the form that was passed to the action.

#### Returns

*the form that was passed to the action.*

---

### setActionURI(String) Method

```
public void setActionURI(String actionURI)
```

Set the URI that was used to execute the action.

## Parameters

*actionURI*

the URI that was used to execute the action.

---

## setForm(ActionForm) Method

```
public void setForm(ActionForm form)
```

Set the form that was passed to the action.

## Parameters

*form*

the form that was passed to the action.

# PreviousPageInfo Class

*public class PreviousPageInfo*

extends Object  
implements Serializable

Stores information about a previously–displayed page, as well as its initialization data. Used with `return-to="currentPage"` or `return-to="previousPage"` on `@jpf:forward`.

---

## **Hierarchy**

Object  
PreviousPageInfo

## **All Implemented Interfaces**

Serializable

---

## **Constructor Summary**

*PreviousPageInfo*(*ActionForward forward*, *ActionForm form*, *ActionMapping mapping*)

Constructor which accepts the *ActionForward* used to display the page, the *ActionForm* used to initialize the page, and the associated *ActionMapping*, which represents the action that forwarded to the page.

## **Method Summary**

public  
*ActionForm* **getForm**()  
Get the form bean that was used to initialize the page.

public  
*ActionForward* **getForward**()  
Get the object that was used to forward to the page.

public  
*ActionMapping* **getMapping**()  
Get information about the action that forwarded to the page.

public void  
**setForm**(*ActionForm form*)  
Set the form bean that was used to initialize the page.

```
public void setForward(ActionForward forward)
    Set the object that was used to forward to the page.

public void
    setMapping(ActionMapping mapping)
    Set information about the action that forwarded to the page.
```

### Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

## Constructor Detail

### PreviousPageInfo

```
public PreviousPageInfo(ActionForward forward,
    ActionForm form,
    ActionMapping mapping)
```

Constructor which accepts the ActionForward used to display the page, the ActionForm used to initialize the page, and the associated ActionMapping, which represents the action that forwarded to the page.

## Method Detail

### getForm() Method

```
public ActionForm getForm()
```

Get the form bean that was used to initialize the page.

#### Returns

*the ActionForm instance that was used to initialize the page, or null if there was none.*

---

### getForward() Method

```
public ActionForward getForward()
```

Get the object that was used to forward to the page.

#### Returns

*the ActionForward returned by the action that forwarded to this page.*

---

## getMapping() Method

```
public ActionMapping getMapping()
```

Get information about the action that forwarded to the page.

### Returns

*an ActionMapping that contains information about the action that forwarded to this page.*

---

## setForm(ActionForm) Method

```
public void setForm(ActionForm form)
```

Set the form bean that was used to initialize the page.

### Parameters

*form*

the ActionForm instance that was used to initialize the page.

---

## setForward(ActionForward) Method

```
public void setForward(ActionForward forward)
```

Set the object that was used to forward to the page.

### Parameters

*forward*

the ActionForward returned by the action that forwarded to this page.

---

## setMapping(ActionMapping) Method

```
public void setMapping(ActionMapping mapping)
```

Set information about the action that forwarded to the page.

### Parameters

*mapping*

an ActionMapping that contains information about the action that forwarded to this page.



# ProcessPopulate.ExpressionUpdateNode Class

*public static final class **ProcessPopulate.ExpressionUpdateNode***

extends Object

An inner class that represents the data that will be used to perform an update. If a key has a prefix handler, this node is constructed and passed to the prefix handler so that the prefix handler can change the expression or values that will be used to execute the expression update.

---

## Hierarchy

Object

ProcessPopulate.ExpressionUpdateNode

---

## Field Summary

```
public String
    expression
    String
public String
    values
    String
```

## Method Summary

```
public
String toString()
```

## Methods from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

## Field Detail

### expression

```
public String expression
```

---

## values

```
public String values
```

## Method Detail

### toString() Method

```
public String toString()
```

#### *Overrides*

```
Object.toString()
```

# RequestParameterHandler Interface

*public interface RequestParameterHandler*

---

*All Known Implementing Classes*

---

## Method

### Summary

```
public
    void process(HttpServletRequest request, String key,
        ProcessPopulate.ExpressionUpdateNode node)
```

### Method Detail

**process(HttpServletRequest, String, ProcessPopulate.ExpressionUpdateNode) Method**

```
public void process(HttpServletRequest request,
    String key,
    ProcessPopulate.ExpressionUpdateNode node)
```

# UnfulfilledRolesException Class

*public class UnfulfilledRolesException*

extends PageFlowException

Exception that is thrown when an action method marked with `@jpf:action roles-allowed="roles"` is hit when there is a logged-in user who does not fulfil any of the given roles.

---

## Hierarchy

```
Object
  Throwable
    Exception
      RuntimeException
        PageFlowException
          UnfulfilledRolesException
```

## All Implemented Interfaces

```
Serializable
```

---

## Constructor Summary

*UnfulfilledRolesException*(String roleNames, String rolesList, String actionName, FlowController fc)

Construct on the list of roles that were allowed access to the action.

## Method Summary

```
protected
Object[] getMessageArgs()
    public
String[] getMessageParts()
    public
String[] getRoleNames()
```

Get the names of the roles that were allowed access to the action.

## Methods from com.bea.wlw.netui.pageflow.PageFlowException

```
getActionName, getFlowController, getFlowControllerURI,
getLocalizedMessage, getMessage, printError, sendError
```

### Methods from `java.lang.Throwable`

`fillInStackTrace`, `getCause`, `getLocalizedMessage`, `getMessage`,  
`getStackTrace`, `initCause`, `printStackTrace`, `printStackTrace`,  
`printStackTrace`, `setStackTrace`, `toString`

### Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`,  
`toString`, `wait`, `wait`, `wait`

## Constructor Detail

### UnfulfilledRolesException

```
public UnfulfilledRolesException(String[] roleNames,  
                                String rolesList,  
                                String actionName,  
                                FlowController fc)
```

Construct on the list of roles that were allowed access to the action.

## Method Detail

### `getMessageArgs()` Method

```
protected Object[] getMessageArgs()
```

#### *Overrides*

```
PageFlowException.getMessageArgs()
```

---

### `getMessageParts()` Method

```
public String[] getMessageParts()
```

#### *Overrides*

```
PageFlowException.getMessageParts()
```

---

### `getRoleNames()` Method

```
public String[] getRoleNames()
```

Get the names of the roles that were allowed access to the action.

## Returns

*an array of String role names.*

# UnresolvableForwardException Class

*public class UnresolvableForwardException*

extends PageFlowException

Exception thrown by Forward when a forward name does not resolve to a forward defined by @jpf:forward.

---

## *Hierarchy*

```
Object
  Throwable
    Exception
      RuntimeException
        PageFlowException
          UnresolvableForwardException
```

## *All Implemented Interfaces*

Serializable

---

## Constructor Summary

*UnresolvableForwardException(String forwardName, String actionName, FlowController fc)*

Constructor.

## Method Summary

```
public
String getForwardName()
    Get the name of the unresolvable Forward.

protected
Object[] getMessageArgs()
    public
String[] getMessageParts()
    public
void setForwardName(String forwardName)
    Set the name of the unresolvable Forward.
```

Methods from `com.bea.wlw.netui.pageflow.PageFlowException`

`getActionName, getFlowController, getFlowControllerURI,  
getLocalizedMessage, getMessage, printError, sendError`

#### Methods from `java.lang.Throwable`

`fillInStackTrace, getCause, getLocalizedMessage, getMessage,  
getStackTrace, initCause, printStackTrace, printStackTrace,  
printStackTrace, setStackTrace, toString`

#### Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll,  
toString, wait, wait, wait`

## Constructor Detail

### UnresolvableForwardException

```
public UnresolvableForwardException(String forwardName,  
                                   String actionName,  
                                   FlowController fc)
```

Constructor.

## Method Detail

### getForwardName() Method

```
public String getForwardName()
```

Get the name of the unresolvable Forward.

#### Returns

*a String that is the name of the unresolvable Forward.*

---

### getMessageArgs() Method

```
protected Object[] getMessageArgs()
```

#### Overrides

`PageFlowException.getMessageArgs()`

---



## **getMessageParts() Method**

```
public String[] getMessageParts()
```

### *Overrides*

```
PageFlowException.getMessageParts()
```

---

## **setForwardName(String) Method**

```
public void setForwardName(String forwardName)
```

Set the name of the unresolvable Forward.

### **Parameters**

*forwardName*

a String that is the name of the unresolvable Forward.

# com.bea.wlw.netui.pageflow.scoping Package

## Interface Summary

ScopedRequest      A wrapper around HttpServletRequest, associated with a given scope-key.  
ScopedResponse      A wrapper around HttpServletResponse, associated with a given scope-key.  
ScopedServletContext      A wrapper around ServletContext, associated with a given scope-key.

## Class Summary

ScopedServletUtils      Utilities for creating scoped wrapper versions of HttpRequest, HttpResponse, ServletContext.

# ScopedRequest Interface

*public interface ScopedRequest*

extends `HttpServletRequest`

A wrapper around `HttpServletRequest`, associated with a given scope-key. All calls to `setAttribute`, `getAttribute`, `removeAttribute`, etc. are scoped to this object, while most other functionality delegates to the wrapped `HttpServletRequest`. Instances of this class also keep track of their own request-URIs, which are independent of the wrapped request-URIs.

---

## All Superinterfaces

`HttpServletRequest`, `ServletRequest`

---

## Field Summary

```
public
static AUTOSCOPE_PREFIX
final String      String
```

## Fields from interface `javax.servlet.http.HttpServletRequest`

`BASIC_AUTH`, `CLIENT_CERT_AUTH`, `DIGEST_AUTH`, `FORM_AUTH`

## Method Summary

```
public void
    addListenScope(Object scopeKey)
        Adds a scope to "listen" to.

public boolean
    didRedirect()

public void
    doForward()

public String
    getForwardedURI()

public String
    getListenScopeParameter(String attrName)

public String
    getLocalParameter(String attrName)

public
HttpServletRequest getOuterRequest()

public String
```

```

        getScopedName(String baseName)
public Object
        getScopeKey()
public boolean
        hasListenScopes()
public void
        persistAttributes()
        Stores the current map of request attributes in the Session.
public void
        registerNonPersistedAttribute(String attrName)
public void
        registerOuterAttribute(String attrName)
public void
        renameScope(Object newScopeKey)
public void
        restoreAttributes()
        Restores the map of request attributes from a map saved in the
        Session.
public void
        setActiveRequest()
        Makes this request listen to specially-prefixed request parameters.
public void
        setRequestURI(String uri)

```

#### Methods from interface `javax.servlet.http.HttpServletRequest`

```

getAuthType, getContextPath, getCookies, getDateHeader, getHeader,
getHeaderNames, getHeaders, getIntHeader, getMethod, getPathInfo,
getPathTranslated, getQueryString, getRemoteUser, getRequestedSessionId,
getRequestURI, getRequestURL, getServletPath, getSession, getSession,
getUserPrincipal, isRequestedSessionIdFromCookie,
isRequestedSessionIdFromUrl, isRequestedSessionIdFromURL,
isRequestedSessionIdValid, isUserInRole

```

#### Methods from interface `javax.servlet.ServletRequest`

```

getAttribute, getAttributeNames, getCharacterEncoding, getContentLength,
getContentType, getInputStream, getLocale, getLocales, getParameter,
getParameterMap, getParameterNames, getParameterValues, getProtocol,
getReader, getRealPath, getRemoteAddr, getRemoteHost,
getRequestDispatcher, getScheme, getServerName, getServerPort, isSecure,
removeAttribute, setAttribute, setCharacterEncoding

```

## Field Detail

## AUTOSCOPE\_PREFIX

```
public static final String AUTOSCOPE_PREFIX
```

## Method Detail

### addListenScope(Object) Method

```
public void addListenScope(Object scopeKey)
```

Adds a scope to "listen" to. This scope will see all request parameters from a ScopedRequest of the given scope.

---

### didRedirect() Method

*DEPRECATED* Use

`com.bea.wlw.netui.pageflow.scoping.ScopedResponse.didRedirect()` instead.

```
public boolean didRedirect()
```

---

### doForward() Method

```
public void doForward()
```

---

### getForwardedURI() Method

```
public String getForwardedURI()
```

---

### getListenScopeParameter(String) Method

```
public String getListenScopeParameter(String attrName)
```

---

### getLocalParameter(String) Method

```
public String getLocalParameter(String attrName)
```

---

### getOuterRequest() Method

```
public HttpServletRequest getOuterRequest()
```

---

## getScopedName(String) Method

```
public String getScopedName(String baseName)
```

---

## getScopeKey() Method

```
public Object getScopeKey()
```

---

## hasListenScopes() Method

```
public boolean hasListenScopes()
```

---

## persistAttributes() Method

```
public void persistAttributes()
```

Stores the current map of request attributes in the Session.

---

## registerNonPersistedAttribute(String) Method

```
public void registerNonPersistedAttribute(String attrName)
```

---

## registerOuterAttribute(String) Method

```
public void registerOuterAttribute(String attrName)
```

---

## renameScope(Object) Method

```
public void renameScope(Object newScopeKey)
```

---

## restoreAttributes() Method

```
public void restoreAttributes()
```

Restores the map of request attributes from a map saved in the Session.

---

## setActiveRequest() Method

```
public void setActiveRequest()
```

Makes this request listen to specially–prefixed request parameters.

---

## **setRequestURI(String) Method**

```
public void setRequestURI(String uri)
```

# ScopedResponse Interface

*public interface ScopedResponse*

extends `HttpServletResponse`

A wrapper around `HttpServletResponse`, associated with a given scope-key. Delegates to the wrapped response object for some functionality, but prevents output or error codes or forwards from actually happening.

---

## All Superinterfaces

`HttpServletResponse`, `ServletResponse`

---

## Field Summary

### Fields from interface `javax.servlet.http.HttpServletResponse`

`SC_ACCEPTED`, `SC_BAD_GATEWAY`, `SC_BAD_REQUEST`, `SC_CONFLICT`, `SC_CONTINUE`,  
`SC_CREATED`, `SC_EXPECTATION_FAILED`, `SC_FORBIDDEN`, `SC_GATEWAY_TIMEOUT`,  
`SC_GONE`, `SC_HTTP_VERSION_NOT_SUPPORTED`, `SC_INTERNAL_SERVER_ERROR`,  
`SC_LENGTH_REQUIRED`, `SC_METHOD_NOT_ALLOWED`, `SC_MOVED_PERMANENTLY`,  
`SC_MOVED_TEMPORARILY`, `SC_MULTIPLE_CHOICES`, `SC_NO_CONTENT`,  
`SC_NON_AUTHORITATIVE_INFORMATION`, `SC_NOT_ACCEPTABLE`, `SC_NOT_FOUND`,  
`SC_NOT_IMPLEMENTED`, `SC_NOT_MODIFIED`, `SC_OK`, `SC_PARTIAL_CONTENT`,  
`SC_PAYMENT_REQUIRED`, `SC_PRECONDITION_FAILED`,  
`SC_PROXY_AUTHENTICATION_REQUIRED`, `SC_REQUEST_ENTITY_TOO_LARGE`,  
`SC_REQUEST_TIMEOUT`, `SC_REQUEST_URI_TOO_LONG`,  
`SC_REQUESTED_RANGE_NOT_SATISFIABLE`, `SC_RESET_CONTENT`, `SC_SEE_OTHER`,  
`SC_SERVICE_UNAVAILABLE`, `SC_SWITCHING_PROTOCOLS`, `SC_TEMPORARY_REDIRECT`,  
`SC_UNAUTHORIZED`, `SC_UNSUPPORTED_MEDIA_TYPE`, `SC_USE_PROXY`

## Method Summary

```
public void
    applyRedirect()
        Actually send the redirect that was suggested by
        ScopedResponse.sendRedirect(String).

public boolean
    didRedirect()
        Tell whether a browser redirect was sent.

public Cookie
    getCookie(String cookieName)
```



## Page Flow API Reference

Get a cookie that was added to the response.

```
public Cookie[]  
    getCookies()  
        Get all Cookies that were added to the response.  
public Object  
    getFirstHeader(String name)  
        Get the first header with the given name.  
public Map  
    getHeaders()  
        Get all headers.  
public List  
    getHeaders(String name)  
        Get all headers with the given name.  
public  
HttpServletResponse getOuterResponse()  
public String  
    getRedirectURI()  
        Get the redirect URI.  
public int  
    getStatusCode()  
        Get the status code on the response.  
public String  
    getStatusMessage()  
        Get the status message on the response.  
public boolean  
    isError()  
        Tell whether the response is in error.
```

### Methods from interface `javax.servlet.http.HttpServletResponse`

`addCookie`, `addDateHeader`, `addHeader`, `addIntHeader`, `containsHeader`,  
`encodeRedirectUrl`, `encodeRedirectURL`, `encodeUrl`, `encodeURL`, `sendError`,  
`sendError`, `sendRedirect`, `setDateHeader`, `setHeader`, `setIntHeader`,  
`setStatus`, `setStatus`

### Methods from interface `javax.servlet.ServletResponse`

`flushBuffer`, `getBufferSize`, `getCharacterEncoding`, `getLocale`,  
`getOutputStream`, `getWriter`, `isCommitted`, `reset`, `resetBuffer`,  
`setBufferSize`, `setContentLength`, `.setContentType`, `setLocale`

## Method Detail

### `applyRedirect()` Method

```
public void applyRedirect()  
    throws IOException
```

Actually send the redirect that was suggested by `ScopedResponse.sendRedirect(String)`.

### Exceptions

*IOException*

if `HttpServletResponse.sendRedirect(String)` causes an `IOException`.

---

### didRedirect() Method

```
public boolean didRedirect()
```

Tell whether a browser redirect was sent.

### Returns

*true* if `ScopedResponse.sendRedirect(String)` was called.

---

### getCookie(String) Method

```
public Cookie getCookie(String cookieName)
```

Get a cookie that was added to the response.

---

### getCookies() Method

```
public Cookie[] getCookies()
```

Get all Cookies that were added to the response.

---

### getFirstHeader(String) Method

```
public Object getFirstHeader(String name)
```

Get the first header with the given name.

### Returns

*an Object (String, Integer, Date) that is the first header with the given name, or null if none is found.*

---

### getHeaders() Method

```
public Map getHeaders()
```

Get all headers.

## Returns

*a Map of header-name (String) -> headers (List).*

---

## getHeaders(String) Method

```
public List getHeaders(String name)
```

Get all headers with the given name.

## Returns

*a List of headers (String, Integer, Date), or null if none are found.*

---

## getOuterResponse() Method

```
public HttpServletResponse getOuterResponse()
```

---

## getRedirectURI() Method

```
public String getRedirectURI()
```

Get the redirect URI.

## Returns

*the URI passed to `ScopedResponse.sendRedirect(String)`, or null if there was no redirect.*

---

## getStatusCode() Method

```
public int getStatusCode()
```

Get the status code on the response.

## Returns

*the status code, set by `ScopedResponse.setStatus(int)`, `ScopedResponse.sendError(int, String)`, or `ScopedResponse.sendError(int)`; -1 if no status was set explicitly.*

---

## getStatusMessage() Method

```
public String getStatusMessage()
```

Get the status message on the response.

## Returns

*the status code, set by `ScopedResponse.sendError(int, String)`, or null if none was set.*

---

## isError() Method

```
public boolean isError()
```

Tell whether the response is in error.

## Returns

*true if `ScopedResponse.sendError(int, String)` or `ScopedResponse.sendError(int)` was called.*

# ScopedServletContext Interface

*public interface ScopedServletContext*

extends ServletContext

A wrapper around ServletContext, associated with a given scope-key. All calls to setAttribute, getAttribute, removeAttribute, etc. are scoped to this object, while most other functionality delegates to the wrapped ServletContext.

---

## *All Superinterfaces*

ServletContext

---

## Method

### Summary

```
    public
ServletContext getRealContext()
    public Object
                getScopeKey()
```

### Methods from interface javax.servlet.ServletContext

```
getAttribute, getAttributeNames, getContext, getInitParameter,
getInitParameterNames, getMajorVersion, getMimeType, getMinorVersion,
getNamedDispatcher, getRealPath, getRequestDispatcher, getResource,
getResourceAsStream, getResourcePaths, getServerInfo, getServlet,
getServletContextName, getServletNames, getServlets, log, log, log,
removeAttribute, setAttribute
```

## Method Detail

### getRealContext() Method

```
public ServletContext getRealContext()
```

---

### getScopeKey() Method

```
public Object getScopeKey()
```

# ScopedServletUtils Class

*public class ScopedServletUtils*

extends Object

Utilities for creating scoped wrapper versions of HttpRequest, HttpResponse, ServletContext. These wrappers are the basis for a scoped servlet environment, which can be used to scope the Struts framework.

---

## Hierarchy

Object  
ScopedServletUtils

---

## Field Summary

```
public
static SCOPE_ID_PARAM
final String String
```

## Constructor Summary

*ScopedServletUtils()*

## Method Summary

```
public static
HttpServletRequest getOuterRequest(HttpServletRequest request)
    Get the outer (unwrapped) request.

public static final
String getRelativeURI(HttpServletRequest request)
    Get the request URI, relative to the webapp root.

public static final
String getRelativeURI(HttpServletRequest request, String uri)
    Get a URI relative to the webapp root.

public static String
getScopedName(String baseName, Object scopeKey)
    Get a scoped version of a given name.

public static
```

## Page Flow API Reference

ScopedRequest **getScopedRequest**(*HttpServletRequest realRequest, String overrideURI, ServletContext servletContext, Object scopeKey*)  
Get the cached wrapper servlet request.

public static  
ScopedResponse **getScopedResponse**(*HttpServletResponse realResponse, ScopedRequest scopedRequest*)  
Get the cached wrapper servlet response.

public static  
ScopedServletContext **getScopedServletContext**(*HttpServletRequest realRequest, ServletContext realServletContext, Object scopeKey*)  
Get the cached wrapper servlet context.

public static Object  
**getScopedSessionAttr**(*String attrName, HttpServletRequest request*)  
If the request is a ScopedRequest, this returns an attribute whose name is scoped to that request's scope-ID; otherwise, it is a straight passthrough to `HttpSession.getAttribute(String)`.

public static String  
**getScopedSessionAttrName**(*String attrName, HttpServletRequest request*)  
If the request is a ScopedRequest, this returns an attribute name scoped to that request's scope-ID; otherwise, it returns the given attribute name.

public static void  
**renameScope**(*Object oldScopeKey, Object newScopeKey, HttpServletRequest request*)  
Find all scoped objects (ScopedRequest, ScopedResponse, ScopedServletContext) which have a certain scope-key, replaces this scope-key with the new one, and re-caches the objects the new scope-key.

public static  
ScopedRequest **unwrapRequest**(*ServletRequest request*)  
Unwraps the contained ScopedRequest from the given ServletRequest, which may be a ServletRequestWrapper.

public static  
ScopedResponse **unwrapResponse**(*ServletResponse response*)  
Unwraps the contained ScopedResponseImpl from the given ServletResponse, which may be a ServletResponseWrapper.

### Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

## Field Detail

## SCOPE\_ID\_PARAM

```
public static final String SCOPE_ID_PARAM
```

## Constructor Detail

### ScopedServletUtils

```
public ScopedServletUtils()
```

## Method Detail

### getOuterRequest(HttpServletRequest) Method

```
public static HttpServletRequest getOuterRequest(HttpServletRequest request)
```

Get the outer (unwrapped) request.

#### Parameters

*request*  
the request to unwrap.

#### Returns

*the outer request, if the given request is a ScopedRequest (or wraps a ScopedRequest); otherwise, the given request itself.*

---

### getRelativeURI(HttpServletRequest) Method

```
public static final String getRelativeURI(HttpServletRequest request)
```

Get the request URI, relative to the webapp root.

#### Parameters

*request*  
the current HttpServletRequest.

---

### getRelativeURI(HttpServletRequest, String) Method

```
public static final String getRelativeURI(HttpServletRequest request,  
                                           String uri)
```



Get a URI relative to the webapp root.

### Parameters

*request*

the current `HttpServletRequest`.

*uri*

the URI which should be made relative.

---

## getScopedName(String, Object) Method

```
public static String getScopedName(String baseName,
                                   Object scopeKey)
```

Get a scoped version of a given name.

### Parameters

*baseName*

the name to be scoped.

*scopeKey*

the context key for scoping the name.

### Returns

*a scoped version of the given name.*

---

## getScopedRequest(HttpServletRequest, String, ServletContext, Object) Method

```
public static ScopedRequest getScopedRequest(HttpServletRequest realRequest,
                                              String overrideURI,
                                              ServletContext servletContext,
                                              Object scopeKey)
```

Get the cached wrapper servlet request. If none exists, creates one and caches it.

### Parameters

*realRequest*

the "real" (outer) `HttpServletRequest`, which will be wrapped.

*overrideURI*

the request-URI for the wrapped object. This is a *webapp-relative* URI, i.e., it does not include the context path.

*servletContext*

the current ServletContext. May be a ScopedServletContext.

## Returns

*the cached (or newly-created) ScopedRequest.*

---

## getScopedResponse(HttpServletResponse, ScopedRequest) Method

```
public static ScopedResponse getScopedResponse(HttpServletResponse realResponse,
                                              ScopedRequest scopedRequest)
```

Get the cached wrapper servlet response. If none exists, creates one and caches it.

## Parameters

*realResponse*

the "real" (outer) ServletResponse, which will be wrapped.

*scopedRequest*

the ScopedRequest returned from  
 ScopedServletUtils.getScopedRequest(HttpServletRequest, String,  
 ServletContext, Object).

## Returns

*the cached (or newly-created) ScopedResponse.*

---

## getScopedServletContext(HttpServletRequest, ServletContext, Object) Method

```
public static ScopedServletContext getScopedServletContext(HttpServletRequest realRequest,
                                                         ServletContext realServletContext,
                                                         Object scopeKey)
```

Get the cached wrapper servlet context. If none exists, creates one and caches it.

## Parameters

*realRequest*

the "real" (outer) request object.

*realServletContext*

the "real" (outer) ServletContext, which will be wrapped.

*scopeKey*

the context-key with which to scope the wrapper objects.

**Returns**

*the cached (or newly-created) ScopedContext.*

---

**getScopedSessionAttr(String, HttpServletRequest) Method**

```
public static Object getScopedSessionAttr(String attrName,
                                         HttpServletRequest request)
```

If the request is a ScopedRequest, this returns an attribute whose name is scoped to that request's scope-ID; otherwise, it is a straight passthrough to HttpSession.getAttribute(String).

---

**getScopedSessionAttrName(String, HttpServletRequest) Method**

```
public static String getScopedSessionAttrName(String attrName,
                                              HttpServletRequest request)
```

If the request is a ScopedRequest, this returns an attribute name scoped to that request's scope-ID; otherwise, it returns the given attribute name.

---

**renameScope(Object, Object, HttpServletRequest) Method**

```
public static void renameScope(Object oldScopeKey,
                              Object newScopeKey,
                              HttpServletRequest request)
```

Find all scoped objects (ScopedRequest, ScopedResponse, ScopedServletContext) which have a certain scope-key, replaces this scope-key with the new one, and re-caches the objects the new scope-key.

---

**unwrapRequest(ServletRequest) Method**

```
public static ScopedRequest unwrapRequest(ServletRequest request)
```

Unwraps the contained ScopedRequest from the given ServletRequest, which may be a ServletRequestWrapper.

**Parameters**

*request*

the ScopedRequest, or a wrapper (ServletRequestWrapper) around it.

**Returns**

*the unwrapped ScopedRequest.*

---

## **unwrapResponse(ServletResponse) Method**

```
public static ScopedResponse unwrapResponse(ServletResponse response)
```

Unwraps the contained ScopedResponseImpl from the given ServletResponse, which may be a ServletResponseWrapper.

### **Parameters**

*response*

the ScopedResponse, or a wrapper (ServletResponseWrapper) around it.

### **Returns**

*the unwrapped ScopedResponseImpl.*

# com.bea.wlw.netui.pageflow.util Package

## Class Summary

DefaultURLRewriter

TemplateHelper      Class for creating URLs from parameterized templates.

URLRewriter      URL rewriter interface.

URLRewriterService      Methods for registering URL rewriters, and for rewriting URLs using registered rewriters.

# DefaultURLRewriter Class

*public class* **DefaultURLRewriter**

extends URLRewriter

---

## Hierarchy

Object  
URLRewriter  
DefaultURLRewriter

---

## Field Summary

Fields from **com.bea.wlw.netui.pageflow.util.URLRewriter**

ACTION\_SECURE, ACTION\_UNSECURE, RESOURCE\_SECURE, RESOURCE\_UNSECURE

## Constructor Summary

*DefaultURLRewriter()*

## Method Summary

public  
String **rewriteName**(ServletContext servletContext, ServletRequest request, String  
name)

public  
String **rewriteURL**(ServletContext servletContext, ServletRequest request,  
ServletResponse response, String url, String type)

Methods from **com.bea.wlw.netui.pageflow.util.URLRewriter**

getNextRewriter, setNextRewriter

Methods from class **java.lang.Object**

clone, equals, finalize, getClass, hashCode, notify, notifyAll,  
toString, wait, wait, wait

## Constructor Detail

### DefaultURLRewriter

```
public DefaultURLRewriter()
```

## Method Detail

### rewriteName(ServletContext, ServletRequest, String) Method

```
public String rewriteName(ServletContext servletContext,
                        ServletRequest request,
                        String name)
```

#### Overrides

```
URLRewriter.rewriteName(ServletContext, ServletRequest, String)
```

#### Parameters

*servletContext*

the current ServletContext.

*request*

the current HttpServletRequest.

*name*

the query parameter name to rewrite.

---

### rewriteURL(ServletContext, ServletRequest, ServletResponse, String, String) Method

```
public String rewriteURL(ServletContext servletContext,
                        ServletRequest request,
                        ServletResponse response,
                        String url,
                        String type)
```

#### Overrides

```
URLRewriter.rewriteURL(ServletContext, ServletRequest,
                        ServletResponse, String, String)
```

#### Parameters

*servletContext*

the current ServletContext.

*request*

the current `HttpServletRequest`.

*response*

the current `HttpServletResponse`.

*url*

the URL to rewrite.

*type*

the type of URL to be rewritten. This is one of the following values:

- ◇ `action`: a standard (non-resource) URL, for which SSL is not required
- ◇ `secure-action`: a standard (non-resource) URL, for which SSL is required
- ◇ `resource`: a resource (e.g., image) URL, for which SSL is not required
- ◇ `secure-resource`: a resource (e.g., image) URL, for which SSL is required



# TemplateHelper Class

*public class* **TemplateHelper**

extends Object  
implements Cloneable

Class for creating URLs from parameterized templates. This mechanism recognizes the following parameters in URL templates:

url:scheme	Scheme	Usage
url:domain	Domain	May be set
url:port	Port	May be set
url:path	Extra path information	Reserved, must be present in a template.
url:queryString	Reserved query parameters	Reserved, must be present in a template.

If one or more parameters of a template are not set at runtime, the implementation will use defaults if appropriate.

---

## Hierarchy

```

Object
  TemplateHelper
  
```

## All Implemented Interfaces

```

Cloneable
  
```

---

## Field Summary

```

    public static
    final String CONTROL_STATE_PARAM
        Control state parameter (reserved).

    protected HttpServletRequest httpRequest
        HTTP request.

    protected HttpServletResponse httpResponse
        HTTP response

    public static
    final String MODE_PARAM
        Window mode query parameter (reserved).

    public static
    final String PAGE_LABEL_PARAM
  
```

## Page Flow API Reference

Page label query parameter (reserved).

protected LinkedHashMap

### ***params***

Query parameters

public static

final String ***POSTBACK\_PARAM***

Postback request query parameter  
(reserved).

public static

final String ***STATE\_PARAM***

Window state query parameter (reserved).

public static

final String ***WINDOW\_LABEL\_PARAM***

Window label query parameter (reserved).

## Constructor Summary

***TemplateHelper***(*HttpServletRequest httpRequest, HttpServletResponse httpResponse*)

Constructs a `GenericURL`.

## Method Summary

public

void ***addParameter***(*String name, String value*)

Add a parameter.

public

Object ***clone***()

Returns a deep copy.

public

String ***getDomain***()

Returns the domain.

public

boolean ***getEncodeSession***()

Returns true if URL rewriting is enabled.

public

String ***getParameter***(*String name*)

Returns the value of the parameter.

public

List ***getParameters***(*String name*)

Returns the values of the given parameter.

public

String ***getPathPrefix***()

Returns the path prefix.

```

public getPort()
    int          Returns the port.

public
String getQueryString()

public
String getScheme()
    Returns the protocol scheme.

public
String getTemplate()
    Returns the current template.

public
String getTemplateName(String templateRef)

public
boolean hasTemplateRef(String templateRef)

public
    void removeParameter(String name)
        Removes the given parameter.

public
    void setDomain(String domain)
        Sets the domain.

public
    void setEncodeSession(boolean encodeSession)
        Enables URL rewriting for encoding a session ID in the
        URL.

public
    void setPath(String path)
        Sets path.

public
    void setPathPrefix(String pathPrefix)
        Sets a path prefix.

public
    void setPort(int port)
        Sets a port.

public
    void setQueryString(String queryString)

public
    void setScheme(String scheme)
        Sets the protocol scheme.

public
    void setTemplate(String templateName)
        Sets a URL template for this URL.

public
    void setUrl(String url)

public
String toString()
    Returns a string form (fully-qualified) of this URL.

```

## Methods from class `java.lang.Object`

`equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `wait`, `wait`, `wait`

## Field Detail

### **CONTROL\_STATE\_PARAM**

```
public static final String CONTROL_STATE_PARAM
```

Control state parameter (reserved).

---

### **httpRequest**

```
protected HttpServletRequest httpRequest
```

HTTP request.

---

### **httpResponse**

```
protected HttpServletResponse httpResponse
```

HTTP response

---

### **MODE\_PARAM**

```
public static final String MODE_PARAM
```

Window mode query parameter (reserved).

---

### **PAGE\_LABEL\_PARAM**

```
public static final String PAGE_LABEL_PARAM
```

Page label query parameter (reserved).

---

### **params**

```
protected LinkedHashMap params
```

Query parameters

---

## POSTBACK\_PARAM

```
public static final String POSTBACK_PARAM
```

Postback request query parameter (reserved).

---

## STATE\_PARAM

```
public static final String STATE_PARAM
```

Window state query parameter (reserved).

---

## WINDOW\_LABEL\_PARAM

```
public static final String WINDOW_LABEL_PARAM
```

Window label query parameter (reserved).

## Constructor Detail

### TemplateHelper

```
public TemplateHelper(HttpServletRequest httpRequest,
                      HttpServletResponse httpResponse)
```

Constructs a GenericURL.

## Method Detail

### addParameter(String, String) Method

```
public void addParameter(String name,
                        String value)
```

Add a parameter. Multiple values for the same parameter can be set by calling this method multiple times with the same name.

#### Parameters

*name*

name

*value*

value

---

## clone() Method

```
public Object clone()
```

Returns a deep copy.

### *Overrides*

```
Object.clone()
```

### Returns

*clone*

---

## getDomain() Method

```
public String getDomain()
```

Returns the domain. If no domain was previously set, returns the domain that was used to make the current request.

### Returns

*domain*

---

## getEncodeSession() Method

```
public boolean getEncodeSession()
```

Returns `true` if URL rewriting is enabled.

### Returns

*boolean*

---

## getParameter(String) Method

```
public String getParameter(String name)
```

Returns the value of the parameter. If the parameter has several values, returns the first value.

### Parameters

*name*

name of the parameter

## Returns

*value value of the parameter*

---

## getParameters(String) Method

```
public List getParameters(String name)
```

Returns the values of the given parameter.

## Parameters

*name*

name of the parameter

## Returns

*values of the parameter*

---

## getPathPrefix() Method

```
public String getPathPrefix()
```

Returns the path prefix.

## Returns

*path prefix*

---

## getPort() Method

```
public int getPort()
```

Returns the port. If no port was previously set, returns the port that was used to make the current request.

## Returns

*port*

---

## getQueryString() Method

```
public String getQueryString()
```

---

## getScheme() Method

```
public String getScheme()
```

Returns the protocol scheme. If no protocol was previously set, returns the protocol that was used to make the current request.

### Returns

*protocol scheme*

---

## getTemplate() Method

```
public String getTemplate()
```

Returns the current template.

### Returns

*URL template*

---

## getTemplateName(String) Method

```
public String getTemplateName(String templateRef)
```

---

## hasTemplateRef(String) Method

```
public boolean hasTemplateRef(String templateRef)
```

---

## removeParameter(String) Method

```
public void removeParameter(String name)
```

Removes the given parameter.

### Parameters

*name*

name

---

## setDomain(String) Method

```
public void setDomain(String domain)
```

Sets the domain. If a domain is not specified, the domain that was used to make the current request will be



used.

### Parameters

*domain*  
domain

---

### setEncodeSession(boolean) Method

```
public void setEncodeSession(boolean encodeSession)
```

Enables URL rewriting for encoding a session ID in the URL. If not set to `true`, does not encode the URL with the session ID. Default is `true`.

### Parameters

*encodeSession*  
boolean

---

### setPath(String) Method

```
public void setPath(String path)
```

Sets path.

### Parameters

*path*  
path

---

### setPathPrefix(String) Method

```
public void setPathPrefix(String pathPrefix)
```

Sets a path prefix.

### Parameters

*pathPrefix*  
path prefix

---

### setPort(int) Method

```
public void setPort(int port)
```

Sets a port. If a port is not specified, the port that was used to make the current request will be used.

## Parameters

*port*  
port

---

## setQueryString(String) Method

```
public void setQueryString(String queryString)
```

---

## setScheme(String) Method

```
public void setScheme(String scheme)
```

Sets the protocol scheme. Should be one of "HTTP" or "HTTPS". If a protocol is not specified, the protocol that was used to make the current request will be used.

## Parameters

*scheme*  
protocol scheme

---

## setTemplate(String) Method

```
public void setTemplate(String templateName)
```

Sets a URL template for this URL.

## Parameters

*templateName*  
name of the url template

---

## setUrl(String) Method

```
public void setUrl(String url)
```

---

## toString() Method

```
public String toString()
```

Returns a string form (fully-qualified) of this URL.

## Overrides

```
Object.toString()
```

## Returns

*string*

# URLRewriter Class

*public abstract class URLRewriter*

extends Object

URL rewriter interface. Offers methods for rewriting URLs/query parameters, and adding URL rewriters to the chain.

---

## Hierarchy

Object  
URLRewriter

## Direct Known Subclasses

DefaultURLRewriter

---

## Field Summary

```
public
static String ACTION_SECURE
    Passed to URLRewriter.rewriteURL(ServletContext,
    ServletRequest, ServletResponse, String, String) for normal
    (non-resource) secure URLs.

public
static String ACTION_UNSECURE
    Passed to URLRewriter.rewriteURL(ServletContext,
    ServletRequest, ServletResponse, String, String) for normal
    (non-resource) non-secure URLs.

public
static String RESOURCE_SECURE
    Passed to URLRewriter.rewriteURL(ServletContext,
    ServletRequest, ServletResponse, String, String) for resource (e.g.,
    image) secure URLs.

public
static String RESOURCE_UNSECURE
    Passed to URLRewriter.rewriteURL(ServletContext,
    ServletRequest, ServletResponse, String, String) for resource (e.g.,
    image) non-secure URLs.
```

## Constructor Summary

*URLRewriter()*

## Method Summary

```

    public
URLRewriter getNextRewriter()
    Get the next URLRewriter in the chain.

    public
abstract rewriteName(ServletContext servletContext, ServletRequest request,
    String String name)
    Rewrite the given query parameter name.

    public
abstract rewriteURL(ServletContext servletContext, ServletRequest request,
    String ServletResponse response, String url, String type)
    Rewrite the given URL.

    public
    void setNextRewriter(URLRewriter nextRewriter)
    Set the next URLRewriter in the chain.

```

### Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

## Field Detail

### ACTION\_SECURE

```
public static String ACTION_SECURE
```

Passed to `URLRewriter.rewriteURL(ServletContext, ServletRequest, ServletResponse, String, String)` for normal (non-resource) secure URLs.

---

### ACTION\_UNSECURE

```
public static String ACTION_UNSECURE
```

Passed to `URLRewriter.rewriteURL(ServletContext, ServletRequest, ServletResponse, String, String)` for normal (non-resource) non-secure URLs.

---

## RESOURCE\_SECURE

```
public static String RESOURCE_SECURE
```

Passed to `URLRewriter.rewriteURL(ServletContext, ServletRequest, ServletResponse, String, String)` for resource (e.g., image) secure URLs.

---

## RESOURCE\_UNSECURE

```
public static String RESOURCE_UNSECURE
```

Passed to `URLRewriter.rewriteURL(ServletContext, ServletRequest, ServletResponse, String, String)` for resource (e.g., image) non-secure URLs.

## Constructor Detail

### URLRewriter

```
public URLRewriter()
```

## Method Detail

### getNextRewriter() Method

```
public URLRewriter getNextRewriter()
```

Get the next `URLRewriter` in the chain.

### Related Topics

```
URLRewriter.setNextRewriter(URLRewriter)
```

---

### rewriteName(ServletContext, ServletRequest, String) Method

```
public abstract String rewriteName(ServletContext servletContext,
                                   ServletRequest request,
                                   String name)
```

Rewrite the given query parameter name.

**Parameters***servletContext*

the current ServletContext.

*request*

the current HttpServletRequest.

*name*

the query parameter name to rewrite.

**Returns***the rewritten query parameter name.***rewriteURL(ServletContext, ServletRequest, ServletResponse, String, String)  
Method**

```
public abstract String rewriteURL(ServletContext servletContext,
                                  ServletRequest request,
                                  ServletResponse response,
                                  String url,
                                  String type)
```

Rewrite the given URL.

**Parameters***servletContext*

the current ServletContext.

*request*

the current HttpServletRequest.

*response*

the current HttpServletResponse.

*url*

the URL to rewrite.

*type*

the type of URL to be rewritten. This is one of the following values:

- ◇ `action`: a standard (non-resource) URL, for which SSL is not required
- ◇ `secure-action`: a standard (non-resource) URL, for which SSL is required
- ◇ `resource`: a resource (e.g., image) URL, for which SSL is not required
- ◇ `secure-resource`: a resource (e.g., image) URL, for which SSL is required

**Returns**

*the rewritten URL.*

---

**setNextRewriter(URLRewriter) Method**

```
public void setNextRewriter(URLRewriter nextRewriter)
```

Set the next URLRewriter in the chain. When

URLRewriterService.rewriteURL(ServletContext, ServletRequest, ServletResponse, String, String) is called, it will start with the URLRewriter registered in the request (URLRewriterService.registerURLRewriter(ServletRequest, URLRewriter)), and call URLRewriter.rewriteURL(ServletContext, ServletRequest, ServletResponse, String, String) on each URLRewriter in the chain.



# URLRewriterService Class

*public class* **URLRewriterService**

extends Object

Methods for registering URL rewriters, and for rewriting URLs using registered rewriters.

---

## Hierarchy

```

Object
  URLRewriterService
  
```

---

## Constructor Summary

**URLRewriterService()**

## Method Summary

```

public
static getActionMappingName(String action)
String      Return the form action converted into an action mapping path.

public
static
final getDefaultRewriter()
URLRewriter      Get the default URLRewriter.

public
static needsSecure(ServletRequest request, ServletContext context,
boolean String uri, boolean stripContextPath)
      Tell whether a given URI should be written to be secure.

public
static registerURLRewriter(ServletRequest request, URLRewriter
void rewriter)
      Register a URLRewriter in the request.

public
static rewriteName(ServletContext servletContext, ServletRequest
String request, String name)
      Rewrite the given parameter name, using the registered
      URLRewriter.

public
static
  
```

String **rewriteURL**(ServletContext servletContext, ServletRequest request, ServletResponse response, String url, String type)  
 Rewrite the given URL, using the registered URLRewriter.

public  
 static **rewriteURL**(ServletContext servletContext, ServletRequest request, ServletResponse response, String url, String type, boolean doEncode)  
 Rewrite the given URL, using the registered URLRewriter.

public  
 static **unregisterURLRewriter**(ServletRequest request)  
 URLRewriter      Unregister the URLRewriter from the request.

### Methods from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructor Detail

### URLRewriterService

public URLRewriterService()

## Method Detail

### getActionMappingName(String) Method

public static String getActionMappingName(String action)

Return the form action converted into an action mapping path. The value of the *action* property is manipulated as follows in computing the name of the requested mapping:

- Any filename extension is removed (on the theory that extension mapping is being used to select the controller servlet).
- If the resulting value does not start with a slash, then a slash is prepended.

### Parameters

*action*  
 the action name to be converted.

### Returns

*an action path, suitable for lookup in the Struts configuration file.*

---

## getDefaultRewriter() Method

```
public static final URLRewriter getDefaultRewriter()
```

Get the default URLRewriter.

### Returns

*an instance of DefaultURLRewriter.*

---

## needsSecure(ServletRequest, ServletContext, String, boolean) Method

```
public static boolean needsSecure(ServletRequest request,
                                 ServletContext context,
                                 String uri,
                                 boolean stripContextPath)
```

Tell whether a given URI should be written to be secure.

### Parameters

*request*

the current HttpServletRequest.

*context*

the current ServletContext.

*uri*

the URI to check.

*stripContextPath*

if true, strip the webapp context path from the URI before processing it.

### Returns

*true when:*

- ◆ *the given URI is configured in the deployment descriptor to be secure (according to PageFlowUtils.isSecureResource(String, ServletContext, HttpServletRequest)), or*
- ◆ *the given URI is not configured in the deployment descriptor, and the current request is secure (HttpServletRequest.isSecure() returns true).*

*false when:*

- ◆ *the given URI is configured explicitly in the deployment descriptor to be unsecure (according to PageFlowUtils.isSecureResource(String, ServletContext, HttpServletRequest)), or*
  - ◆ *the given URI is not configured in the deployment descriptor, and the current request is unsecure (HttpServletRequest.isSecure() returns false).*
-

## registerURLRewriter(ServletRequest, URLRewriter) Method

```
public static void registerURLRewriter(ServletRequest request,
                                       URLRewriter rewriter)
```

Register a URLRewriter in the request. This rewriter will be used if URLRewriterService.rewriteURL(ServletContext, ServletRequest, ServletResponse, String, String) is called.

### Parameters

*request*

the current HttpServletRequest.

*rewriter*

the URLRewriter to register.

---

## rewriteName(ServletContext, ServletRequest, String) Method

```
public static String rewriteName(ServletContext servletContext,
                                ServletRequest request,
                                String name)
```

Rewrite the given parameter name, using the registered URLRewriter.

### Parameters

*servletContext*

the current ServletContext.

*request*

the current HttpServletRequest.

*name*

the parameter name to rewrite.

### Returns

*the rewritten parameter name.*

---

## rewriteURL(ServletContext, ServletRequest, ServletResponse, String, String) Method

```
public static String rewriteURL(ServletContext servletContext,
                                ServletRequest request,
                                ServletResponse response,
                                String url,
                                String type)
```

Rewrite the given URL, using the registered URLRewriter.

### Parameters

*servletContext*

the current ServletContext.

*request*

the current HttpServletRequest.

*response*

the current HttpServletResponse.

*url*

the URL to rewrite.

*type*

the type of URL to be rewritten. This is one of the following values:

- ◊ *action*: a standard (non-resource) URL, for which SSL is not required
- ◊ *secure-action*: a standard (non-resource) URL, for which SSL is required
- ◊ *resource*: a resource (e.g., image) URL, for which SSL is not required
- ◊ *secure-resource*: a resource (e.g., image) URL, for which SSL is required

### Returns

*the rewritten URL.*

### Related Topics

`URLRewriterService.registerURLRewriter(ServletRequest, URLRewriter)`

---

## rewriteURL(ServletContext, ServletRequest, ServletResponse, String, String, boolean) Method

```
public static String rewriteURL(ServletContext servletContext,
                               ServletRequest request,
                               ServletResponse response,
                               String url,
                               String type,
                               boolean doEncode)
```

Rewrite the given URL, using the registered URLRewriter.

### Parameters

*servletContext*

the current ServletContext.

*request*

the current HttpServletRequest.

*response*

the current `HttpServletResponse`.

*url*

the URL to rewrite.

*type*

the type of URL to be rewritten. This is one of the following values:

- ◇ `action`: a standard (non-resource) URL, for which SSL is not required
- ◇ `secure-action`: a standard (non-resource) URL, for which SSL is required
- ◇ `resource`: a resource (e.g., image) URL, for which SSL is not required
- ◇ `secure-resource`: a resource (e.g., image) URL, for which SSL is required

*doEncode*

if true, the rewritten URL will be encoded using  
`HttpServletResponse.encodeRedirectURL(String)`.

## Returns

*the rewritten URL.*

## Related Topics

`URLRewriterService.registerURLRewriter(ServletRequest, URLRewriter)`

---

## unregisterURLRewriter(ServletRequest) Method

```
public static URLRewriter unregisterURLRewriter(ServletRequest request)
```

Unregister the `URLRewriter` from the request.

## Parameters

*request*

the current `HttpServletRequest`.

## Returns

*the `URLRewriter` that was unregistered, or null if there was none registered.*

## Related Topics

`URLRewriterService.registerURLRewriter(ServletRequest, URLRewriter)`

# com.bea.wlw.netui.tags Package

## Interface Summary

DataProvider	The DataProvider interface is implemented by objects that need to provide their children with data and wish to make it available to them with the container binding context.
ErrorReporter	
IAttributeConsumer	This interfaces is implemented by tags that allow attributes to be set externally to the tag.
ScriptReporter	
TagValue	

## Class Summary

AbstractBaseTag	Abstract tag class that most netui tags are based from.
AbstractPageError	This is a "struct" class that contains the base information for an error occuring in a tag on a page.
DataSourceTag	Abstract tag class that adds support for a dataSource on the tag.
DefaultableDataSourceTag	Abstract tag class that adds support for a default value on the tag.
EvalErrorInfo	This is a "struct" class that contains the information for an expression evaluation error.
JavaScriptUtils	
OptionsDataSourceTag	Abstract tag class that adds support for an OptionsDataSource.
TagErrorInfo	This is a "struct" class that contains the information for an expression evaluation error.

# AbstractBaseTag Class

*public abstract class AbstractBaseTag*

extends BodyTagSupport

Abstract tag class that most netui tags are based from. AbstractBaseTag provides the following functionality:

- Error registration and display
  - NetUI Expression handling
  - Base attribute rendering
- 

## *Hierarchy*

```
Object
  TagSupport
    BodyTagSupport
      AbstractBaseTag
```

## *All Implemented Interfaces*

BodyTag, IterationTag, Serializable, Tag

## *Direct Known Subclasses*

AbstractCallMethod, Anchor, Attribute, Base, BindingUpdateErrors, Button, CheckBoxOption, Content, DataSourceTag, DeclareBundle, DeclareControl, DeclarePageInput, Error, Errors, Form, FormatTag, GetData, GetNetuiTagName, Image, ImageButton, IncludeSection, Label, Message, MessageArg, MethodParameter, Node, Parameter, ParameterMap, RadioButtonOption, RewriteName, RewriteURL, ScriptContainer, Section, SelectOption, SetAttribute, StructuredBaseTag, Template, Visible

---

## Field Summary

```
protected
  static ATTR_GENERAL
final int      int

protected
  static ATTR_GENERAL_EXPRESSION
final int      int

protected
  static ATTR_JAVASCRIPT
final int      int

protected
  static ATTR_STYLE
```



```

    final int          int
    protected
        static
    final String CLASS
                String

    protected
        static
    final String ID
                String

    public
        static
    final String JAVASCRIPT_STATUS
                String

    protected
        static
    final String NAME
                String

    public
        static
    final String NETUI_UNIQUE_CNT
                String

    protected
        static
    final String ONCLICK
                String

    protected
        static
    final String ONDBLCLICK
                String

    protected
        static
    final String ONKEYDOWN
                String

    protected
        static
    final String ONKEYPRESS
                String

    protected
        static
    final String ONKEYUP
                String

    protected
        static
    final String ONMOUSEDOWN
                String

    protected
        static
    final String ONMOUSEMOVE
                String

    protected
        static
    final String ONMOUSEOUT
                String

    protected
        static
    final String ONMOUSEOVER
                String

    protected
        static
    final String ONMOUSEUP
                String

    protected

```

```

    static STYLE
final String      String
    protected
    static TABINDEX
final String      String

```

#### Fields from `javax.servlet.jsp.tagext.BodyTagSupport`

`bodyContent`

#### Fields from `javax.servlet.jsp.tagext.TagSupport`

`id`, `pageContext`

#### Fields from interface `javax.servlet.jsp.tagext.BodyTag`

`EVAL_BODY_BUFFERED`, `EVAL_BODY_TAG`

#### Fields from interface `javax.servlet.jsp.tagext.IterationTag`

`EVAL_BODY_AGAIN`

#### Fields from interface `javax.servlet.jsp.tagext.Tag`

`EVAL_BODY_INCLUDE`, `EVAL_PAGE`, `SKIP_BODY`, `SKIP_PAGE`

## Constructor Summary

*`AbstractBaseTag()`*

## Method Summary

```

    protected
        String addTagIdMapping(String tagId, String value)
            This method will add a tagId and value to the ScriptRepoter TagId map.
    protected
        String evaluateAttributeToString(String expression, String attrName)
            This method will evaluate an attribute and return the real value or null.
protected final
        String filter(String value)
    protected
        String filter(String value, boolean markupHTMLSpaceReturn)
            Filter the specified string for characters that are sensitive to HTML
            interpreters, returning the string with these characters replaced by the
            corresponding character entities.

```

```

    protected getJavaScriptUtils(HttpServletRequest request)
JavaScriptUtils      This method will return the javaScriptUtils that is associated with this
                      request.

    public Form
                      getNearestForm()
                      Returns the closest parent form tag, or null if there is none.

    protected
ScriptReporter getScriptReporter()
public abstract
    String getTagName()
    protected
    Locale getUserLocale()
protected void
    localRelease()
    This is a method that will initialize all temp state on a tag and should be called
    in the doEndTag method.

    public String
    registerTagError(String message)
    This will report an error from a tag.

    public void
    release()

protected final
    void renderAttribute(UnsyncStringBuffer buf, String name, String
    value)
    This method will write append an attribute value to a StringBuffer.

    protected
    String rewriteName(String name)
    This method will rewrite the name (id) by passing it to the URL Rewriter and
    getting back a value.

protected final
    void write(String string)
    This mehod will write the passed string to the response.

```

## Methods from javax.servlet.jsp.tagext.BodyTagSupport

doAfterBody, doEndTag, doInitBody, doStartTag, getBodyContent, getPreviousOut, setBodyContent

## Methods from javax.servlet.jsp.tagext.TagSupport

doAfterBody, doEndTag, doStartTag, findAncestorWithClass, getId, getParent, getValue, getValues, release, removeValue, setId, setPageContext, setParent, setValue

## Methods from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

**Methods from interface `javax.servlet.jsp.tagext.BodyTag`**

`doInitBody`, `setBodyContent`

**Methods from interface `javax.servlet.jsp.tagext.IterationTag`**

`doAfterBody`

**Methods from interface `javax.servlet.jsp.tagext.Tag`**

`doEndTag`, `doStartTag`, `getParent`, `release`, `setPageContext`, `setParent`

## Field Detail

### ATTR\_GENERAL

```
protected static final int ATTR_GENERAL
```

---

### ATTR\_GENERAL\_EXPRESSION

```
protected static final int ATTR_GENERAL_EXPRESSION
```

---

### ATTR\_JAVASCRIPT

```
protected static final int ATTR_JAVASCRIPT
```

---

### ATTR\_STYLE

```
protected static final int ATTR_STYLE
```

---

### CLASS

```
protected static final String CLASS
```

---

### ID

```
protected static final String ID
```

---

### JAVASCRIPT\_STATUS

```
public static final String JAVASCRIPT_STATUS
```

---

## NAME

```
protected static final String NAME
```

---

## NETUI\_UNIQUE\_CNT

```
public static final String NETUI_UNIQUE_CNT
```

---

## ONCLICK

```
protected static final String ONCLICK
```

---

## ONDBLCLICK

```
protected static final String ONDBLCLICK
```

---

## ONKEYDOWN

```
protected static final String ONKEYDOWN
```

---

## ONKEYPRESS

```
protected static final String ONKEYPRESS
```

---

## ONKEYUP

```
protected static final String ONKEYUP
```

---

## ONMOUSEDOWN

```
protected static final String ONMOUSEDOWN
```

---

## ONMOUSEMOVE

```
protected static final String ONMOUSEMOVE
```

---

## ONMOUSEOUT

```
protected static final String ONMOUSEOUT
```

---

## ONMOUSEOVER

```
protected static final String ONMOUSEOVER
```

---

## ONMOUSEUP

```
protected static final String ONMOUSEUP
```

---

## STYLE

```
protected static final String STYLE
```

---

## TABINDEX

```
protected static final String TABINDEX
```

## Constructor Detail

### AbstractBaseTag

```
public AbstractBaseTag()
```

## Method Detail

### addTagIdMapping(String, String) Method

```
protected String addTagIdMapping(String tagId,
                                String value)
```

This method will add a tagId and value to the ScriptReporter TagId map. The a ScriptContainer tag will create a JavaScript table that allows the container, such as a portal, to rewrite the id so it's unique. The real name may be looked up based upon the tagId. If the no ScriptReporter is found, a script string will be returned to the caller so they can output the script block.

---

### evaluateAttributeToString(String, String) Method

```
protected String evaluateAttributeToString(String expression,
                                           String attrName)
```

This method will evaluate an attribute and return the real value or null. If the attribute contains an expression the expression will be evaluated and the object found returned as a String. If the attributes

contains an expression and doesn't return a value then `null` will be returned. if the attribute doesn't contain an expression it will be returned unchanged.

---

## filter(String) Method

```
protected final String filter(String value)
```

---

## filter(String, boolean) Method

```
protected String filter(String value,  
                        boolean markupHTMLSpaceReturn)
```

Filter the specified string for characters that are sensitive to HTML interpreters, returning the string with these characters replaced by the corresponding character entities.

### Parameters

*value*

The string to be filtered and returned

*markupHTMLSpaceReturn*

convert space characters and return characters to `&nbsp;`; and `<br />` markup for html.

---

## getJavaScriptUtils(HttpServletRequest request) Method

```
protected JavaScriptUtils getJavaScriptUtils(HttpServletRequest request)
```

This method will return the `JavaScriptUtils` that is associated with this request. If this doesn't exist, it will be created before it is returned.

### Parameters

*request*

the `HttpServletRequest` associated with this request

### Returns

*a JavaScriptUtils associated with the request*

---

## getNearestForm() Method

```
public Form getNearestForm()
```

Returns the closest parent form tag, or null if there is none.

---

## getScriptReporter() Method

```
protected ScriptReporter getScriptReporter()
```

---

## getTagName() Method

```
public abstract String getTagName()
```

---

## getUserLocale() Method

```
protected Locale getUserLocale()
```

### Returns

*the Locale object to use when rendering this tag*

---

## localRelease() Method

```
protected void localRelease()
```

This is a method that will initialize all temp state on a tag and should be called in the doEndTag method.

---

## registerTagError(String) Method

```
public String registerTagError(String message)
```

This will report an error from a tag. The error will contain a message. If error reporting is turned off, the message will be returned and the caller should throw a JspException to report the error.

### Parameters

*message*  
– the message to register with the error

### Returns

*null*

---

## release() Method

```
public void release()
```

### Overrides

```
BodyTagSupport.release()
```

---



## renderAttribute(UnsyncStringBuffer, String, String) Method

```
protected final void renderAttribute(UnsyncStringBuffer buf,  
                                     String name,  
                                     String value)
```

This method will write append an attribute value to a StringBuffer. The method assumes that the attr is not null. If the value is null the attribute will not be appended to the StringBuffer.

---

## rewriteName(String) Method

```
protected String rewriteName(String name)
```

This method will rewrite the name (id) by passing it to the URL Rewriter and getting back a value.

---

## write(String) Method

```
protected final void write(String string)
```

This method will write the passed string to the response.

### Parameters

*string*  
to be written to the response.

# AbstractPageError Class

*public abstract class AbstractPageError*

extends Object

This is a "struct" class that contains the base information for an error occuring in a tag on a page.

---

## Hierarchy

Object  
AbstractPageError

## Direct Known Subclasses

EvalErrorInfo, TagErrorInfo

---

## Field Summary

public int

**errorNo**  
The number of the error on the  
page.

public String

**tagType**  
The number of the error on the  
page.

## Constructor Summary

*AbstractPageError()*

## Method Summary

Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`,  
`toString`, `wait`, `wait`, `wait`

## Field Detail

### **errorNo**

```
public int errorNo
```

The number of the error on the page.

---

### **tagType**

```
public String tagType
```

The number of the error on the page.

## Constructor Detail

### **AbstractPageError**

```
public AbstractPageError()
```

# DataAccessProvider Interface

*public interface **DataAccessProvider***

The `DataAccessProvider` interface is implemented by objects that need to provide their children with data and wish to make it available to them with the `container` binding context. Expression evaluation will process all `container` context references against this interface; several read-only properties are exposed:

Method	NetUI Data Binding Expression	Required
<code>getCurrentIndex()</code>	<code>container.index</code>	Yes
<code>getCurrentItem()</code>	<code>container.item</code>	Yes
<code>getCurrentMetadata()</code>	<code>container.metadata</code>	No
<code>getDataSource()</code>	<code>container.dataSource</code>	Yes
<code>getProviderParent()</code>	<code>container.container</code>	Yes

In cases where a `DataAccessProvider` contains another `DataAccessProvider`, the grandparent `DataAccessProvider` may be referenced with the binding expression `container.container`. For example, the item, with the property `firstName`, may be accessed with the expression `container.container.item.firstName`. The general use of the `DataAccessProvider` is as an interface that is implemented by repeating databound tags that iterate over a data set and render each item in that data set. The item and iteration index are exposed through this interface and can be bound to by tags inside of the repeating tag that implements the `DataAccessProvider` interface. This binding expression should start with `container` and reference one of the properties above.

## All Known Implementing Classes

`CellRepeater`, `Grid`, `Repeater`

## Method Summary

```

    public int
        getCurrentIndex()
            Get the current index in this iteration.

    public Object
        getCurrentItem()
            Get the current data item in this DataAccessProvider.

    public Object
        getCurrentMetadata()
            Get a metadata object for the current item.

    public String
        getDataSource()
            Get the expression that references the data item to which the
            DataAccessProvider is bound.

    public

```

DataProvider **getProviderParent()**

Get the parent DataAccessProvider of a DataAccessProvider.

```
public void
    setDataSource(String dataSource)
```

## Method Detail

### getCurrentIndex() Method

```
public int getCurrentIndex()
```

Get the current index in this iteration. This should be a zero based integer that increments after each iteration.

#### Returns

*the current index of iteration or 0*

---

### getCurrentItem() Method

```
public Object getCurrentItem()
```

Get the current data item in this DataAccessProvider.

#### Returns

*the current data item or null*

---

### getCurrentMetadata() Method

```
public Object getCurrentMetadata()
```

Get a metadata object for the current item. This interface is optional, and implementations of this interface are provided by the DataAccessProvider interface. See these implementations for information about their support for current item metadata.

#### Returns

*the current metadata or null if no metadata can be found or metadata is not supported by a DataAccessProvider implementation*

---

### getDataSource() Method

```
public String getDataSource()
```

Get the expression that references the data item to which the DataAccessProvider is bound.

## Returns

*the expression referencing the data source or `null` if no `dataSource` is set*

---

## getProviderParent() Method

```
public DataAccessProvider getProviderParent()
```

Get the parent `DataAccessProvider` of a `DataAccessProvider`. A `DataAccessProvider` implementation may be able to nest `DataAccessProviders`. In this case, it can be useful to be able to also nest access to data from parent providers. Implementations of this interface are left with having to discover and export parents. The return value from this call on an implementing Object can be `null`.

## Returns

*the parent `DataAccessProvider` or `null` if this method is not supported or the parent can not be found.*

---

## setDataSource(String) Method

```
public void setDataSource(String dataSource)
```

# DataSourceTag Class

*public abstract class DataSourceTag*

extends AbstractBaseTag

Abstract tag class that adds support for a dataSource on the tag. The dataSource is where this tag gets its data from and writes its data to.

For example, would pull from the form bean's foo property to populate the textbox. It would also set the value of the textbox on a post back into the foo property of the form bean.

The dataSource can either be an expression or the name of a form bean property. In the above example, would have the exact same behavior.

---

## Hierarchy

```
Object
  TagSupport
    BodyTagSupport
      AbstractBaseTag
        DataSourceTag
```

## All Implemented Interfaces

BodyTag, IterationTag, Serializable, Tag

## Direct Known Subclasses

CellRepeater, DefaultableDataSourceTag, FileUpload, Grid, Hidden, Repeater, Tree

---

## Field Summary

### Fields from com.bea.wlw.netui.tags.AbstractBaseTag

ATTR\_GENERAL, ATTR\_GENERAL\_EXPRESSION, ATTR\_JAVASCRIPT, ATTR\_STYLE, CLASS, ID, JAVASCRIPT\_STATUS, NAME, NETUI\_UNIQUE\_CNT, ONCLICK, ONDBLCLICK, ONKEYDOWN, ONKEYPRESS, ONKEYUP, ONMOUSEDOWN, ONMOUSEMOVE, ONMOUSEOUT, ONMOUSEOVER, ONMOUSEUP, STYLE, TABINDEX

### Fields from javax.servlet.jsp.tagext.BodyTagSupport

bodyContent

### Fields from `javax.servlet.jsp.tagext.TagSupport`

`id`, `pageContext`

### Fields from interface `javax.servlet.jsp.tagext.BodyTag`

`EVAL_BODY_BUFFERED`, `EVAL_BODY_TAG`

### Fields from interface `javax.servlet.jsp.tagext.IterationTag`

`EVAL_BODY_AGAIN`

### Fields from interface `javax.servlet.jsp.tagext.Tag`

`EVAL_BODY_INCLUDE`, `EVAL_PAGE`, `SKIP_BODY`, `SKIP_PAGE`

## Constructor Summary

*`DataSourceTag()`*

## Method Summary

```
public
String getDataSource()
    Gets the tag's data source (can be an
    expression).

protected
void localRelease()
    Release any acquired resources.

public
void setDataSource(String dataSource)
    Sets the tag's data source (can be an
    expression).
```

### Methods from `com.bea.wlw.netui.tags.AbstractBaseTag`

`addTagIdMapping`, `evaluateAttributeToString`, `filter`, `filter`,  
`getJavaScriptUtils`, `getNearestForm`, `getScriptReporter`, `getTagName`,  
`getUserLocale`, `registerTagError`, `release`, `renderAttribute`, `rewriteName`,  
`write`

### Methods from `javax.servlet.jsp.tagext.BodyTagSupport`

`doAfterBody`, `doEndTag`, `doInitBody`, `doStartTag`, `getBodyContent`,  
`getPreviousOut`, `release`, `setBodyContent`



### Methods from `javax.servlet.jsp.tagext.TagSupport`

`doAfterBody`, `doEndTag`, `doStartTag`, `findAncestorWithClass`, `getId`, `getParent`, `getValue`, `getValues`, `release`, `removeValue`, `setId`, `setPageContext`, `setParent`, `setValue`

### Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

### Methods from interface `javax.servlet.jsp.tagext.BodyTag`

`doInitBody`, `setBodyContent`

### Methods from interface `javax.servlet.jsp.tagext.IterationTag`

`doAfterBody`

### Methods from interface `javax.servlet.jsp.tagext.Tag`

`doEndTag`, `doStartTag`, `getParent`, `release`, `setPageContext`, `setParent`

## Constructor Detail

### `DataSourceTag`

```
public DataSourceTag()
```

## Method Detail

### `getDataSource()` Method

```
public String getDataSource()
```

Gets the tag's data source (can be an expression).

#### Returns

*the data source*

---

### `localRelease()` Method

```
protected void localRelease()
```

Release any acquired resources.

***Overrides***

`AbstractBaseTag.localRelease()`

---

**setDataSource(String) Method**

`public void setDataSource(String dataSource)`

Sets the tag's data source (can be an expression).

**Parameters**

*dataSource*

– the data source

# DefaultableDataSourceTag Class

*public abstract class DefaultableDataSourceTag*

extends DataSourceTag

Abstract tag class that adds support for a default value on the tag. The default value is used if the dataSource or value of the tag is null.

For example, would display "bar" for the label if the expression {actionForm.foo} resolved to null.

---

## ***Hierarchy***

```
Object
  TagSupport
    BodyTagSupport
      AbstractBaseTag
        DataSourceTag
          DefaultableDataSourceTag
```

## ***All Implemented Interfaces***

BodyTag, IterationTag, Serializable, Tag

## ***Direct Known Subclasses***

CheckBox, OptionsDataSourceTag, TextArea, TextBox

---

## **Field Summary**

### **Fields from com.bea.wlw.netui.tags.AbstractBaseTag**

ATTR\_GENERAL, ATTR\_GENERAL\_EXPRESSION, ATTR\_JAVASCRIPT, ATTR\_STYLE, CLASS, ID, JAVASCRIPT\_STATUS, NAME, NETUI\_UNIQUE\_CNT, ONCLICK, ONDBLCLICK, ONKEYDOWN, ONKEYPRESS, ONKEYUP, ONMOUSEDOWN, ONMOUSEMOVE, ONMOUSEOUT, ONMOUSEOVER, ONMOUSEUP, STYLE, TABINDEX

### **Fields from javax.servlet.jsp.tagext.BodyTagSupport**

bodyContent

### **Fields from javax.servlet.jsp.tagext.TagSupport**

id, pageContext

**Fields from interface javax.servlet.jsp.tagext.BodyTag**

EVAL\_BODY\_BUFFERED, EVAL\_BODY\_TAG

**Fields from interface javax.servlet.jsp.tagext.IterationTag**

EVAL\_BODY\_AGAIN

**Fields from interface javax.servlet.jsp.tagext.Tag**

EVAL\_BODY\_INCLUDE, EVAL\_PAGE, SKIP\_BODY, SKIP\_PAGE

## Constructor Summary

*DefaultableDataSourceTag()*

## Method Summary

```
public
String getDefaultValue()
    Gets the default value (can be an
    expression).
```

```
protected
void localRelease()
    Release any acquired resources.
```

```
public
void setDefaultValue(String defaultValue)
    Sets the default value (can be an
    expression).
```

**Methods from com.bea.wlw.netui.tags.DataSourceTag**

getDataSource, setDataSource

**Methods from com.bea.wlw.netui.tags.AbstractBaseTag**

```
addTagIdMapping, evaluateAttributeToString, filter, filter,
getJavaScriptUtils, getNearestForm, getScriptReporter, getTagName,
getUserLocale, localRelease, registerTagError, release, renderAttribute,
rewriteName, write
```

**Methods from javax.servlet.jsp.tagext.BodyTagSupport**

```
doAfterBody, doEndTag, doInitBody, doStartTag, getBodyContent,
getPreviousOut, release, setBodyContent
```

### Methods from `javax.servlet.jsp.tagext.TagSupport`

`doAfterBody`, `doEndTag`, `doStartTag`, `findAncestorWithClass`, `getId`, `getParent`, `getValue`, `getValues`, `release`, `removeValue`, `setId`, `setPageContext`, `setParent`, `setValue`

### Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

### Methods from interface `javax.servlet.jsp.tagext.BodyTag`

`doInitBody`, `setBodyContent`

### Methods from interface `javax.servlet.jsp.tagext.IterationTag`

`doAfterBody`

### Methods from interface `javax.servlet.jsp.tagext.Tag`

`doEndTag`, `doStartTag`, `getParent`, `release`, `setPageContext`, `setParent`

## Constructor Detail

### DefaultableDataSourceTag

```
public DefaultableDataSourceTag()
```

## Method Detail

### getDefaultValue() Method

```
public String getDefaultValue()
```

Gets the default value (can be an expression).

#### Returns

*the default value*

---

### localRelease() Method

```
protected void localRelease()
```

Release any acquired resources.

***Overrides***

`DataSourceTag.localRelease()`

---

**setDefaultValue(String) Method**

```
public void setDefaultValue(String defaultValue)
    throws JspException
```

Sets the default value (can be an expression).

**Parameters**

*defaultValue*  
– the default value

**Exceptions**

*JspException*

# ErrorReporter Interface

*public interface* **ErrorReporter**

---

## *All Known Implementing Classes*

Html, Template, Grid

---

## Field

### Summary

```
public
static CONTAINER_ERRORS
final String String
```

## Method

### Summary

```
public
void addError(AbstractPageError ape)
public
boolean isReporting()
    This boolean indicates if an ErrorReporter is reporting errors or
    not.
public
ArrayList returnErrors()
    Return an ArrayList of the errors
```

## Field Detail

### CONTAINER\_ERRORS

```
public static final String CONTAINER_ERRORS
```

## Method Detail

### addError(AbstractPageError) Method

```
public void addError(AbstractPageError ape)
```

---

## isReporting() Method

```
public boolean isReporting()
```

This boolean indicates if an ErrorReporter is reporting errors or not. The caller should check this before calling addError because the ErrorReporter may be off for some reason.

### Returns

*a boolean indicating if the tag is reporting errors or not.*

---

## returnErrors() Method

```
public ArrayList returnErrors()
```

Return an ArrayList of the errors

### Returns

*an ArrayList of all errors.*



# EvalErrorInfo Class

*public class EvalErrorInfo*

extends AbstractPageError

This is a "struct" class that contains the information for an expression evaluation error.

---

## Hierarchy

Object  
AbstractPageError  
EvalErrorInfo

---

## Field Summary

public String	<b>attr</b>	The attribute of the tag that contained the expression
public ExpressionEvaluationException	<b>evalExcp</b>	The exception which occurred
public String	<b>expression</b>	The expression that was being evaluated.

Fields from com.bea.wlw.netui.tags.AbstractPageError

errorNo, tagType

## Constructor Summary

*EvalErrorInfo()*

## Method Summary

Methods from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

## Field Detail

### **attr**

`public String attr`

The attribute of the tag that contained the expression

---

### **evalExcp**

`public ExpressionEvaluationException evalExcp`

The exception which occurred

---

### **expression**

`public String expression`

The expression that was being evaluated.

## Constructor Detail

### **EvalErrorInfo**

`public EvalErrorInfo()`

# IAttributeConsumer Interface

*public interface IAttributeConsumer*

This interfaces is implemented by tags that allow attributes to be set externally to the tag. The HTML tags for example, allow certain attributes to be set by children tags through the use of the `attribute` tag.

---

## All Known Implementing Classes

TextBox, TextArea, SelectOption, Select, RadioButtonOption, Label, ImageButton, Anchor, Image, Hidden, Form, FileUpload, CheckBoxOption, CheckBox, Button, Base

---

## Method Summary

```
public
    void setAttribute(String name, String value)
        Set an attribute value on the implementing class.
```

## Method Detail

### setAttribute(String, String) Method

```
public void setAttribute(String name,
                        String value)
    throws JspException
```

Set an attribute value on the implementing class. The `name` represents the name of the attribute. The `value` represents the value and may contain an expression. The `facet` is optional and may be used by complex types to target the attribute to a sub part of the generated markup. This method may result in errors being generated.

### Parameters

*name*

The name of the attribute. This value may not be null or the empty string.

*value*

The value of the attribute. This may contain an expression.

### Exceptions

*JspException*

A `JspException` may be thrown if there is an error setting the attribute.

# JavaScriptUtils Class

*public class JavaScriptUtils*

extends Object  
implements Serializable

---

## Hierarchy

Object  
JavaScriptUtils

## All Implemented Interfaces

Serializable

---

## Constructor Summary

*JavaScriptUtils()*

## Method Summary

```
public
static getFocusScript(String formName, String fieldName)
String          this will write out the form focus script block

public
static getString(String aKey, Object[] args)
String          Returns the string specified by aKey from the errors.properties
                bundle.

public
String getTagIdMapping(String tagId, String value)
                This method will write out a tagId map entry for when there isn't a
                ScriptContainer defined.

public
String writeAnchorFormSubmit(ScriptReporter scriptReporter)

public
void writeAnchorFormSubmitAction(UnsyncStringBuffer results,
String event, String action, String newAction)

public
String writeIdMap(ScriptReporter scriptRepoter, String entries)
```

```

public writeNetuiNameFunctions(ScriptReporter scriptReporter)
String
public
String writeRollover(ScriptReporter scriptReporter)
public
static writeScriptBlock(UnsyncStringBuffer results, String script)
void          This is a static method that will write a consistent look/feel to the tags
               and comment markup that appears around the JavaScript.

```

### Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`,  
`toString`, `wait`, `wait`, `wait`

## Constructor Detail

### JavaScriptUtils

```
public JavaScriptUtils()
```

## Method Detail

### getFocusScript(String, String) Method

```

public static String getFocusScript(String formName,
                                   String fieldName)

```

this will write out the form focus script block

---

### getString(String, Object[]) Method

```

public static String getString(String aKey,
                              Object[] args)

```

Returns the string specified by `aKey` from the `errors.properties` bundle.

#### Parameters

*aKey*

The key for the message pattern in the bundle.

*args*

The args to use in the message format.

---

## getTagIdMapping(String, String) Method

```
public String getTagIdMapping(String tagId,  
                             String value)
```

This method will write out a tagId map entry for when there isn't a ScriptContainer defined.

---

## writeAnchorFormSubmit(ScriptReporter) Method

```
public String writeAnchorFormSubmit(ScriptReporter scriptReporter)
```

---

## writeAnchorFormSubmitAction(UnsyncStringBuffer, String, String, String) Method

```
public void writeAnchorFormSubmitAction(UnsyncStringBuffer results,  
                                       String event,  
                                       String action,  
                                       String newAction)
```

---

## writeIdMap(ScriptReporter, String) Method

```
public String writeIdMap(ScriptReporter scriptReporter,  
                        String entries)
```

---

## writeNetuiNameFunctions(ScriptReporter) Method

```
public String writeNetuiNameFunctions(ScriptReporter scriptReporter)
```

---

## writeRollover(ScriptReporter) Method

```
public String writeRollover(ScriptReporter scriptReporter)
```

---

## writeScriptBlock(UnsyncStringBuffer, String) Method

```
public static void writeScriptBlock(UnsyncStringBuffer results,  
                                   String script)
```

This is a static method that will write a consistent look/feel to the tags and comment markup that appears around the JavaScript.

### Parameters

*results*

the UnsyncStringBuffer that will have the <script> tag written into

*script*

the JavaScript block

# OptionsDataSourceTag Class

*public abstract class OptionsDataSourceTag*

extends DefaultableDataSourceTag

Abstract tag class that adds support for an OptionsDataSource. OptionsDataSource is used to populate a set of option tag children for the containing tag.

For example, a set of CheckBoxOptions for a CheckBoxGroup.

---

## ***Hierarchy***

```
Object
  TagSupport
    BodyTagSupport
      AbstractBaseTag
        DataSourceTag
          DefaultableDataSourceTag
            OptionsDataSourceTag
```

## ***All Implemented Interfaces***

BodyTag, IterationTag, Serializable, Tag

## ***Direct Known Subclasses***

CheckBoxGroup, RadioButtonGroup, Select

---

## **Field Summary**

### **Fields from com.bea.wlw.netui.tags.AbstractBaseTag**

ATTR\_GENERAL, ATTR\_GENERAL\_EXPRESSION, ATTR\_JAVASCRIPT, ATTR\_STYLE, CLASS, ID, JAVASCRIPT\_STATUS, NAME, NETUI\_UNIQUE\_CNT, ONCLICK, ONDBLCLICK, ONKEYDOWN, ONKEYPRESS, ONKEYUP, ONMOUSEDOWN, ONMOUSEMOVE, ONMOUSEOUT, ONMOUSEOVER, ONMOUSEUP, STYLE, TABINDEX

### **Fields from javax.servlet.jsp.tagext.BodyTagSupport**

bodyContent

### **Fields from javax.servlet.jsp.tagext.TagSupport**

id, pageContext



**Fields from interface javax.servlet.jsp.tagext.BodyTag**

EVAL\_BODY\_BUFFERED, EVAL\_BODY\_TAG

**Fields from interface javax.servlet.jsp.tagext.IterationTag**

EVAL\_BODY\_AGAIN

**Fields from interface javax.servlet.jsp.tagext.Tag**

EVAL\_BODY\_INCLUDE, EVAL\_PAGE, SKIP\_BODY, SKIP\_PAGE

## Constructor Summary

*OptionsDataSourceTag()*

## Method Summary

```

    public
    String getOptionsDataSource()
        Gets the options datasource value (an expression).

    protected
    void localRelease()
        Release any acquired resources.

    public
    void setOptionsDataSource(String optionsDataSource)
        Sets the options datasource value (an expression).

```

**Methods from com.bea.wlw.netui.tags.DefaultableDataSourceTag**

getDefaultValue, setDefaultValue

**Methods from com.bea.wlw.netui.tags.DataSourceTag**

getDataSource, localRelease, setDataSource

**Methods from com.bea.wlw.netui.tags.AbstractBaseTag**

```

addTagIdMapping, evaluateAttributeToString, filter, filter,
getJavaScriptUtils, getNearestForm, getScriptReporter, getTagName,
getUserLocale, localRelease, registerTagError, release, renderAttribute,
rewriteName, write

```

**Methods from javax.servlet.jsp.tagext.BodyTagSupport**

`doAfterBody, doEndTag, doInitBody, doStartTag, getBodyContent, getPreviousOut, release, setBodyContent`

**Methods from `javax.servlet.jsp.tagext.TagSupport`**

`doAfterBody, doEndTag, doStartTag, findAncestorWithClass, getId, getParent, getValue, getValues, release, removeValue, setId, setPageContext, setParent, setValue`

**Methods from class `java.lang.Object`**

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

**Methods from interface `javax.servlet.jsp.tagext.BodyTag`**

`doInitBody, setBodyContent`

**Methods from interface `javax.servlet.jsp.tagext.IterationTag`**

`doAfterBody`

**Methods from interface `javax.servlet.jsp.tagext.Tag`**

`doEndTag, doStartTag, getParent, release, setPageContext, setParent`

## Constructor Detail

### OptionsDataSourceTag

```
public OptionsDataSourceTag()
```

## Method Detail

### getOptionsDataSource() Method

```
public String getOptionsDataSource()
```

Gets the options datasource value (an expression).

**Returns**

*the options datasource*

---

## localRelease() Method

```
protected void localRelease()
```

Release any acquired resources.

### *Overrides*

```
DefaultableDataSourceTag.localRelease()
```

---

## setOptionsDataSource(String) Method

```
public void setOptionsDataSource(String optionsDataSource)  
    throws JspException
```

Sets the options datasource value (an expression).

### Parameters

*optionsDataSource*  
– the options datasource

### Exceptions

*JspException*

# ScriptReporter Interface

*public interface ScriptReporter*

---

*All Known Implementing Classes*  
ScriptContainer

---

## Method Summary

```
public
    void addScriptCode(String script)
        This method will add Script as top level code that runs when the page is
        loaded.

public
    void addScriptFunction(String script)
        This method will add Script as a function.

public
    void addTagId(String tagId, String tagName)
        Adds a tagID and tagName to the id to name map.
```

## Method Detail

### addScriptCode(String) Method

```
public void addScriptCode(String script)
```

This method will add Script as top level code that runs when the page is loaded.

#### Parameters

*script*  
the text of the script

---

### addScriptFunction(String) Method

```
public void addScriptFunction(String script)
```

This method will add Script as a function.

#### Parameters

*script*  
the text of the function

---

## addTagId(String, String) Method

```
public void addTagId(String tagId,  
                    String tagName)
```

Adds a tagID and tagName to the id to name map. A script container can maintain a mapping of user ids to actual ids generated by the tagId. This method will track these.

### Parameters

*tagId*

– the id of a child tag.

*tagName*

– the name of a child tag.

# TagErrorInfo Class

*public class TagErrorInfo*

extends AbstractPageError

This is a "struct" class that contains the information for an expression evaluation error.

---

## Hierarchy

Object  
AbstractPageError  
TagErrorInfo

---

## Field Summary

public String  
    *message*  
        The attribute of the tag that contained the  
        expression

Fields from com.bea.wlw.netui.tags.AbstractPageError  
errorNo, tagType

## Constructor Summary

*TagErrorInfo()*

## Method Summary

Methods from class java.lang.Object  
clone, equals, finalize, getClass, hashCode, notify, notifyAll,  
toString, wait, wait, wait

## Field Detail

### **message**

```
public String message
```

The attribute of the tag that contained the expression

## Constructor Detail

### **TagErrorInfo**

```
public TagErrorInfo()
```

# TagValue Interface

*public interface TagValue*

## Method Summary

```
public  
Object getTagValue()  
    Get the value of the tag (the 'text' of a TextBox, for example)  
  
public  
    void setTagValue(Object value)  
        Set the value of the tag (the 'text' of a TextBox, for example)
```

## Method Detail

### getTagValue() Method

```
public Object getTagValue()
```

Get the value of the tag (the 'text' of a TextBox, for example)

---

### setTagValue(Object) Method

```
public void setTagValue(Object value)
```

Set the value of the tag (the 'text' of a TextBox, for example)



# com.bea.wlw.netui.tags.databinding Package

This package defines a set of JSP tag extensions providing repeating databinding support.

The tags in this package fall into one of two categories:

- Method invocation tags
- Repeating databinding tags

The first tag set is used to invoke methods on Controls or PageFlowController instances. The former allows access to controls from pages, and the latter allows code that would normally be written in a JSP page to be shared among pages through the pages' shared PageFlow.

The second tagset consists of repeating tags that are used to render data sets into a JSP page. A repeating tag is a tag that, given a data set, renders each item or a subset of items in the data set into a page. The top level tags in these tag sets are:

- **Repeater**
- **CellRepeater**

Inside the bodies of each of these tags, other NetUI tags can be placed. In general, the body of a repeating tag is rendered once for each of the items in the data set, and the child tags can bind to properties on the "current" item by using the expression context `container` and referencing the current item with the syntax `container.item`. The **Repeater** tag can also be used in a more structured way where it can render a header, item, and footer where the item is rendered once for each item in the data set. Many different types of data sets can be rendered by these tags including Java Collections, Maps, and arrays. For each of these tags, the `dataSource` attribute is used to reference a data set to render.

A final repeating, databinding tag set is the **Grid** tag set, which is used to render data from a `javax.sql.RowSet` object. A RowSet is a disconnected data set that contains data from a relational database, and the **Grid** tag set provides the ability to page, sort, and filter this data. The **Grid** tag does not render its item using other NetUI HTML tags, rather the **Grid** renders using its child **Columns** and specific column tags that render UI for each cell of each record in the RowSet.

The **Repeater** and **Grid** tags are complex tags that render their bodies many times but only render part of their bodies each time. These tags use a state machine and a "render state" that is exposed to tags that are members of the **Repeater** tag set and **Grid** tag sets respectively. Tags that understand these states render in the appropriate state in order to produce the markup that is written to a JSP pages.

## Class Summary

### SortFilterService

The `SortFilterService` is used to keep track of sorts and filters query parameters that are stored on the URL.

# SortFilterService Class

*public final class SortFilterService*

extends Object

The SortFilterService is used to keep track of sorts and filters query parameters that are stored on the URL. These terms are used to create a DatabaseFilter which can be passed to a DatabaseControl to parameterize a SQL query.

The SortFilterService can be used as a storage location for parameters that need to be stored on the URL but may not be encoded in every anchor on every page in a Java Page Flow.

The SortFilterService is used to encode several bits of state for a specific grid onto the URL including:

- the current page
- sorts
- filters

This state should be specific to a given grid and is namespaced by the grid's name attribute. Multiple grids that may have parameters on the URL at the same time should have unique names.

In general, the methods on this class are used to manipulate the query parameters on URLs given the query parameters on the current request. Starting with the request's parameter map, additional parameters may be added or removed in order to build a map of values that can be attached to a given anchor or form that will render as the result of the current request. For example, a sort added to a column using the `SortFilterService.getQueryParamsMapWithSort(String, String)` could be available in an anchor that is rendered at the end of this request.

## Hierarchy

```
Object
  SortFilterService
```

## Field Summary

```
public
static PRIMARY_KEY
final String           The request key for a row identifier.

public
static REQUEST_KEY
final String           Key used to store a SortFilterService object in the request.

public
static ROW_IDENTIFIER
final String
```

The request key for a row identifier.

```

public
static SORT_ASCENDING
final int
    Constant representing an ascending sort applied to a column.

public
static SORT_DESCENDING
final int
    Constant representing a descending sort applied to a column.

public
static SORT_NONE
final int
    Constant representing no sorts applied to a column.

```

## Method Summary

```

public void
    addRowIdentifier(String gridName, String columnName, Object
    value)
        Add a row identifier which is accessible using the
        SortFilterService.getRowIdQueryParamsMap( ) method.

public static
String decodeUniqueIdentifier(HttpServletRequest request, String
    gridName, String columnName)
    Decode a value from the given request's query parameters.

public static
String encodeUniqueIdentifier(String gridName, String columnName,
    Object columnValue)
    Encode a grid/column value constraint.

public int
    getCurrentPage(String gridName)
        Get the current page for a specific grid.

    public
DatabaseFilter getDatabaseFilter(String gridName)
    Gets a DatabaseFilter object that contains the sorts and filters on the
    current request's URL.

public static
SortFilterService getInstance(Object object)
    Factory method to create a SortFilterService object.

public int
    getPageSize()
        Get the default page size.

public Map
    getQueryParamsMap()
        Get a Map of the query parameters that are on the current URL.

public Map
    getQueryParamsMapWithPage(String gridName, int page)
        Get a Map of query parameters that contains the request's parameters and a
        new parameter that sets the current page for the grid named gridName.

```

## Page Flow API Reference

```
public Map getQueryParamsMapWithSort(String gridName, String columnName)
    Get a Map of query parameters that contains the request's parameters and a
    new parameter that sets a new sort on some column of a grid.

public Map
    getRowIdQueryParamsMap()
    Get the query params map that contains only the row identifiers that should
    be encoded on the URL.

public int
    getSortDirection(String gridName, String columnName)
    Get the sort direction of a column in a grid.

public boolean
    isFiltered(String gridName, String columnName)
    Check the filter state of a given column in a grid for the current request.

public boolean
    isSorted(String gridName, String columnName)
    Check the sort state of a given column in a grid for the current request.

public void
    reset()
    Clears all query information including the current page, sorts, and filters
    for all grid names that are on the URL.

public void
    reset(String gridName)
    Clears the current page, sorts, and filters for the grid named gridName.

public void
    resetCurrentPage(String gridName)
    Reset the current page of a specific grid.

public String
    toString()
    Utility method that prints the current state of a SortFilterService
    instance.

public void
    updateRowId(ServletRequest request)
    Update the row identifiers given the current request.
```

### Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `wait`,  
`wait`, `wait`

## Field Detail

### PRIMARY\_KEY

```
public static final String PRIMARY_KEY
```

The request key for a row identifier. This identifier is deprecated; use  
`SortFilterService.ROW_IDENTIFIER` instead.

---

## REQUEST\_KEY

```
public static final String REQUEST_KEY
```

Key used to store a SortFilterService object in the request.

---

## ROW\_IDENTIFIER

```
public static final String ROW_IDENTIFIER
```

The request key for a row identifier.

---

## SORT\_ASCENDING

```
public static final int SORT_ASCENDING
```

Constant representing an ascending sort applied to a column.

---

## SORT\_DESCENDING

```
public static final int SORT_DESCENDING
```

Constant representing a descending sort applied to a column.

---

## SORT\_NONE

```
public static final int SORT_NONE
```

Constant representing no sorts applied to a column.

## Method Detail

### addRowIdentifier(String, String, Object) Method

```
public void addRowIdentifier(String gridName,
                             String columnName,
                             Object value)
```

Add a row identifier which is accessible using the `SortFilterService.getRowIdQueryParamsMap()` method. Row identifiers added for a particular grid name can be used in an Anchor or a Form to select a particular row to display on the next page.

**Parameters***gridName*

the name of the grid to which the row identifier is scoped

*columnName*

the name of the column that the identifier constrains

*value*

the value of the constraint on the row identifier.

**decodeUniqueIdentifier(HttpServletRequest, String, String) Method**

```
public static String decodeUniqueIdentifier(HttpServletRequest request,
                                           String gridName,
                                           String columnName)
```

Decode a value from the given request's query parameters. If the grid and column names are present in the row identifiers in the query parameters, this returns a String representation of the value's constraint.

**Parameters***request*

the request from which to find the unique identifier

*gridName*

the name of the grid whose unique identifiers to check

*columnName*

the name of the column whose constraint value to find

**Returns***the String representation of a value to which the given column is constrained***encodeUniqueIdentifier(String, String, Object) Method**

```
public static String encodeUniqueIdentifier(String gridName,
                                           String columnName,
                                           Object columnValue)
```

Encode a grid/column value constraint. This method formats the grid/column/value group into a syntax that can be used as the value of a query parameter on a URL.

**Parameters***gridName*

the grid name for the encoded identifier

*columnName*

the column name for the encoded identifier

*columnValue*

the value to encode

## Returns

*an encoded row identifier*

---

## getCurrentPage(String) Method

```
public int getCurrentPage(String gridName)
```

Get the current page for a specific grid.

## Parameters

*gridName*

the name of a grid whose current page may appear in the current query parameters

## Returns

*if available, the current page; zero otherwise*

---

## getDatabaseFilter(String) Method

```
public DatabaseFilter getDatabaseFilter(String gridName)
```

Gets a DatabaseFilter object that contains the sorts and filters on the current request's URL.

The gridName is used to locate specific URL parameters that are associated with a particular Grid tag.

## Parameters

*gridName*

the name of a grid on the page

## Returns

*a DatabaseFilter object that can be passed to a DatabaesControl instance to receive a sorted and filtered query result. Returns null if the gridName is null.*

---

## getInstance(Object) Method

```
public static SortFilterService getInstance(Object object)
```

Factory method to create a `SortFilterService` object. Usually, the object passed here is an `HttpServletRequest`.

### Parameters

*object*

the object from which to create the `SortFilterService`, usually the `HttpServletRequest`

### Returns

*a `SortFilterService` for use during the current request.*

---

## getPageSize() Method

```
public int getPageSize()
```

Get the default page size.

### Returns

*the default size of a page, 10.*

---

## getQueryParamsMap() Method

```
public Map getQueryParamsMap()
```

Get a `Map` of the query parameters that are on the current URL. In general, this method is used to pass these parameters from page to page on an anchor or inside of a form.

### Returns

*a `Map` of the current query parameters*

---

## getQueryParamsMapWithPage(String, int) Method

```
public Map getQueryParamsMapWithPage(String gridName,  
                                     int page)
```

Get a `Map` of query parameters that contains the request's parameters and a new parameter that sets the current page for the grid named `gridName`.

If the grid specified here already has a current page query parameter, it is replaced with the new value.

### Parameters

*gridName*

the name of the grid for which to set the current page



*page*

the page to display

## Returns

*a Map containing all of the current request's parameters and a new parameter that sets the page for the given grid.*

---

## getQueryParamsMapWithSort(String, String) Method

```
public Map getQueryParamsMapWithSort(String gridName,
                                     String columnName)
```

Get a Map of query parameters that contains the request's parameters and a new parameter that sets a new sort on some column of a grid.

The criteria for creating a sort are these:

- A sort on an already sorted column reverses the direction of the sort
- Only two columns may be sorted at any given time. The most recently clicked column is the first sort.
- When two columns are sorted, Sorting on the most recently sorted column removes the second sorted column.

## Parameters

*gridName*

the name of the grid for which to add a sort

*columnName*

the name of the column to sort

## Returns

*a Map containing all of the current request's parameters and a new parameter that sorts the grid on the given column.*

---

## getRowIdQueryParamsMap() Method

```
public Map getRowIdQueryParamsMap()
```

Get the query params map that contains only the row identifiers that should be encoded on the URL.

## Returns

*a Map that contains the query parameters which contain the row identifiers that uniquely represent a row in a data set.*

---

## getSortDirection(String, String) Method

```
public int getSortDirection(String gridName,  
                           String columnName)
```

Get the sort direction of a column in a grid. This method checks the current request's parameters and returns an identifier for the sort direction of the column.

### Parameters

*gridName*  
the name of the grid whose column to check

*columnName*  
the name of the column

### Returns

*SortFilterService.SORT\_ASCENDING* if the column is sorted in ascending order,  
*SortFilterService.SORT\_DESCENDING* if the column is sorted in descending order, and  
*SortFilterService.SORT\_NONE* if the column is not sorted.

---

## isFiltered(String, String) Method

```
public boolean isFiltered(String gridName,  
                         String columnName)
```

Check the filter state of a given column in a grid for the current request.

### Parameters

*gridName*  
the name of the grid whose column to check

*columnName*  
the name of the column

### Returns

*true* if the column is filtered for the grid; *false* otherwise.

---

## isSorted(String, String) Method

```
public boolean isSorted(String gridName,  
                       String columnName)
```

Check the sort state of a given column in a grid for the current request.

## Parameters

*gridName*  
the name of the grid whose column to check

*columnName*  
the name of the column

## Returns

*If this column in the given grid is sorted in the current URL, return true; otherwise, return false.*

---

## reset() Method

```
public void reset()
```

Clears all query information including the current page, sorts, and filters for all grid names that are on the URL. Future calls to methods that return query parameter maps will not contain any of these parameters from the current request.

---

## reset(String) Method

```
public void reset(String gridName)
```

Clears the current page, sorts, and filters for the grid named `gridName`. This will leave all pages, sorts, and filters for all other grid names.

## Parameters

*gridName*  
the name of the grid whose current page, sorts, and filters to reset.

---

## resetCurrentPage(String) Method

```
public void resetCurrentPage(String gridName)
```

Reset the current page of a specific grid. The page is reset to the first grid page.

## Parameters

*gridName*  
the name of the grid whose page to reset

---

## toString() Method

```
public String toString()
```

Utility method that prints the current state of a `SortFilterService` instance.

### *Overrides*

```
Object.toString()
```

### Returns

*a String representing the state*

---

## updateRowId(ServletRequest) Method

```
public void updateRowId(ServletRequest request)
```

Update the row identifiers given the current request.

### Parameters

*request*

the current request object

# com.bea.wlw.netui.tags.databinding.bundle Package

## Class Summary

BundleContext.BundleNode

BundleContext.NetUIBundleNode

BundleContext.StrutsBundleNode

DeclareBundle

A tag used to declare a `ResourceBundle` that can be used inside of the declaring page as a source for displaying internationalized messages.

# BundleContext.BundleNode Class

*public abstract static class **BundleContext.BundleNode***

extends Object

---

## **Hierarchy**

Object  
BundleContext.BundleNode

## **Direct Known Subclasses**

BundleContext.NetUIBundleNode, BundleContext.StrutsBundleNode

---

## Field Summary

protected String  
    **bundleName**  
        String  
protected Locale  
    **locale**  
        Locale

## Method Summary

public  
abstract boolean **containsKey**(String key)  
public  
abstract Enumeration **getKeys**()  
public  
abstract String **getString**(String key)  
public  
void **init**()

## Methods from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Field Detail

### bundleName

```
protected String bundleName
```

---

### locale

```
protected Locale locale
```

## Method Detail

### containsKey(String) Method

```
public abstract boolean containsKey(String key)
```

---

### getKeys() Method

```
public abstract Enumeration getKeys()
```

---

### getString(String) Method

```
public abstract String getString(String key)
```

---

### init() Method

```
public void init()
```

# BundleContext.NetUIBundleNode Class

*public static class BundleContext.NetUIBundleNode*

extends BundleContext.BundleNode

---

## Hierarchy

Object  
BundleContext.BundleNode  
BundleContext.NetUIBundleNode

---

## Field Summary

Fields from com.bea.wlw.netui.tags.databinding.bundle.BundleContext.BundleNode

bundleName, locale

## Method Summary

```
public
boolean containsKey(String key)
public
Enumeration getKeys()
public
String getString(String key)
public
void init()
```

Methods from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Method Detail

### containsKey(String) Method

```
public boolean containsKey(String key)
```

#### Overrides

```
BundleContext.BundleNode.containsKey(String)
```



## getKeys() Method

```
public Enumeration getKeys()
```

### *Overrides*

```
BundleContext.BundleNode.getKeys()
```

---

## getString(String) Method

```
public String getString(String key)
```

### *Overrides*

```
BundleContext.BundleNode.getString(String)
```

---

## init() Method

```
public void init()
```

### *Overrides*

```
BundleContext.BundleNode.init()
```

# BundleContext.StrutsBundleNode Class

*public static class **BundleContext.StrutsBundleNode***

extends BundleContext.BundleNode

---

## Hierarchy

Object  
BundleContext.BundleNode  
BundleContext.StrutsBundleNode

---

## Field Summary

Fields from com.bea.wlw.netui.tags.databinding.bundle.BundleContext.BundleNode

bundleName, locale

## Method Summary

public  
boolean **containsKey**(String key)  
public  
Enumeration **getKeys**()  
public  
String **getString**(String key)

### Methods from

com.bea.wlw.netui.tags.databinding.bundle.BundleContext.BundleNode  
init

### Methods from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll,  
toString, wait, wait, wait

## Method Detail

## **containsKey(String) Method**

```
public boolean containsKey(String key)
```

### *Overrides*

```
BundleContext.BundleNode.containsKey(String)
```

---

## **getKeys() Method**

```
public Enumeration getKeys()
```

### *Overrides*

```
BundleContext.BundleNode.getKeys()
```

---

## **getString(String) Method**

```
public String getString(String key)
```

### *Overrides*

```
BundleContext.BundleNode.getString(String)
```

## DeclareBundle Class

*public class **DeclareBundle***

extends `AbstractBaseTag`

A tag used to declare a `ResourceBundle` that can be used inside of the declaring page as a source for displaying internationalized messages. The declared resource bundle is accessible using the `bundle` databinding context. The required `name` attribute specifies the identifier via which to refer to the `ResourceBundle` in an expression. For example:

```
<netui-data:declareBundle name="someMessages" bundlePath="com/foobar/resources/WebAppMessages"
```

This tag declares a bundle that is referenced in a data binding expression as `{bundle.someMessages}`.

The bundle that is referenced depends on the `java.util.Locale` specified. The resource bundle properties files that are accessed are located in the package `com/foobar/resources` with the root properties file name of `WebAppMessages`. The naming conventions for properties can be found in Sun's Java documentation at `ResourceBundle.getBundle(String, Locale, ClassLoader)`. These files must be located in a classpath that is available to the web application. Often, they are stored in `WEB-INF/classes`. If the properties file contains a key called `helloWorld`, then the expression `{bundle.someMessages.helloWorld}` would look-up the message matching the `Locale` specified on the tag. Bundle binding expressions can be used in any data bindable `<netui...>` tag attribute.

It is possible to have keys that contain multiple words separated by spaces, commas, or periods. If this is the case, then you must use slightly different syntax to reference those keys in your data binding statement. The following list illustrates three ways to access the key `My helloWorld` from the `someMessages` property file:

- `<netui:label value="{bundle.someMessages['My helloWorld']}" />`
- `<netui:label value='{bundle.someMessages["My helloWorld"]}' />`
- `<netui:label value="{bundle.someMessages[\"My helloWorld\"]}" />`

WebLogic Workshop automatically generates the last example listed above when you set the label control in the property editor to `bundle.someMessages["My helloWorld"]`. In all three cases, you WebLogic Workshop can successfully retrieve the multi-word key.

**Note:** the name `default` is a bundle identifier that is reserved for use by the `DeclareBundle` tag. If this value is used for the `name` attribute on a `DeclareBundle` tag, an error will be reported in the page. The `default` bundle is reserved for use when accessing internationalized messages from the "current" Struts module's default properties file.

This tag provides a high level of customizability for rendering internationalized messages. Specifically, the `Locale` for which to look-up messages can be specified on the `DeclareBundle` tag. By default, the `Locale` for the current request is used, but this `Locale` can be overridden by setting the `language`, `country`, and `variant` tag attributes as necessary. See `Locale` for more information on the possible values for these attributes.

The Locale can be overridden by setting these attributes in three combinations:

- country, language, variant
- country, language
- country

Any other combinations will throw an exception.

Attribute Descriptions			
Attribute	Required	Runtime Expression Evaluation	Data Bindable
<b>bundlePath</b>	No	No	No
	<p>The path to a package in classpath that contains a set of <code>ResourceBundle</code> properties files. This can be a slash or dot delimited classpath. Valid values might be:</p> <ul style="list-style-type: none"> <li>• <code>com/foobar/resources/WebAppProperties</code></li> <li>• <code>com.foobar.resources.WebAppProperties</code></li> </ul> <p>These are treated as equivalent values. The <code>ResourceBundle</code> class will handle appending the <code>.properties</code> file type and locale information as necessary. For details about the <code>.properties</code> file naming conventions, see <code>ResourceBundle.getBundle(String, Locale, ClassLoader)</code>. These files must be available in classpath for the webapp in order to be successfully located.</p>		
<b>country</b>	No	No	Yes
	<p>The two-letter uppercase ISO-3166 country / region code for the Locale from which to look-up resource bundle messages. This value is used to further specify the name of the <code>.properties</code> file from which message keys will be read.</p>		
<b>language</b>	No	No	Yes
	<p>The two-letter lowercase ISO-639 language code for the Locale from which to look-up resource bundle messages. This value is used to further specify the name of the <code>.properties</code> file from which message keys will be read.</p>		
<b>name</b>	No	No	No
	<p>The name inside of the <i>bundle</i> databinding context under which the properties in this bundle are available. The identifier <code>default</code> is an illegal value for this attribute and is reserved for use by this tag.</p>		
<b>variant</b>	No	No	Yes
	<p>A vendor / browser specific code for further parameterizign the Locale from which to look-up resource bundle messages.</p>		

**Related Topics**

ResourceBundle  
Locale

---

***Hierarchy***

```

Object
  TagSupport
    BodyTagSupport
      AbstractBaseTag
        DeclareBundle

```

***All Implemented Interfaces***

BodyTag, IterationTag, Serializable, Tag

---

## Field

### Summary

**Fields from `com.bea.wlw.netui.tags.AbstractBaseTag`**

ATTR\_GENERAL, ATTR\_GENERAL\_EXPRESSION, ATTR\_JAVASCRIPT, ATTR\_STYLE, CLASS, ID, JAVASCRIPT\_STATUS, NAME, NETUI\_UNIQUE\_CNT, ONCLICK, ONDBLCLICK, ONKEYDOWN, ONKEYPRESS, ONKEYUP, ONMOUSEDOWN, ONMOUSEMOVE, ONMOUSEOUT, ONMOUSEOVER, ONMOUSEUP, STYLE, TABINDEX

**Fields from `javax.servlet.jsp.tagext.BodyTagSupport`**

bodyContent

**Fields from `javax.servlet.jsp.tagext.TagSupport`**

id, pageContext

**Fields from interface `javax.servlet.jsp.tagext.BodyTag`**

EVAL\_BODY\_BUFFERED, EVAL\_BODY\_TAG

**Fields from interface `javax.servlet.jsp.tagext.IterationTag`**

EVAL\_BODY\_AGAIN

**Fields from interface `javax.servlet.jsp.tagext.Tag`**

EVAL\_BODY\_INCLUDE, EVAL\_PAGE, SKIP\_BODY, SKIP\_PAGE

## Constructor

### Summary

*DeclareBundle()*

## Method

### Summary

```

public
    int doEndTag()
        Register a ResourceBundle that is available for the scope of this page.

public
    int doStartTag()
        Start the JSP rendering lifecycle for this tag; it skips its body.

public
    String getBundlePath()
        Get the path to the resource bundle's properties files.

public
    String getCountry()
        Get the country to use when looking-up resource bundle messages.

public
    String getLanguage()
        Get the language to use when looking-up resource bundle messages.

public
    String getName()
        Get the name via which this bundles messages can be accessed using the bundle
        databinding context.

public
    String getTagName()
        Get the name of this tag.

public
    String getVariant()
        Get the variant to use when looking-up resource bundle messages.

protected
    void localRelease()
        Reset all of the fields of this tag.

public
    void setBundlePath(String bundlePath)
        Set the path to the resource bundle's properties files.

public
    void setCountry(String country)
        Set the country to use when looking-up resource bundle messages.

public
    void setLanguage(String language)
        Set the language to use when looking-up resource bundle messages.

```

```
public setName(String name)
    void          Set the name via which this bundle can be accessed using the bundle databinding
                  context.

public
    void setVariant(String variant)
        Set the vendor / browser specific code that further parameterizes the .properties file
        messages will be read from.
```

### Methods from `com.bea.wlw.netui.tags.AbstractBaseTag`

```
addTagIdMapping, evaluateAttributeToString, filter, filter,
getJavaScriptUtils, getNearestForm, getScriptReporter, getUserLocale,
registerTagError, release, renderAttribute, rewriteName, write
```

### Methods from `javax.servlet.jsp.tagext.BodyTagSupport`

```
doAfterBody, doInitBody, getBodyContent, getPreviousOut, release,
setBodyContent
```

### Methods from `javax.servlet.jsp.tagext.TagSupport`

```
doAfterBody, doEndTag, doStartTag, findAncestorWithClass, getId,
getParent, getValue, getValues, release, removeValue, setId,
setPageContext, setParent, setValue
```

### Methods from class `java.lang.Object`

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll,
toString, wait, wait, wait
```

### Methods from interface `javax.servlet.jsp.tagext.BodyTag`

```
doInitBody, setBodyContent
```

### Methods from interface `javax.servlet.jsp.tagext.IterationTag`

```
doAfterBody
```

### Methods from interface `javax.servlet.jsp.tagext.Tag`

```
doEndTag, doStartTag, getParent, release, setPageContext, setParent
```

## Constructor Detail

### DeclareBundle

```
public DeclareBundle()
```



## Method Detail

### doEndTag() Method

```
public int doEndTag()  
    throws JspException
```

Register a ResourceBundle that is available for the scope of this page. Errors raised during the execution of this tag will be reported in the page, which will continue rendering.

#### *Overrides*

```
BodyTagSupport.doEndTag( )
```

#### Returns

*Tag.EVAL\_PAGE*

#### Exceptions

*JspException*

if error conditions are encountered during this method which can not be reported in the page.

---

### doStartTag() Method

```
public int doStartTag()
```

Start the JSP rendering lifecycle for this tag; it skips its body.

#### *Overrides*

```
BodyTagSupport.doStartTag( )
```

#### Returns

*Tag.SKIP\_BODY*

---

### getBundlePath() Method

```
public String getBundlePath()
```

Get the path to the resource bundle's properties files.

#### Returns

*the bundle path*

---

## getCountry() Method

```
public String getCountry()
```

Get the country to use when looking-up resource bundle messages.

### Returns

*the country code*

---

## getLanguage() Method

```
public String getLanguage()
```

Get the language to use when looking-up resource bundle messages.

### Returns

*the language code*

---

## getName() Method

```
public String getName()
```

Get the name via which this bundles messages can be accessed using the bundle databinding context.

### Returns

*String the name of the bundle*

---

## getTagName() Method

```
public String getTagName()
```

Get the name of this tag. This is used to identify the type of this tag for reporting tag errors.

### Overrides

```
AbstractBaseTag.getTagName()
```

### Returns

*a constant String representing the name of this tag.*

---

## getVariant() Method

```
public String getVariant()
```

Get the variant to use when looking-up resource bundle messages.

### Returns

*the variant*

---

## localRelease() Method

```
protected void localRelease()
```

Reset all of the fields of this tag.

### Overrides

```
AbstractBaseTag.localRelease()
```

---

## setBundlePath(String) Method

```
public void setBundlePath(String bundlePath)
```

Set the path to the resource bundle's properties files. This can be a slash or dot delimited classpath. Valid values might be:

- com/foobar/resources/WebAppProperties
- com.foobar.resources.WebAppProperties

These are treated as equivalent values. The `ResourceBundle` class will handle appending the `.properties` file type and locale information as necessary. See `ResourceBundle.getBundle(String, Locale, ClassLoader)` for the `.properties` file naming conventions. These files must be available in classpath for the webapp in order to be successfully located.

### Parameters

*bundlePath*

the path to the bundle's properties files.

---

## setCountry(String) Method

```
public void setCountry(String country)
```

Set the country to use when looking-up resource bundle messages. This value is used to further specify the name of the `.properties` file message keys will be read from.

## Parameters

*country*

the two-letter uppercase ISO-3166 code for a country

---

## setLanguage(String) Method

```
public void setLanguage(String language)
```

Set the language to use when looking-up resource bundle messages. This value is used to further specify the name of the .properties file message keys will be read from.

## Parameters

*language*

the two-letter lowercase ISO-639 code for a language.

---

## setName(String) Method

```
public void setName(String name)
```

Set the name via which this bundle can be accessed using the `bundle` databinding context. The identifier `default` is an illegal value for this attribute and is reserved for use by this tag.

## Parameters

*name*

the name of the bundle

---

## setVariant(String) Method

```
public void setVariant(String variant)
```

Set the vendor / browser specific code that further parameterizes the .properties file messages will be read from.

## Parameters

*variant*

the variant

---

# **com.bea.wlw.netui.tags.databinding.cellrepeater**

## **Package**

### **Class Summary**

#### **CellRepeater**

The CellRepeater tag is a repeating, databound tag that renders its body each cell of a table of the specified dimensions.

## CellRepeater Class

*public class CellRepeater*

extends DataSourceTag  
implements DataAccessProvider

The CellRepeater tag is a repeating, databound tag that renders its body each cell of a table of the specified dimensions. The tag is bound to a dataset that is specified in the `dataSource` attribute. For each item in the data set, the body of this tag is rendered, and NetUI tags in the body that are databound can use the `container.item` syntax to access the "current" data item in the iteration. Properties on this data item can be accessed using expressions.

The tag will automatically insert the open and close table, row, and cell tags which will comprise the table that is rendered. Style attributes may be set using attributes on this tag in order to customize the tag's final appearance. The dimensions of the table are specified by using at least one of the attributes `columns` and `rows`. If only one is specified, the other will be inferred by using the size of the given data set. As a result, the entire dataset will be rendered. For example, if a table should be four columns wide and the data set has twenty items, the resulting table will have five rows. If the data set is fewer items than the number of cells that should be rendered, the cells are padded with HTML table cells:

```
<td>&nbsp;</td>
```

This will prevent rendering a malformed HTML table. If the number of cells to render is smaller than the fully specified dimensions of the table, only this number of cells will be rendered. For example, if the data set is size fifty but the `rows` and the `columns` attributes are both seven, only the first forty-nine items in the dataset will be rendered and the fiftieth will not be shown. The values of the `rows` and the `columns` can be databound with an expression; in this case, each value will be converted into an integer. An error will be reported on the page if this conversion fails.

This tag implements the `DataAccessProvider` interface which provides tags access to the "current" data item. Properties on the `DataAccessProvider` interface are available through the "container" binding context, which can be used inside of the body of the CellRepeater. Properties of the `DataAccessProvider` interface that are available include:

Name	Description
index	the current index in the iteration; this index is absolute to the dataset
parent	any <code>DataAccessProvider</code> parent of this tag
item	the current data item

**Note:** the metadata property of the `container` binding context is not supported on the CellRepeater.

<b>Attribute Descriptions</b>
-------------------------------

Attribute	Required	Runtime Expression Evaluation	Data Bindable
<b>alternatingCellClass</b>	No	No	No
	<p>The CSS style class to render on every other generated table cell; the cell class is used if this attribute is not set. The starting cell is alternated for each row, which results in a checkerboard colored table being displayed. For example, if the alternating cell class value is <code>"alternatingCellClass"</code>, every other table cell tag in the rendered JSP is:</p> <pre>&lt;td class="alternatingCellClass"&gt;</pre>		
<b>cellClass</b>	No	No	No
	The CSS style class to render on generated table cells.		
<b>columns</b>	No	No	Yes
	A data bindable attribute that specifies how many columns should appear in the generated table; the value should either be a <code>String</code> which can be converted into an expression that references an <code>int</code> or <code>Integer</code> . If the <code>columns</code> attribute was specified but the <code>rows</code> attribute was not, the <code>rows</code> attribute is inferred using the size of the dataset.		
<b>dataSource</b>	No	No	Yes
	An expression that references a dataset that should be rendered by the <code>cellRepeater</code> . If there was no data in the dataset but the <code>rows</code> and <code>columns</code> attributes were specified, an empty table of the given dimensions is rendered.		
<b>rowClass</b>	No	No	No
	The CSS style class to render on every generated table row.		
<b>rows</b>	No	No	Yes
	A data bindable attribute that specifies how many rows should appear in the generated table; the value should either be a <code>String</code> which can be converted into an integer or an expression that references an <code>int</code> or <code>Integer</code> . If the <code>rows</code> attribute was specified but the <code>columns</code> attribute was not, the <code>columns</code> attribute is inferred using the size of the dataset.		
<b>tableClass</b>	No	No	No

	The CSS style class to render on the generated table.		
<b>verticalRepeat</b>	No	No	No
	The boolean that specifies in which direction to repeat the cells, horizontally or vertically. If the rows are rendered horizontally, the items in the dataset are rendered across each row from top to bottom. Otherwise, they are rendered down each column from left to right. The default is to render the items horizontally.		

## Related Topics

<netui-data:cellRepeater> Tag Sample

---

### *Hierarchy*

```

Object
  TagSupport
    BodyTagSupport
      AbstractBaseTag
        DataSourceTag
          CellRepeater

```

### *All Implemented Interfaces*

BodyTag, DataAccessProvider, IterationTag, Serializable, Tag

---

## Field Summary

### Fields from `com.bea.wlw.netui.tags.AbstractBaseTag`

ATTR\_GENERAL, ATTR\_GENERAL\_EXPRESSION, ATTR\_JAVASCRIPT, ATTR\_STYLE, CLASS, ID, JAVASCRIPT\_STATUS, NAME, NETUI\_UNIQUE\_CNT, ONCLICK, ONDBLCLICK, ONKEYDOWN, ONKEYPRESS, ONKEYUP, ONMOUSEDOWN, ONMOUSEMOVE, ONMOUSEOUT, ONMOUSEOVER, ONMOUSEUP, STYLE, TABINDEX

### Fields from `javax.servlet.jsp.tagext.BodyTagSupport`

bodyContent

### Fields from `javax.servlet.jsp.tagext.TagSupport`

id, pageContext

### Fields from interface `javax.servlet.jsp.tagext.BodyTag`

EVAL\_BODY\_BUFFERED, EVAL\_BODY\_TAG



**Fields from interface javax.servlet.jsp.tagext.IterationTag**

EVAL\_BODY\_AGAIN

**Fields from interface javax.servlet.jsp.tagext.Tag**

EVAL\_BODY\_INCLUDE, EVAL\_PAGE, SKIP\_BODY, SKIP\_PAGE

## Constructor Summary

*CellRepeater()*

## Method Summary

```

    public void
        addContent(String content)
            Add content to the content that is being buffered by this
            tag.

    public int
        doAfterBody()
            Continue rendering the body of this tag until the
            dimensions of the table have been reached or the entire
            dataset has been rendered.

    public int
        doEndTag()
            Complete rendering the tag.

    public int
        doStartTag()
            Prepare to render the dataset that was specified in the
            dataSource attribute.

    public String
        getAlternatingCellClass()
            Get the HTML style class for every other table cell that
            is rendered in the HTML table that is rendered by this
            tag.

    public String
        getCellClass()
            Get the HTML style class for a table cell that is rendered
            in the HTML table that is rendered by this tag.

    public String
        getColumns()
            Get the number of columns that should be rendered in
            the table generated by this tag.

    public String

```

## Page Flow API Reference

***getContent()***  
Get the content which should be rendered by this tag.

public int  
***getCurrentIndex()***  
Get the index of the current iteration through the body of this tag.

public Object  
***getCurrentItem()***  
Get the item that is currently being rendered by this repeating tag.

public Object  
***getCurrentMetadata()***  
Get the metadata for the current item.

public  
DataAccessProvider ***getProviderParent()***  
Get the parent DataAccessProvider for this tag.

public String  
***getRowClass()***  
Get the HTML style class for a table row that is rendered in the HTML table that is rendered by this tag.

public String  
***getRows()***  
Get the number of rows that should be rendered in the table generated by this tag.

public String  
***getTableClass()***  
Get the HTML style class for the table that is rendered in the HTML table that is opened by this tag.

public String  
***getTagName()***  
Get the name of this tag.

public boolean  
***getVerticalRepeat()***  
Get the repeat direction.

protected void  
***localRelease()***  
Reset all of the fields of this tag.

public void  
***setAlternatingCellClass(String alternatingCellClass)***  
Set the HTML style class that is rendered on each HTML table cell that is opened by this tag.

public void  
***setCellClass(String cellClass)***  
Set the HTML style class that is rendered on each HTML table cell.

public void  
***setColumns(String columnsString)***

## Page Flow API Reference

Set the number of columns that should be rendered in the table generated by the tag.

```
public void
    setRowClass(String rowClass)
        Set the HTML style class that is rendered on each
        HTML table row that is opened by this tag.

public void
    setRows(String rowsString)
        Set the number of rows that should be rendered in the
        table generated by the tag.

public void
    setTableClass(String tableClass)
        Set the HTML style class that is rendered on the HTML
        table that is opened by this tag.

public void
    setVerticalRepeat(boolean verticalRepeat)
        This tag can render the items in its dataset horizontally
        or vertically.
```

### Methods from `com.bea.wlw.netui.tags.DataSourceTag`

`getDataSource`, `setDataSource`

### Methods from `com.bea.wlw.netui.tags.AbstractBaseTag`

`addTagIdMapping`, `evaluateAttributeToString`, `filter`, `filter`,  
`getJavaScriptUtils`, `getNearestForm`, `getScriptReporter`, `getUserLocale`,  
`localRelease`, `registerTagError`, `release`, `renderAttribute`, `rewriteName`,  
`write`

### Methods from `javax.servlet.jsp.tagext.BodyTagSupport`

`doInitBody`, `getBodyContent`, `getPreviousOut`, `release`, `setBodyContent`

### Methods from `javax.servlet.jsp.tagext.TagSupport`

`doAfterBody`, `doEndTag`, `doStartTag`, `findAncestorWithClass`, `getId`,  
`getParent`, `getValue`, `getValues`, `release`, `removeValue`, `setId`,  
`setPageContext`, `setParent`, `setValue`

### Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`,  
`toString`, `wait`, `wait`, `wait`

### Methods from interface `javax.servlet.jsp.tagext.BodyTag`

`doInitBody`, `setBodyContent`

### Methods from interface `com.bea.wlw.netui.tags.DataAccessProvider`

`getCurrentIndex, getCurrentItem, getCurrentMetadata, getDataSource, getProviderParent, setDataSource`

#### Methods from interface `javax.servlet.jsp.tagext.IterationTag`

`doAfterBody`

#### Methods from interface `javax.servlet.jsp.tagext.Tag`

`doEndTag, doStartTag, getParent, release, setPageContext, setParent`

## Constructor Detail

### CellRepeater

```
public CellRepeater()
```

## Method Detail

### addContent(String) Method

```
public void addContent(String content)
```

Add content to the content that is being buffered by this tag. All content written by the body of this tag is added to this buffer. The buffer is rendered at the end of the tag's lifecycle if no fatal errors have occurred during this tag's lifecycle.

#### Parameters

*content*

content that this tag should render.

---

### doAfterBody() Method

```
public int doAfterBody()
```

Continue rendering the body of this tag until the dimensions of the table have been reached or the entire dataset has been rendered. The buffered body content from the previous iteration of the body is added to the content this tag will render, @see `addContent(java.lang.String)`. Pad the table if the dimensions have not been met but the dataset is empty.

#### Overrides

```
BodyTagSupport.doAfterBody()
```

**Returns**

*EVAL\_BODY\_BUFFERED* if there is more data to render in the dataset or *SKIP\_BODY* if the end of the dataset is reached or an error occurs

---

**doEndTag() Method**

```
public int doEndTag()
    throws JspException
```

Complete rendering the tag. If no errors have occurred, the content that the tag buffered is rendered.

**Overrides**

`BodyTagSupport.doEndTag()`

**Returns**

*EVAL\_PAGE* to continue evaluating the page

**Exceptions**

*JspException*  
if an error occurs that can not be reported on the page

---

**doStartTag() Method**

```
public int doStartTag()
    throws JspException
```

Prepare to render the dataset that was specified in the dataSource attribute. The dataSource expression is evaluated and the table's dimensions are computed. If there is no data in the dataset but the rows and columns attributes were specified, an empty table of the given dimensions is rendered.

**Overrides**

`BodyTagSupport.doStartTag()`

**Returns**

*EVAL\_BODY\_BUFFERED* or *SKIP\_BODY* if errors are reported, the data set is null, or there is no data in the data set

**Exceptions**

*JspException*  
if errors occurred that could not be reported in the page

---

## getAlternatingCellClass() Method

```
public String getAlternatingCellClass()
```

Get the HTML style class for every other table cell that is rendered in the HTML table that is rendered by this tag.

### Returns

*the name of the style class in a CSS or null if none is set*

---

## getCellClass() Method

```
public String getCellClass()
```

Get the HTML style class for a table cell that is rendered in the HTML table that is rendered by this tag.

### Returns

*the name of the style class in a CSS or null if none is set*

---

## getColumns() Method

```
public String getColumns()
```

Get the number of columns that should be rendered in the table generated by this tag.

### Returns

*a String of the integer or an expression that references an integer.*

---

## getContent() Method

```
public String getContent()
```

Get the content which should be rendered by this tag.

### Returns

*the content String if there is content; null otherwise*

---

## getCurrentIndex() Method

```
public int getCurrentIndex()
```

Get the index of the current iteration through the body of this tag. This data can be accessed using the

expression `container.index` on an attribute of a databindable NetUI tag that is contained within the repeating body of this tag. This expression is only valid when the dataset is being rendered.

### Returns

*the integer index of the current data item in the data set*

### Related Topics

DataAccessProvider

---

## getCurrentItem() Method

```
public Object getCurrentItem()
```

Get the item that is currently being rendered by this repeating tag. This can be accessed using the expression `expression.item` on an attribute of a databindable netUI tag that is contained within the repeating body of this tag. The expression is only valid when the dataset is being rendered.

### Returns

*the current item in the data set*

### Related Topics

DataAccessProvider

---

## getCurrentMetadata() Method

```
public Object getCurrentMetadata()
```

Get the metadata for the current item. This method is not supported by this tag.

### Returns

*the current metadata or null if no metadata can be found or metadata is not supported by a DataAccessProvider implementation*

### Related Topics

DataAccessProvider

---

## getProviderParent() Method

```
public DataAccessProvider getProviderParent()
```

Get the parent DataAccessProvider for this tag. If this tag is contained within a DataAccessProvider, the

containing `DataProvider` is available through the expression `container.container`. Any valid properties of the parent `DataProvider` can be accessed through this expression. This method will return null if there is no parent `DataProvider`

### Returns

*a containing `DataProvider` if one exists, null otherwise.*

### Related Topics

`DataProvider`

---

## `getRowClass()` Method

```
public String getRowClass()
```

Get the HTML style class for a table row that is rendered in the HTML table that is rendered by this tag.

### Returns

*the name of a style class in a CSS or null if none is set*

---

## `getRows()` Method

```
public String getRows()
```

Get the number of rows that should be rendered in the table generated by this tag.

### Returns

*a String of the integer or an expression*

---

## `getTableClass()` Method

```
public String getTableClass()
```

Get the HTML style class for the table that is rendered in the HTML table that is opened by this tag.

### Returns

*the name of a style class in a CSS or null if none is set*

---

## `getTagName()` Method

```
public String getTagName()
```



Get the name of this tag. This is used to identify the type of this tag for reporting tag errors.

**Overrides**

`AbstractBaseTag.getTagName()`

**Returns**

*a constant String representing the name of this tag.*

---

## **getVerticalRepeat() Method**

`public boolean getVerticalRepeat()`

Get the repeat direction. If the verticalRepeat attribute is set to true, the dataset is rendered down each column. Otherwise, it is rendered across each row.

**Returns**

*true if the dataset should be rendered down each column. The default is false.*

---

## **localRelease() Method**

`protected void localRelease()`

Reset all of the fields of this tag.

**Overrides**

`DataSourceTag.localRelease()`

---

## **setAlternatingCellClass(String) Method**

`public void setAlternatingCellClass(String alternatingCellClass)`

Set the HTML style class that is rendered on each HTML table cell that is opened by this tag. The starting cell is alternated for each row, which results in a checkerboard colored table being displayed. For example, if the alternating cell class is `"alternatingCellClass"`, every other table cell tag is:

```
<td cell="alternatingCellClass">
```

**Parameters**

*alternatingCellClass*

the name of a style class in a CSS

---

## setCellClass(String) Method

```
public void setCellClass(String cellClass)
```

Set the HTML style class that is rendered on each HTML table cell. For example, if the cell class is "cellClass", each opening table cell tag is:

```
<td class="cellClass">
```

### Parameters

*cellClass*

the name of a style class in a CSS

---

## setColumns(String) Method

```
public void setColumns(String columnsString)
```

Set the number of columns that should be rendered in the table generated by the tag. If the columns attribute is specified but the rows attribute is not, the rows attribute will be inferred using the size of the dataset.

### Parameters

*columnsString*

an integer or an expression

---

## setRowClass(String) Method

```
public void setRowClass(String rowClass)
```

Set the HTML style class that is rendered on each HTML table row that is opened by this tag. For example, if the row class is "rowClass", each opening table row tag is:

```
<tr class="rowClass">
```

### Parameters

*rowClass*

the name of a style class in the CSS

---

## setRows(String) Method

```
public void setRows(String rowsString)
```

Set the number of rows that should be rendered in the table generated by the tag. If the rows attribute is specified but the columns attribute is not, the columns attribute will be inferred using the size of the dataset.

## Parameters

*rowsString*

an integer or an expression whose value can be converted into an integer.

---

## setTableClass(String) Method

```
public void setTableClass(String tableClass)
```

Set the HTML style class that is rendered on the HTML table that is opened by this tag. For example, if the row class is "tableClass", each opening table tag is:

```
<table class="tableClass">
```

## Parameters

*tableClass*

the name of a style class in a CSS

---

## setVerticalRepeat(boolean) Method

```
public void setVerticalRepeat(boolean verticalRepeat)
```

This tag can render the items in its dataset horizontally or vertically. If the rows are rendered horizontally, the items in the dataset are rendered across each row from top to bottom. Otherwise, they are rendered down each column from left to right. The default is to render the items horizontally.

## Parameters

*verticalRepeat*

if set to `true`, the dataset is rendered down each column; otherwise it is rendered across each row, the default.

# com.bea.wlw.netui.tags.databinding.grid Package

## Class Summary

Columns	This tag is a container for the columns that will render the header, data, and footer for each column in a Grid.
Grid	A <netui-data:grid...
GridComponent	An abstract base class for all tags that can be contained directly inside of a Grid tag.
GridContext	A JavaBean that represents the runtime data of a specific Grid tag instance.

## Columns Class

*public class Columns*

extends GridComponent

This tag is a container for the columns that will render the header, data, and footer for each column in a Grid. Tags contained immediately within the Columns tag should all be of type GridColumn. Those tags are:

- AnchorColumn
- BasicColumn
- ExpressionColumn
- ImageColumn

Additionally, the Columns tag has the capability to set "global" properties for all of the contained GridColumn tags. Specifically, the ability to sort / filter and the actions associated with performing these operations can be set here. All sortable and filterable GridColumn tags that are contained within a Columns tag may use these values or may override them if a column needs to be handled specially.

Attribute Descriptions			
Attribute	Required	Runtime Expression Evaluation	Data Bindable
filterable	No	No	No
	A boolean that enables all of the contained GridColumn objects that are SortFilterColumn types to be filterable. Individual GridColumn tags can override this to enable / disable filtering.		
filterAction	No	No	No
	The action that will be executed if any filterable GridColumn tags are filtered. Individual GridColumn tags can override this to call a different action.		
sortable	No	No	No
	A boolean that enables all of the contained GridColumn objects that are SortFilterColumn types to be sortable. Individual GridColumn tags can override this to enable / disable sorting.		
sortAction	No	No	No
	The action that will be executed if any sortable GridColumn tags are sorted. Individual GridColumn tags can override this to call a different action.		

<b>styleClassPrefix</b>	No	No	No
	A style class prefix that will be applied to all cells in all columns inside of a GridColumns tag. This attribute provides a way to customize the header, data, and footer cells of all grid columns with a single attribute. See <code>GridColumn.setStyleClassPrefix(String)</code> for more information on how columns apply CSS styles to cells.		

**Related Topics**

<netui-data:grid> Tag Sample  
Grid

---

***Hierarchy***

```

Object
  TagSupport
    BodyTagSupport
      AbstractBaseTag
        StructuredBaseTag
          GridComponent
            Columns

```

***All Implemented Interfaces***

BodyTag, IterationTag, Serializable, Tag

---

## Field Summary

**Fields from `com.bea.wlw.netui.tags.databinding.base.StructuredBaseTag`**

EMPTY\_STRING

**Fields from `com.bea.wlw.netui.tags.AbstractBaseTag`**

ATTR\_GENERAL, ATTR\_GENERAL\_EXPRESSION, ATTR\_JAVASCRIPT, ATTR\_STYLE, CLASS, ID, JAVASCRIPT\_STATUS, NAME, NETUI\_UNIQUE\_CNT, ONCLICK, ONDBLCLICK, ONKEYDOWN, ONKEYPRESS, ONKEYUP, ONMOUSEDOWN, ONMOUSEMOVE, ONMOUSEOUT, ONMOUSEOVER, ONMOUSEUP, STYLE, TABINDEX

**Fields from `javax.servlet.jsp.tagext.BodyTagSupport`**

bodyContent

**Fields from `javax.servlet.jsp.tagext.TagSupport`**

id, pageContext

**Fields from interface javax.servlet.jsp.tagext.BodyTag**

EVAL\_BODY\_BUFFERED, EVAL\_BODY\_TAG

**Fields from interface javax.servlet.jsp.tagext.IterationTag**

EVAL\_BODY\_AGAIN

**Fields from interface javax.servlet.jsp.tagext.Tag**

EVAL\_BODY\_INCLUDE, EVAL\_PAGE, SKIP\_BODY, SKIP\_PAGE

## Constructor Summary

*Columns()*

## Method Summary

```

public
String getFilterAction()
    Get the action that will be invoked if a contained GridColumn tag is
    filtered.

public
String getSortAction()
    Get the action that will be invoked if a contained GridColumn tag is sorted.

public
String getStyleClassPrefix()
    Get the default style class prefix that is applied to all columns in the grid.

public
String getTagName()
    Get the name of this tag.

public
boolean isFilterable()
    Get whether or not this column can be filtered.

public
boolean isSortable()
    Get whether or not this column can be sorted.

protected
void localRelease()
    Reset all of the fields of this tag.

protected
int renderEndTag(int gridState)

protected
int renderStartTag(int gridState)

```

```

public setFilterable(boolean filterable)
    void          Set whether or not this column can be filtered.
public
    void setFilterAction(String filterAction)
        Set the action that will be invoked if a contained GridColumn tag is
        filtered.
public
    void setSortable(boolean sortable)
        Set whether or not this column can be sorted.
public
    void setSortAction(String sortAction)
        Set the action that will be invoked if a contained GridColumn tag is sorted.
public
    void setStyleClassPrefix(String styleClassPrefix)
        Set a default style class prefix that is applied to all columns in the grid.

```

**Methods from `com.bea.wlw.netui.tags.databinding.grid.GridComponent`**

`getCurrentRenderState`, `getGrid`, `getGridContext`, `getValidContainerType`,  
`prepare`

**Methods from `com.bea.wlw.netui.tags.databinding.base.StructuredBaseTag`**

`addContent`, `doEndTag`, `doStartTag`, `getContent`, `getCurrentRenderState`,  
`getValidContainerType`, `localRelease`, `prepare`, `verifyAttributes`,  
`verifyStructure`

**Methods from `com.bea.wlw.netui.tags.AbstractBaseTag`**

`addTagIdMapping`, `evaluateAttributeToString`, `filter`, `filter`,  
`getJavaScriptUtils`, `getNearestForm`, `getScriptReporter`, `getUserLocale`,  
`localRelease`, `registerTagError`, `release`, `renderAttribute`, `rewriteName`,  
`write`

**Methods from `javax.servlet.jsp.tagext.BodyTagSupport`**

`doAfterBody`, `doEndTag`, `doInitBody`, `doStartTag`, `getBodyContent`,  
`getPreviousOut`, `release`, `setBodyContent`

**Methods from `javax.servlet.jsp.tagext.TagSupport`**

`doAfterBody`, `doEndTag`, `doStartTag`, `findAncestorWithClass`, `getId`,  
`getParent`, `getValue`, `getValues`, `release`, `removeValue`, `setId`,  
`setPageContext`, `setParent`, `setValue`

**Methods from class `java.lang.Object`**

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`,  
`toString`, `wait`, `wait`, `wait`



**Methods from interface `javax.servlet.jsp.tagext.BodyTag`**

`doInitBody`, `setBodyContent`

**Methods from interface `javax.servlet.jsp.tagext.IterationTag`**

`doAfterBody`

**Methods from interface `javax.servlet.jsp.tagext.Tag`**

`doEndTag`, `doStartTag`, `getParent`, `release`, `setPageContext`, `setParent`

## Constructor Detail

### Columns

```
public Columns()
```

## Method Detail

### `getFilterAction()` Method

```
public String getFilterAction()
```

Get the action that will be invoked if a contained `GridColumn` tag is filtered.

#### Returns

*the action that will be called to sort a `GridColumn` inside of a `Columns` tag.*

---

### `getSortAction()` Method

```
public String getSortAction()
```

Get the action that will be invoked if a contained `GridColumn` tag is sorted.

#### Returns

*the action that will be called to sort a `GridColumn` inside of a `Columns` tag.*

---

### `getStyleClassPrefix()` Method

```
public String getStyleClassPrefix()
```

Get the default style class prefix that is applied to all columns in the grid. See `GridColumn.setStyleClassPrefix(String)` for more information.

### Returns

*the style class prefix that is used by all columns in a grid.*

---

## getTagName() Method

```
public String getTagName()
```

Get the name of this tag. This is used to identify the type of this tag for reporting tag errors.

### Overrides

```
AbstractBaseTag.getTagName()
```

### Returns

*a constant String representing the name of this tag.*

---

## isFilterable() Method

```
public boolean isFilterable()
```

Get whether or not this column can be filtered.

### Returns

*true if this column is filtered; false otherwise.*

---

## isSortable() Method

```
public boolean isSortable()
```

Get whether or not this column can be sorted.

### Returns

*true if this column is filtered; false otherwise.*

---

## localRelease() Method

```
protected void localRelease()
```

Reset all of the fields of this tag.

**Overrides**

`GridComponent.localRelease()`

---

**renderEndTag(int) Method**

`protected int renderEndTag(int gridState)`

**Overrides**

`StructuredBaseTag.renderEndTag(int)`

---

**renderStartTag(int) Method**

`protected int renderStartTag(int gridState)`

**Overrides**

`StructuredBaseTag.renderStartTag(int)`

---

**setFilterable(boolean) Method**

`public void setFilterable(boolean filterable)`

Set whether or not this column can be filtered. This attribute will help determine the capability of a contained `GridColumn` tags.

**Parameters**

*filterable*

a boolean that sets this capability. If `true` this column may be filterable; otherwise it is not.

---

**setFilterAction(String) Method**

`public void setFilterAction(String filterAction)`

Set the action that will be invoked if a contained `GridColumn` tag is filtered. The use of this attribute requires the presence of a `PageFlowController` in the JSP's directory. This attribute will help determine the sort action of a contained `GridColumn` tags.

**Parameters**

*filterAction*

the action that will be called to filter the column

---

## setSortable(boolean) Method

```
public void setSortable(boolean sortable)
```

Set whether or not this column can be sorted. This attribute will help determine the capability of a contained `GridColumn` tags.

### Parameters

*sortable*

a boolean that sets this capability. If `true` this column may be sortable; otherwise it is not.

---

## setSortAction(String) Method

```
public void setSortAction(String sortAction)
```

Set the action that will be invoked if a contained `GridColumn` tag is sorted. The use of this attribute requires the presence of a `PageFlowController` in the JSP's directory. This attribute will help determine the sort action of a contained `GridColumn` tags.

### Parameters

*sortAction*

the action that will be called to sort the column

---

## setStyleClassPrefix(String) Method

```
public void setStyleClassPrefix(String styleClassPrefix)
```

Set a default style class prefix that is applied to all columns in the grid. See `GridColumn.setStyleClassPrefix(String)` for more information. &@param `styleClassPrefix` a style class prefix that is used by all columns in a grid

## Grid Class

*public class* **Grid**

extends DataSourceTag

implements DataAccessProvider, ErrorReporter

A <netui-data:grid...> tag is the containing tag of a tag set that renders regular data with the ability to page, sort, and filter the data set. A grid renders a table that represents the data in a data set. In addition, metadata about the data set may be used to parameterize aspects of grid rendering.

The grid uses two objects to encapsulate its attributes and the data set that is rendered. These items are:

- GridContext
- DataContext

The former is used to encapsulate the attributes set on the Grid tag, while the latter is used to provide a consistent interface over different types of data sets. This enables the tags to interact with the same interface regardless of the type of data being displayed.

A grid renders a table in order to display the data. This table consists of three parts:

- header
- data rows
- footer

These parts are structured in a grid as follows:

column 1: header	column 2: header	column 3: header
column 1: data cell	column 2: data cell	column 3: data cell
column 1: data cell	column 2: data cell	column 3: data cell
column 1: data cell	column 2: data cell	column 3: data cell
column 1: footer	column 2: footer	column 3: footer

The grid is rendered either automatically or by nesting tags of type GridColumn tags inside of a Columns tag. The GridColumn tags are:

- AnchorColumn
- BasicColumn
- ExpressionColumn
- ImageColumn

If the grid is automatically rendered, all parts of the table above are rendered from the data set. The resulting Grid is not sortable or filterable. If the grid is rendered using its GridColumn, each column renders a single cell for each row. Regardless of how a grid is rendered, a row is rendered for each item in the data set, unless a Pager tag is used to set the page size.

## Page Flow API Reference

The types that are supported for rendering in the grid are `RowSet`. Any other types that are passed to the grid via an expression in the `dataSource` attribute will be illegal and an error will be reported in the JSP page.

Attribute Descriptions			
Attribute	Required	Runtime Expression Evaluation	Data Bindable
<b>autoRender</b>	No	No	No
	A boolean specifying whether the grid should automatically render or should render using its body.		
<b>dataSource</b>	Yes	No	Yes
	An expression that references a data object which should be rendered by the grid. Only data sets of type <code>RowSet</code> are supported.		
<b>name</b>	Yes	No	Yes
	A unique identifier for this grid.		

### Related Topics

<netui-data:grid> Tag Sample  
Presenting Complex Data Sets in JSPs  
Columns

---

### Hierarchy

```
Object
  TagSupport
    BodyTagSupport
      AbstractBaseTag
        DataSourceTag
          Grid
```

### All Implemented Interfaces

```
BodyTag, DataAccessProvider, ErrorReporter, IterationTag, Serializable,
Tag
```

---

## Field Summary

```
public
static END
final int
    A Grid rendering state that runs after the Grid.FOOT_PAGER state has completed.

public
static FOOT_PAGER
final int
    A Grid rendering state that runs after the grid to give the optional Pager the
    opportunity to render after the grid renders.
```

```

    public FOOTER
    static
    final int
        A Grid rendering state that runs when the Grid's columns are rendering their footer
        cells.

    public
    static HEAD_PAGER
    final int
        A Grid rendering state that runs to give the optional Pager the opportunity to render
        before the grid renders.

    public
    static HEADER
    final int
        A Grid rendering state that runs when the Grid's columns are rendering their header cell.

    public
    static INIT
    final int
        A Grid rendering state that runs on the Grid's first pass through its body.

    public
    static ROWS
    final int
        A Grid rendering state that runs when the Grid's columns are rendering their data cells.

    public
    static START
    final int
        A Grid rendering state which signals that an error has occurred during rendering.

```

### Fields from `com.bea.wlw.netui.tags.AbstractBaseTag`

```

ATTR_GENERAL, ATTR_GENERAL_EXPRESSION, ATTR_JAVASCRIPT, ATTR_STYLE,
CLASS, ID, JAVASCRIPT_STATUS, NAME, NETUI_UNIQUE_CNT, ONCLICK,
ONDBLCLICK, ONKEYDOWN, ONKEYPRESS, ONKEYUP, ONMOUSEDOWN, ONMOUSEMOVE,
ONMOUSEOUT, ONMOUSEOVER, ONMOUSEUP, STYLE, TABINDEX

```

### Fields from `javax.servlet.jsp.tagext.BodyTagSupport`

```
bodyContent
```

### Fields from `javax.servlet.jsp.tagext.TagSupport`

```
id, pageContext
```

### Fields from interface `javax.servlet.jsp.tagext.BodyTag`

```
EVAL_BODY_BUFFERED, EVAL_BODY_TAG
```

### Fields from interface `com.bea.wlw.netui.tags.ErrorReporter`

```
CONTAINER_ERRORS
```

### Fields from interface `javax.servlet.jsp.tagext.IterationTag`

```
EVAL_BODY_AGAIN
```

### Fields from interface `javax.servlet.jsp.tagext.Tag`

```
EVAL_BODY_INCLUDE, EVAL_PAGE, SKIP_BODY, SKIP_PAGE
```

## Constructor Summary

*Grid()*

## Method Summary

```
public final void
    addContent(String content)
        Add content to the content that is being buffered by this tag.

public void
    addError(AbstractPageError ape)

public int
    doAfterBody()
        Conginue rendering the grid until it has passed through all of its rendering
        states.

public int
    doEndTag()
        Render the content for the grid.

public int
    doStartTag()
        Start the rendering of the grid.

public boolean
    getAutoRender()
        Get a boolean that tells whether a grid will render automatically.

public final
    String getContent()
        Get the content which should be rendered by this tag.

public int
    getCurrentIndex()
        Get the index of the current iteration through the body of this tag.

public Object
    getCurrentItem()
        Get the item that is currently being rendered by this repeating tag.

public Object
    getCurrentMetadata()
        Get the metadata for the current item.

public final
    DataContext getDataContext()

public String
    getDataSource()
        Get the expression that references the data item to which the
        DataAccessProvider is bound.
```



```

protected String getErrorsReport()
    public final
    GridContext getGridContext()
    public String
        getImageRoot()
    public String
        getName()
        Get the name of a grid.
    public
DataAccessProvider getProviderParent()
        Get the parent DataAccessProvider for this tag.
    public int
        getRenderState()
        Get the current render state for the Grid.
    public String
        getTagName()
        Get the name of this tag.
protected boolean
    hasErrors()
    public boolean
        isReporting()
        This boolean indicates if an ErrorReporter is reporting errors or not.
protected void
    localRelease()
        Reset all of the fields of this tag.
public ArrayList
    returnErrors()
        Return an ArrayList of the errors
    public void
        setAutoRender(boolean autoRender)
        Set a boolean that tells a Grid to render it self automatically.
    public void
        setDataSource(String dataSource)
        Set the dataSource attribute.
    public void
        setGridStyle(GridStyleContext gridStyle)
        Set the GridStyleContext that encapsulates the style information for
        a Grid tag.
    public void
        setImageRoot(String imageRoot)
    public void
        setName(String name)
        Set the name of a grid.

```

#### Methods from `com.bea.wlw.netui.tags.DataSourceTag`

**Methods from `com.bea.wlw.netui.tags.AbstractBaseTag`**

`addTagIdMapping`, `evaluateAttributeToString`, `filter`, `filter`,  
`getJavaScriptUtils`, `getNearestForm`, `getScriptReporter`, `getUserLocale`,  
`localRelease`, `registerTagError`, `release`, `renderAttribute`, `rewriteName`,  
`write`

**Methods from `javax.servlet.jsp.tagext.BodyTagSupport`**

`doInitBody`, `getBodyContent`, `getPreviousOut`, `release`, `setBodyContent`

**Methods from `javax.servlet.jsp.tagext.TagSupport`**

`doAfterBody`, `doEndTag`, `doStartTag`, `findAncestorWithClass`, `getId`,  
`getParent`, `getValue`, `getValues`, `release`, `removeValue`, `setId`,  
`setPageContext`, `setParent`, `setValue`

**Methods from class `java.lang.Object`**

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`,  
`toString`, `wait`, `wait`, `wait`

**Methods from interface `javax.servlet.jsp.tagext.BodyTag`**

`doInitBody`, `setBodyContent`

**Methods from interface `com.bea.wlw.netui.tags.DataAccessProvider`**

`getCurrentIndex`, `getCurrentItem`, `getCurrentMetadata`, `getDataSource`,  
`getProviderParent`, `setDataSource`

**Methods from interface `com.bea.wlw.netui.tags.ErrorReporter`**

`addError`, `isReporting`, `returnErrors`

**Methods from interface `javax.servlet.jsp.tagext.IterationTag`**

`doAfterBody`

**Methods from interface `javax.servlet.jsp.tagext.Tag`**

`doEndTag`, `doStartTag`, `getParent`, `release`, `setPageContext`, `setParent`

## Field Detail

### END

```
public static final int END
```

A Grid rendering state that runs after the `Grid.FOOT_PAGER` state has completed. This state signals that the Grid has completed rendering.

## FOOT\_PAGER

```
public static final int FOOT_PAGER
```

A Grid rendering state that runs after the grid to give the optional `Pager` the opportunity to render after the grid renders. This state executes exactly once.

---

## FOOTER

```
public static final int FOOTER
```

A Grid rendering state that runs when the Grid's columns are rendering their footer cells. In this state, there is no data available to the Grid on the `DataAccessProvider` interface and expressions that reference the `container` context are invalid. This state executes exactly once.

---

## HEAD\_PAGER

```
public static final int HEAD_PAGER
```

A Grid rendering state that runs to give the optional `Pager` the opportunity to render before the grid renders. This state executes exactly once.

---

## HEADER

```
public static final int HEADER
```

A Grid rendering state that runs when the Grid's columns are rendering their header cell. In this state, there is no data available to the Grid on the `DataAccessProvider` interface and expressions that reference the `container` context are invalid. This state executes exactly once.

---

## INIT

```
public static final int INIT
```

A Grid rendering state that runs on the Grid's first pass through its body. Initialization by the child tags may occur on at this time. This state executes exactly once.

---

## ROWS

```
public static final int ROWS
```

A Grid rendering state that runs when the Grid's columns are rendering their data cells. The data cells contain data that is extracted from the "current" data item. This state executes once for each data item.

## START

```
public static final int START
```

A Grid rendering state which signals that an error has occurred during rendering.

## Constructor Detail

### Grid

```
public Grid()
```

## Method Detail

### addContent(String) Method

```
public final void addContent(String content)
```

Add content to the content that is being buffered by this tag. All content written by the body of this tag is added to this buffer. The buffer is rendered at the end of the tag's lifecycle if no fatal errors have occurred during this tag's lifecycle.

#### Parameters

*content*

content that this tag should render.

---

### addError(AbstractPageError) Method

```
public void addError(AbstractPageError ape)
```

---

### doAfterBody() Method

```
public int doAfterBody()
```

Continue rendering the grid until it has passed through all of its rendering states. This JSP lifecycle method transitions to the next grid rendering state and advances through the dataset until each of the following rendering states has completed:

1. INIT
2. HEAD\_PAGER

- 3. HEADER
- 4. ROWS
- 5. FOOTER
- 6. FOOT\_PAGER
- 7. END

While the execution of these lifecycle states is ongoing, the body content is buffered and added to the grid's content to render using the `Grid.addContent(String)` method. Once the END state is reached, the body is skipped. Each of these stages makes a pass through the tag's body once except for the ROWS state, which is executed once for each item in the dataset.

#### **Overrides**

`BodyTagSupport.doAfterBody()`

#### **Returns**

*SKIP\_BODY if in the END rendering state; EVAL\_BODY\_BUFFERED otherwise.*

---

### **doEndTag() Method**

```
public int doEndTag()
    throws JspException
```

Render the content for the grid.

#### **Overrides**

`BodyTagSupport.doEndTag()`

#### **Returns**

*EVAL\_PAGE*

#### **Exceptions**

*JspException*  
if an error has occurred that can not be reported in the page

---

### **doStartTag() Method**

```
public int doStartTag()
    throws JspException
```

Start the rendering of the grid. The `dataSource` attribute is evaluated as an expression and the result is converted into a `DataContext` object. The body is rendered the first time in the INIT state.

#### **Overrides**

`BodyTagSupport.doStartTag()`

## Returns

*EVAL\_BODY\_BUFFERED or SKIP\_BODY if errors have occurred*

## Exceptions

*JspException*

thrown when an error occurs during grid rendering that can not be reported in the page.

---

## getAutoRender() Method

```
public boolean getAutoRender()
```

Get a boolean that tells whether a grid will render automatically.

## Returns

*true if the grid should automatically render; false otherwise.*

---

## getContent() Method

```
public final String getContent()
```

Get the content which should be rendered by this tag.

## Returns

*the content String if there is content; null otherwise*

---

## getCurrentIndex() Method

```
public int getCurrentIndex()
```

Get the index of the current iteration through the body of this tag. This data can be accessed using the expression `container.index` on an attribute of a databindable NetUI tag that is contained within the repeating body of this tag. This expression is only valid when the dataset is being rendered.

## Returns

*the integer index of the current data item in the data set*

## Related Topics

DataAccessProvider

---

## getCurrentItem() Method

```
public Object getCurrentItem()
```

Get the item that is currently being rendered by this repeating tag. This can be accessed using the expression `expression.item` on an attribute of a databindable NetUI tag that is contained within the repeating body of this tag. The expression is only valid when the dataset is being rendered.

### Returns

*the current item in the data set*

### Related Topics

DataAccessProvider

---

## getCurrentMetadata() Method

```
public Object getCurrentMetadata()
```

Get the metadata for the current item. This method is not supported by this tag.

### Returns

*the current metadata or null if no metadata can be found or metadata is not supported by a DataAccessProvider implementation*

### Related Topics

DataAccessProvider

---

## getDataContext() Method

```
public final DataContext getDataContext()
```

### Returns

*the DataContext that is associated with a Grid tag*

---

## getDataSource() Method

```
public String getDataSource()
```

*Description copied from DataAccessProvider.getDataSource()*

Get the expression that references the data item to which the DataAccessProvider is bound.

### Overrides

```
DataSourceTag.getDataSource()
```

### Returns

*the expression referencing the data source or null if no dataSource is set*

---

## getErrorsReport() Method

```
protected String getErrorsReport()
```

### Overrides

```
AbstractBaseTag.getErrorsReport()
```

---

## getGridContext() Method

```
public final GridContext getGridContext()
```

### Returns

*the GridContext that is associated with a Grid tag*

---

## getImageRoot() Method

```
public String getImageRoot()
```

---

## getName() Method

```
public String getName()
```

Get the name of a grid.

### Returns

*the String name of the grid or an expression that references a String name of the grid*

---

## getProviderParent() Method

```
public DataAccessProvider getProviderParent()
```

Get the parent DataAccessProvider for this tag. If this tag is contained within a DataAccessProvider, the containing DataAccessProvider is available through the expression `container.container`. Any valid properties of the parent DataAccessProvider can be accessed through this expression. This method will return null if there is no parent DataAccessProvider



**Returns**

*a containing `DataProvider` if one exists, null otherwise.*

**Related Topics**

`DataProvider`

---

**getRenderState() Method**

```
public int getRenderState()
```

Get the current render state for the Grid. This tag is used by child tags to access the current location in the Grid's rendering lifecycle.

**Returns**

*an integer that represents the current state of the grid; this is one of `Grid.START`, `Grid.INIT`, `Grid.HEAD_PAGER`, `Grid.HEADER`, `Grid.ROWS`, `Grid.FOOTER`, `Grid.FOOT_PAGER`, or `Grid.END`*

---

**getTagName() Method**

```
public String getTagName()
```

Get the name of this tag. This is used to identify the type of this tag for reporting tag errors.

**Overrides**

```
AbstractBaseTag.getTagName()
```

**Returns**

*a constant String representing the name of this tag.*

---

**hasErrors() Method**

```
protected boolean hasErrors()
```

**Overrides**

```
AbstractBaseTag.hasErrors()
```

---

**isReporting() Method**

```
public boolean isReporting()
```

*Description copied from `ErrorReporter.isReporting()`*

This boolean indicates if an `ErrorReporter` is reporting errors or not. The caller should check this before calling `addError` because the `ErrorReporter` may be off for some reason.

### Returns

*a boolean indicating if the tag is reporting errors or not.*

---

### `localRelease()` Method

```
protected void localRelease()
```

Reset all of the fields of this tag.

#### *Overrides*

```
DataSourceTag.localRelease()
```

---

### `returnErrors()` Method

```
public ArrayList returnErrors()
```

*Description copied from `ErrorReporter.returnErrors()`*  
Return an `ArrayList` of the errors

### Returns

*an `ArrayList` of all errors.*

---

### `setAutoRender(boolean)` Method

```
public void setAutoRender(boolean autoRender)
```

Set a boolean that tells a Grid to render it self automatically. Automatic rendering of a dataset bypasses any user paramteerization in the body of the Grid and disallows several grid features including sorts, filters, and paging.

### Parameters

*autoRender*

a boolean signifying whether a grid should automatically render. If `true` auto render; otherwise render using the tags in the grid's body.

---

### `setDataSource(String)` Method

```
public void setDataSource(String dataSource)
```

Set the `dataSource` attribute. The `dataSource` attribute is an expression that references an object whose data

will be rendered by the Grid.

### ***Overrides***

`DataSourceTag.setDataSource(String)`

### **Parameters**

*dataSource*

an expression that references a data object

---

## **setGridStyle(GridStyleContext) Method**

```
public void setGridStyle(GridStyleContext gridStyle)
```

Set the GridStyleContext that encapsulates the style information for a Grid tag. A GridStyleContext is created by the GridStyle tag in order to parameterize the Grid with style information. The context is set here because it will be reused in many states during a Grid tag's lifecycle, and the specific GridStyle tag may not be valid when the style information is needed.

### **Returns**

*a GridStyleContext object that encapsulates the style information for a Grid tag*

---

## **setImageRoot(String) Method**

```
public void setImageRoot(String imageRoot)
```

---

## **setName(String) Method**

```
public void setName(String name)
```

Set the name of a grid. The name attribute of a grid is used to identify parameters in the URL that further parameterize how a grid is rendered. Specifically, the URL contains information about any sorts and filters that may be applied to a dataset that is associated with a grid. Grid names should be unique within the scope of the pages where the grid may exist; if a <netui-data:grid> exists in a Page Flow, its grid name should be unique across the page flow so that parameters in the URL which are scoped to a particular grid on a particular page do not collide.

### **Parameters**

*name*

the String name of the grid or an expression that references a String name of the grid

# GridComponent Class

*public abstract class GridComponent*

extends StructuredBaseTag

An abstract base class for all tags that can be contained directly inside of a Grid tag. This class provides services to subclasses such as methods to access:

- the grid's current render state
- the Grid tag itself
- the Grid tag's GridContext

Additionally, the requirement that any subclass be parented by a Grid is enforced here.

---

## ***Hierarchy***

```
Object
  TagSupport
    BodyTagSupport
      AbstractBaseTag
        StructuredBaseTag
          GridComponent
```

## ***All Implemented Interfaces***

BodyTag, IterationTag, Serializable, Tag

## ***Direct Known Subclasses***

Columns, GridColumn, GridStyle, Pager

---

## Field Summary

### Fields from com.bea.wlw.netui.tags.databinding.base.StructuredBaseTag

EMPTY\_STRING

### Fields from com.bea.wlw.netui.tags.AbstractBaseTag

ATTR\_GENERAL, ATTR\_GENERAL\_EXPRESSION, ATTR\_JAVASCRIPT, ATTR\_STYLE, CLASS, ID, JAVASCRIPT\_STATUS, NAME, NETUI\_UNIQUE\_CNT, ONCLICK, ONDBLCLICK, ONKEYDOWN, ONKEYPRESS, ONKEYUP, ONMOUSEDOWN, ONMOUSEMOVE, ONMOUSEOUT, ONMOUSEOVER, ONMOUSEUP, STYLE, TABINDEX

**Fields from `javax.servlet.jsp.tagext.BodyTagSupport`**`bodyContent`**Fields from `javax.servlet.jsp.tagext.TagSupport`**`id, pageContext`**Fields from interface `javax.servlet.jsp.tagext.BodyTag`**`EVAL_BODY_BUFFERED, EVAL_BODY_TAG`**Fields from interface `javax.servlet.jsp.tagext.IterationTag`**`EVAL_BODY_AGAIN`**Fields from interface `javax.servlet.jsp.tagext.Tag`**`EVAL_BODY_INCLUDE, EVAL_PAGE, SKIP_BODY, SKIP_PAGE`

## Constructor Summary

*`GridComponent()`*

## Method Summary

```

protected
    int getCurrentRenderState()
        Get the current render state of the parent Grid tag.

protected
final Grid getGrid()
    Get the parent Grid tag.

protected
    final getGridContext()
GridContext    Get the Grid tag's GridContext.

protected
    Class getValidContainerType()
        Get the default container tag type for subclasses of this tag.

protected
    void localRelease()
        Reset all of the fields of this tag.

protected
    void prepare()
        Prepare data that is provided to subclasses of this tag.

```

**Methods from `com.bea.wlw.netui.tags.databinding.base.StructuredBaseTag`**

`addContent`, `doEndTag`, `doStartTag`, `getContent`, `renderEndTag`,  
`renderStartTag`, `verifyAttributes`, `verifyStructure`

**Methods from `com.bea.wlw.netui.tags.AbstractBaseTag`**

`addTagIdMapping`, `evaluateAttributeToString`, `filter`, `filter`,  
`getJavaScriptUtils`, `getNearestForm`, `getScriptReporter`, `getTagName`,  
`getUserLocale`, `localRelease`, `registerTagError`, `release`, `renderAttribute`,  
`rewriteName`, `write`

**Methods from `javax.servlet.jsp.tagext.BodyTagSupport`**

`doAfterBody`, `doEndTag`, `doInitBody`, `doStartTag`, `getBodyContent`,  
`getPreviousOut`, `release`, `setBodyContent`

**Methods from `javax.servlet.jsp.tagext.TagSupport`**

`doAfterBody`, `doEndTag`, `doStartTag`, `findAncestorWithClass`, `getId`,  
`getParent`, `getValue`, `getValues`, `release`, `removeValue`, `setId`,  
`setPageContext`, `setParent`, `setValue`

**Methods from class `java.lang.Object`**

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`,  
`toString`, `wait`, `wait`, `wait`

**Methods from interface `javax.servlet.jsp.tagext.BodyTag`**

`doInitBody`, `setBodyContent`

**Methods from interface `javax.servlet.jsp.tagext.IterationTag`**

`doAfterBody`

**Methods from interface `javax.servlet.jsp.tagext.Tag`**

`doEndTag`, `doStartTag`, `getParent`, `release`, `setPageContext`, `setParent`

## Constructor Detail

### GridComponent

```
public GridComponent()
```

## Method Detail

### getCurrentRenderState() Method

```
protected int getCurrentRenderState()
```

Get the current render state of the parent Grid tag.

#### *Overrides*

StructuredBaseTag.getCurrentRenderState()

#### Returns

*the Grid's current render state*

#### Related Topics

Grid.getRenderState()

---

### getGrid() Method

```
protected final Grid getGrid()
```

Get the parent Grid tag.

#### Returns

*the Grid tag*

---

### getGridContext() Method

```
protected final GridContext getGridContext()
```

Get the Grid tag's GridContext.

#### Returns

*the GridContext*

---

### getValidContainerType() Method

```
protected Class getValidContainerType()
```

Get the default container tag type for subclasses of this tag. Subclasses may further specify the valid container type.

#### *Overrides*

com.bea.wlw.netui.tags.databinding.gridGridComponent Class

StructuredBaseTag.getValidContainerType()

## Returns

*a class that is a valid container type for this tag*

---

## localRelease() Method

```
protected void localRelease()
```

Reset all of the fields of this tag.

### Overrides

StructuredBaseTag.localRelease()

---

## prepare() Method

```
protected void prepare()
```

Prepare data that is provided to subclasses of this tag.

### Overrides

StructuredBaseTag.prepare()



# GridContext Class

*public final class GridContext*

extends Object

A JavaBean that represents the runtime data of a specific Grid tag instance. This bean provides access to properties about a Grid tag without having to have a reference to the Grid tag itself. Upon initialization, the Grid will create a GridContext object that the GridComponent class will provide to its subclasses. Information provided here includes the JavaBean that encapsulates the style data about a Grid and a DataContext object which exposes the data and metadata that a Grid will use to render. A GridContext is both created and destroyed in a Grid tag's lifecycle. It is guaranteed to be dereferenced by the Grid in its Grid.release() JSP lifecycle method call.

---

## Hierarchy

```

Object
  GridContext
  
```

---

## Constructor Summary

### *GridContext()*

Create a GridContext object.

## Method Summary

```

public boolean
    getAutoRender()
        Get the Grid's autoRender attribute,
        Grid.setAutoRender(boolean).

public
    DataContext getDataContext()
        Get the DataContext that contains the data the Grid will render

public String
    getDataSource()
        Get the dataSource attribute.

public
    GridStyleContext getGridStyle()
        Get a JavaBean that contains the style information for the grid.

public String
  
```

```

        getImageRoot()
public String
        getName()
            Get the name of the Grid.

public void
        setDataContext(DataContext dataContext)
            Set the DataContext object that the Grid uses to render data from a data set.

public void
        setDataSource(String dataSource)
            Set the dataSource attribute from the Grid tag.

public void
        setGridStyle(GridStyleContext gridStyleContext)
            Set a GridStyleContext JavaBean that contains style information for
            the grid.

public void
        setImageRoot(String imageRoot)

public void
        setName(String name)
            Set the name of the Grid using the Grid tag's Grid.setName(String)
            attribute.

```

#### Methods from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructor Detail

### GridContext

```
public GridContext()
```

Create a GridContext object. This initializes the DataContext to a basic, empty DataContext that provides basic data services to Grid tags. If an error occurs in creating a real DataContext that is specific to the dataset of the Grid, the empty DataContext may be used to render the grid in an empty state.

## Method Detail

### getAutoRender() Method

```
public boolean getAutoRender()
```

Get the Grid's autoRender attribute, Grid.setAutoRender(boolean).

**Returns**

*the autoRender attribute*

---

**getDataContext() Method**

```
public DataContext getDataContext()
```

Get the DataContext that contains the data the Grid will render

**Returns**

*the DataContext object*

---

**getDataSource() Method**

```
public String getDataSource()
```

Get the dataSource attribute.

**Returns**

*the expression that references the data a Grid will render.*

---

**getGridStyle() Method**

```
public GridStyleContext getGridStyle()
```

Get a JavaBean that contains the style information for the grid.

**Returns**

*the style information for the Grid*

---

**getImageRoot() Method**

```
public String getImageRoot()
```

---

**getName() Method**

```
public String getName()
```

Get the name of the Grid.

## Returns

*the name of the Grid*

---

## setDataContext(DataContext) Method

```
public void setDataContext(DataContext dataContext)
    throws DataContextException
```

Set the DataContext object that the Grid uses to render data from a data set. It is illegal to try to set a null DataContext on a GridContext.

## Parameters

*dataContext*  
the DataContext object to set.

## Exceptions

*DataContextException*  
if the dataContext parameter is null

## Related Topics

[DataContext](#)

---

## setDataSource(String) Method

```
public void setDataSource(String dataSource)
```

Set the dataSource attribute from the Grid tag. This attribute is an expression that references a dataset that should be rendered in the grid.

## Parameters

*dataSource*  
the expression that references an Object which the grid will render.

---

## setGridStyle(GridStyleContext) Method

```
public void setGridStyle(GridStyleContext gridStyleContext)
```

Set a GridStyleContext JavaBean that contains style information for the grid.

## Parameters

*gridStyleContext*  
the style information for the Grid

## setImageRoot(String) Method

```
public void setImageRoot(String imageRoot)
```

---

## setName(String) Method

```
public void setName(String name)
```

Set the name of the Grid using the Grid tag's `Grid.setName(String)` attribute.

### Parameters

*name*

the Grid tag's name attribute

# com.bea.wlw.netui.tags.databinding.grid.column Package

## Class Summary

AnchorColumn	A column that renders an HTML anchor into each data cell in the column.
BasicColumn	A BasicColumn simply renders data from the data set into the page.
ExpressionColumn	A column that can use expressions in addition to formatters to format the value of a data cell.
FormattableColumn	The abstract base class of a GridColumn that can accept NetUI FormatTag tags.
GridColumn	The abstract base class for all column tags that can be children of the Columns tag.
ImageColumn	A column that renders an image in HTML into each data cell in a column.
SortFilterColumn	The abstract base class of a grid column that is sortable and / or filterable.

# AnchorColumn Class

*public class **AnchorColumn***

extends ImageColumn  
implements URLParams

A column that renders an HTML anchor into each data cell in the column. The anchor that is created is constructed from attributes on this tag and any nested `Parameter` or `ParameterMap` tags; this `GridColumn` implements the `URLParams` interface. The anchor can be constructed in one of two ways — using text provided in the `title` attribute or using data from the current item in the data set. If using the `title` attribute, the column is not filterable or sortable as there is no data from the data set to filter or sort. If using data from the current item, the column can be filtered and sorted.

The `AnchorColumn` has the ability to render an image inside the anchor, resulting in the image being clickable. This can be accomplished by setting the `src` attribute and other attributes that are available on the `ImageColumn` tag. An image will be rendered if the `src` attribute is set and the `name` attribute is null. Otherwise, the data in the current data item that matches the `name` attribute will be rendered. If rendering an image using the `src` attribute, the column can be sorted and filtered on the name of the image.

In constructing an anchor from the attributes on the tag, exactly one of the `href` and `action` attributes must be specified. Providing both or neither will result in an error being reported.

## Required resources:

The JavaScript file `_grid.js` must be included in any JSP page with a grid that uses a filter-enabled `AnchorColumn`. The Page Flow Wizard adds the `_grid.js` file to a project's `/resources/grid` directory, when you build a page flow over a database control. The JSP should contain:

```
<script language="javascript" src="/resources/grid/_grid.js" type="text/javascript"></script>
```

Attribute Descriptions			
Attribute	Required	Runtime Expression Evaluation	Data Bindable
<b>action</b>	No	No	No
	The action that should be part of the HTML anchor that is rendered in this column.		
<b>addRowId</b>	No	Yes	No
	A boolean that denotes that the unique identifier for this row, if one is available, will be rendered onto the generated HTML anchor.		
<b>alt</b>	No	Yes	No

	The alternate text displayed in the browser if the image cannot be displayed.		
<b>border</b>	No	Yes	No
	The number of pixels to display in a border around this image. The default border is zero.		
<b>filterable</b>	No	Yes	No
	A boolean specifying whether the anchors can be filtered.		
<b>filterAction</b>	No	Yes	No
	The name of the action that will perform the filter operation.		
<b>href</b>	No	Yes	No
	The web addressable resource that will be part of the HTML anchor that is rendered in this column.		
<b>name</b>	No	Yes	Yes
	If specified, the value of the data is extracted from the Grid's current data item. Otherwise, the <code>title</code> attribute is used for the cell's content.		
<b>sortable</b>	No	Yes	No
	A boolean specifying whether the anchor column's contents can be sorted.		
<b>sortAction</b>	No	Yes	No
	The name of the action that will perform the sort operation.		
<b>title</b>	No	No	No
	The text that will be rendered in the header for this column.		

## Related Topics

<netui-data:grid> Tag Sample

---

## Hierarchy

```

Object
  TagSupport
    BodyTagSupport
      AbstractBaseTag
        StructuredBaseTag
          GridComponent
            GridColumn
              FormattableColumn
                SortFilterColumn
                  ImageColumn

```



**All Implemented Interfaces**

BodyTag, Formattable, IterationTag, Serializable, Tag, URLParams

---

## Field Summary

### Fields from `com.bea.wlw.netui.tags.databinding.base.StructuredBaseTag`

EMPTY\_STRING

### Fields from `com.bea.wlw.netui.tags.AbstractBaseTag`

ATTR\_GENERAL, ATTR\_GENERAL\_EXPRESSION, ATTR\_JAVASCRIPT, ATTR\_STYLE, CLASS, ID, JAVASCRIPT\_STATUS, NAME, NETUI\_UNIQUE\_CNT, ONCLICK, ONDBLCLICK, ONKEYDOWN, ONKEYPRESS, ONKEYUP, ONMOUSEDOWN, ONMOUSEMOVE, ONMOUSEOUT, ONMOUSEOVER, ONMOUSEUP, STYLE, TABINDEX

### Fields from `javax.servlet.jsp.tagext.BodyTagSupport`

bodyContent

### Fields from `javax.servlet.jsp.tagext.TagSupport`

id, pageContext

### Fields from interface `javax.servlet.jsp.tagext.BodyTag`

EVAL\_BODY\_BUFFERED, EVAL\_BODY\_TAG

### Fields from interface `javax.servlet.jsp.tagext.IterationTag`

EVAL\_BODY\_AGAIN

### Fields from interface `javax.servlet.jsp.tagext.Tag`

EVAL\_BODY\_INCLUDE, EVAL\_PAGE, SKIP\_BODY, SKIP\_PAGE

## Constructor Summary

*AnchorColumn()*

## Method Summary

```

public
    void addParameter(String name, Object value)
        This method allows the key / value pairs that will be written onto the URL to be added
        to the anchor that is rendered for each cell in the column that is rendered for an item in
        the dataset.

public
    String getAction()
        Get the action that is rendered in the anchor of this column.

public
    boolean getAddRowId()
        Get the boolean for whether this tag will encode a record's unique identifier into the
        URL.

public
    String getHref()
        Get the href that is rendered in the anchor of this column.

public
    String getTagName()
        Get the name of this tag.

protected
    void localRelease()
        Reset all of the fields of this tag.

protected
    void prepare()
        Perform initialization for this tag.

public
    String renderDataCell()
        Render the contents of this AnchorColumn for the current data item from the
        containing Grid tag.

public
    String renderFooterCell()
        Render the footer.

protected
    int renderStartTag(int gridState)
        This tag only evaluates its body when the grid is rendering its ROWS.

public
    void setAction(String action)
        Set the action that is rendered in the anchor of this column.

public
    void setAddRowId(boolean addRowId)
        Set a boolean that, if true, will encode the record's unique identifier, often the
        primary key, onto each anchor's URL link.

public
    void setHref(String href)
        Set the href that is rendered in the anchor of this column.

```

protected **verifyAttributes()**  
 void          Verify the attributes on this tag.

#### Methods from **com.bea.wlw.netui.tags.databinding.grid.column.ImageColumn**

getAlt, getBorder, getHeight, getSrc, getWidth, renderImage, setAlt, setBorder, setHeight, setSrc, setWidth,

#### Methods from

##### **com.bea.wlw.netui.tags.databinding.grid.column.SortFilterColumn**

canFilter, canSort, computeFilterAction, computeSortAction, getFilterAction, getName, getSortAction, getTitle, getTitleText, isFilterable, isSortable, localRelease, prepare, renderHeaderCell, setFilterable, setFilterAction, setName, setSortable, setSortAction, setTitle, verifyAttributes

#### Methods from

##### **com.bea.wlw.netui.tags.databinding.grid.column.FormatTableColumn**

addFormatter, format, localRelease, renderStartTag

#### Methods from **com.bea.wlw.netui.tags.databinding.grid.column.GridColumn**

getColumns, getStyleClassPrefix, getValidContainerType, localRelease, prepare, renderCloseCellTag, renderDataCell, renderDataCellTag, renderEndTag, renderFooterCell, renderFooterCellTag, renderHeaderCell, renderHeaderCellTag, renderOpenCellTag, renderStartTag, setStyleClassPrefix

#### Methods from **com.bea.wlw.netui.tags.databinding.grid.GridComponent**

getCurrentRenderState, getGrid, getGridContext, getValidContainerType, localRelease, prepare

#### Methods from **com.bea.wlw.netui.tags.databinding.base.StructuredBaseTag**

addContent, doEndTag, doStartTag, getContent, getCurrentRenderState, getValidContainerType, localRelease, prepare, renderEndTag, renderStartTag, verifyAttributes, verifyStructure

#### Methods from **com.bea.wlw.netui.tags.AbstractBaseTag**

addTagIdMapping, evaluateAttributeToString, filter, filter, getJavaScriptUtils, getNearestForm, getScriptReporter, getTagName, getUserLocale, localRelease, registerTagError, release, renderAttribute, rewriteName, write

#### Methods from **javax.servlet.jsp.tagext.BodyTagSupport**

doAfterBody, doEndTag, doInitBody, doStartTag, getBodyContent, getPreviousOut, release, setBodyContent

### Methods from `javax.servlet.jsp.tagext.TagSupport`

`doAfterBody`, `doEndTag`, `doStartTag`, `findAncestorWithClass`, `getId`,  
`getParent`, `getValue`, `getValues`, `release`, `removeValue`, `setId`,  
`setPageContext`, `setParent`, `setValue`

### Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`,  
`toString`, `wait`, `wait`, `wait`

### Methods from interface `javax.servlet.jsp.tagext.BodyTag`

`doInitBody`, `setBodyContent`

### Methods from interface `com.bea.wlw.netui.tags.html.Formattable`

`addFormatter`

### Methods from interface `javax.servlet.jsp.tagext.IterationTag`

`doAfterBody`

### Methods from interface `javax.servlet.jsp.tagext.Tag`

`doEndTag`, `doStartTag`, `getParent`, `release`, `setPageContext`, `setParent`

### Methods from interface `com.bea.wlw.netui.tags.html.URLParams`

`addParameter`

## Constructor Detail

### AnchorColumn

```
public AnchorColumn()
```

## Method Detail

### `addParameter(String, Object)` Method

```
public void addParameter(String name,  
                        Object value)
```

This method allows the key / value pairs that will be written onto the URL to be added to the anchor that is rendered for each cell in the column that is rendered for an item in the dataset.

**Related Topics**

URLParams  
Parameter  
ParameterMap

---

**getAction() Method**

```
public String getAction()
```

Get the action that is rendered in the anchor of this column.

**Returns**

*the name of the action or null*

---

**getAddRowId() Method**

```
public boolean getAddRowId()
```

Get the boolean for whether this tag will encode a record's unique identifier into the URL.

**Returns**

*true if the unique identifier will be encoded; false otherwise.*

---

**getHref() Method**

```
public String getHref()
```

Get the href that is rendered in the anchor of this column.

**Returns**

*the href that was specified or null*

---

**getTagName() Method**

```
public String getTagName()
```

Get the name of this tag. This is used to identify the type of this tag for reporting tag errors.

**Overrides**

```
ImageColumn.getTagName()
```

**Returns**

*a constant String representing the name of this tag.*

---

**localRelease() Method**

```
protected void localRelease()
```

Reset all of the fields of this tag.

**Overrides**

```
ImageColumn.localRelease()
```

---

**prepare() Method**

```
protected void prepare()
```

Perform initialization for this tag.

**Overrides**

```
ImageColumn.prepare()
```

---

**renderDataCell() Method**

```
public String renderDataCell()
    throws JspException
```

Render the contents of this AnchorColumn for the current data item from the containing Grid tag. An AnchorColumn uses either the href or the action attribute to render an HTML anchor into each data cell in the column. If a name attribute is specified, the value of the data is extracted from the Grid's current data item. Otherwise, the title attribute is used for the cell's content.

**Overrides**

```
ImageColumn.renderDataCell()
```

**Exceptions**

*JspException*

if an error occurred that can not be reported in the page

---

**renderFooterCell() Method**

```
public String renderFooterCell()
```

Render the footer. This footer contains no text.

**Overrides**

```
ImageColumn.renderFooterCell()
```

## Returns

*the empty string*

---

## renderStartTag(int) Method

```
protected int renderStartTag(int gridState)
```

This tag only evaluates its body when the grid is rendering its ROWS.

### Overrides

```
ImageColumn.renderStartTag(int)
```

## Parameters

*gridState*

the state of this column's Grid tag parent

---

## setAction(String) Method

```
public void setAction(String action)
```

Set the action that is rendered in the anchor of this column. The use of the action attribute on an AnchorColumn requires a PageFlowController as an action must be defined in order to render the href attribute on the resulting anchor.

## Parameters

*action*

the name of the action from a PageFlow

## Related Topics

PageFlowController

---

## setAddRowId(boolean) Method

```
public void setAddRowId(boolean addRowId)
```

Set a boolean that, if true, will encode the record's unique identifier, often the primary key, onto each anchor's URL link.

## Parameters

*addRowId*

a boolean that will encode the unique identifier if `true`

---

## setHref(String) Method

```
public void setHref(String href)
```

Set the href that is rendered in the anchor of this column. Only one of the href and action attributes may be set on this tag.

## Parameters

*href*

the href to render in the HTML anchor tag.

---

## verifyAttributes() Method

```
protected void verifyAttributes()  
    throws JspException
```

Verify the attributes on this tag. Exactly one of the attributes href and action must be specified.

## Overrides

```
ImageColumn.verifyAttributes()
```

## Exceptions

*JspException*

if an error occurs that can not be reported in the page



# BasicColumn Class

*public class* **BasicColumn**

extends `SortFilterColumn`

A `BasicColumn` simply renders data from the data set into the page. The data is referenced with the `name` attribute and the title is specified using the `title` attribute.

## Required resources:

The JavaScript files `_grid.js` must be included in any JSP page with a grid that uses a filter-enabled `BasicColumn`. The Page Flow Wizard adds the `_grid.js` file to a project's `/resources/grid` directory, when you build a page flow over a database control. The JSP should contain:

```
<script language="javascript" src="/resources/grid/_grid.js" type="text/javascript"></script>
```

Attribute Descriptions			
Attribute	Required	Runtime Expression Evaluation	Data Bindable
<b>filterable</b>	No	Yes	No
	A boolean specifying whether the anchors can be filtered.		
<b>filterAction</b>	No	Yes	No
	The name of the action that will perform the filter operation.		
<b>name</b>	No	Yes	Yes
	If specified, the value of the data is extracted from the Grid's current data item. Otherwise, the <code>title</code> attribute is used for the cell's content.		
<b>sortable</b>	No	Yes	No
	A boolean specifying whether the basic column's contents can be sorted.		
<b>sortAction</b>	No	Yes	No
	The name of the action that will perform the sort operation.		
<b>title</b>	No	No	No
	The text that will be rendered in the header for this column.		

## Related Topics

<netui-data:grid> Tag Sample

***Hierarchy***

```
Object
  TagSupport
    BodyTagSupport
      AbstractBaseTag
        StructuredBaseTag
          GridComponent
            GridColumn
              FormattableColumn
                SortFilterColumn
                  BasicColumn
```

***All Implemented Interfaces***

BodyTag, Formattable, IterationTag, Serializable, Tag

---

## Field Summary

**Fields from `com.bea.wlw.netui.tags.databinding.base.StructuredBaseTag`**

EMPTY\_STRING

**Fields from `com.bea.wlw.netui.tags.AbstractBaseTag`**

ATTR\_GENERAL, ATTR\_GENERAL\_EXPRESSION, ATTR\_JAVASCRIPT, ATTR\_STYLE, CLASS, ID, JAVASCRIPT\_STATUS, NAME, NETUI\_UNIQUE\_CNT, ONCLICK, ONDBLCLICK, ONKEYDOWN, ONKEYPRESS, ONKEYUP, ONMOUSEDOWN, ONMOUSEMOVE, ONMOUSEOUT, ONMOUSEOVER, ONMOUSEUP, STYLE, TABINDEX

**Fields from `javax.servlet.jsp.tagext.BodyTagSupport`**

bodyContent

**Fields from `javax.servlet.jsp.tagext.TagSupport`**

id, pageContext

**Fields from interface `javax.servlet.jsp.tagext.BodyTag`**

EVAL\_BODY\_BUFFERED, EVAL\_BODY\_TAG

**Fields from interface `javax.servlet.jsp.tagext.IterationTag`**

EVAL\_BODY\_AGAIN

**Fields from interface `javax.servlet.jsp.tagext.Tag`**

EVAL\_BODY\_INCLUDE, EVAL\_PAGE, SKIP\_BODY, SKIP\_PAGE

## Constructor Summary

*BasicColumn()*

## Method Summary

```

public
String getTagName()
    Get the name of this tag.

protected
void localRelease()
    Reset all of the fields of this tag.

public
String renderDataCell()
    For each cell in a column that is rendered for an item in the Grid's dataset, this method
    is called to render the data for the "current" item.

public
String renderFooterCell()
    Render the footer.

```

### Methods from

**com.bea.wlw.netui.tags.databinding.grid.column.SortFilterColumn**

canFilter, canSort, computeFilterAction, computeSortAction, getFilterAction, getName, getSortAction, getTitle, getTitleText, isFilterable, isSortable, prepare, renderHeaderCell, setFilterable, setFilterAction, setName, setSortable, setSortAction, setTitle, verifyAttributes

### Methods from

**com.bea.wlw.netui.tags.databinding.grid.column.FormattableColumn**

addFormatter, format, localRelease, renderStartTag

### Methods from **com.bea.wlw.netui.tags.databinding.grid.column.GridColumn**

getColumns, getStyleClassPrefix, getValidContainerType, localRelease, prepare, renderCloseCellTag, renderDataCellTag, renderEndTag, renderFooterCellTag, renderHeaderCell, renderHeaderCellTag, renderOpenCellTag, renderStartTag, setStyleClassPrefix

### Methods from **com.bea.wlw.netui.tags.databinding.grid.GridComponent**

`getCurrentRenderState, getGrid, getGridContext, getValidContainerType, localRelease, prepare`

**Methods from `com.bea.wlw.netui.tags.databinding.base.StructuredBaseTag`**

`addContent, doEndTag, doStartTag, getContent, getCurrentRenderState, getValidContainerType, localRelease, prepare, renderEndTag, renderStartTag, verifyAttributes, verifyStructure`

**Methods from `com.bea.wlw.netui.tags.AbstractBaseTag`**

`addTagIdMapping, evaluateAttributeToString, filter, filter, getJavaScriptUtils, getNearestForm, getScriptReporter, getUserLocale, localRelease, registerTagError, release, renderAttribute, rewriteName, write`

**Methods from `javax.servlet.jsp.tagext.BodyTagSupport`**

`doAfterBody, doEndTag, doInitBody, doStartTag, getBodyContent, getPreviousOut, release, setBodyContent`

**Methods from `javax.servlet.jsp.tagext.TagSupport`**

`doAfterBody, doEndTag, doStartTag, findAncestorWithClass, getId, getParent, getValue, getValues, release, removeValue, setId, setPageContext, setParent, setValue`

**Methods from class `java.lang.Object`**

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

**Methods from interface `javax.servlet.jsp.tagext.BodyTag`**

`doInitBody, setBodyContent`

**Methods from interface `com.bea.wlw.netui.tags.html.Formattable`**

`addFormatter`

**Methods from interface `javax.servlet.jsp.tagext.IterationTag`**

`doAfterBody`

**Methods from interface `javax.servlet.jsp.tagext.Tag`**

`doEndTag, doStartTag, getParent, release, setPageContext, setParent`

## Constructor Detail

### BasicColumn

```
public BasicColumn()
```

## Method Detail

### getTagName() Method

```
public String getTagName()
```

Get the name of this tag. This is used to identify the type of this tag for reporting tag errors.

#### *Overrides*

```
AbstractBaseTag.getTagName()
```

#### Returns

*a constant String representing the name of this tag.*

---

### localRelease() Method

```
protected void localRelease()
```

Reset all of the fields of this tag.

#### *Overrides*

```
SortFilterColumn.localRelease()
```

---

### renderDataCell() Method

```
public String renderDataCell()
```

For each cell in a column that is rendered for an item in the Grid's dataset, this method is called to render the data for the "current" item. Using the name attribute on this tag, the value associated with that name for the current data item is rendered in a table cell here. The String contents of this cell are added to the Columns content buffer.

#### *Overrides*

```
GridColumn.renderDataCell()
```

## Returns

*the String contents of this table cell.*

---

## renderFooterCell() Method

```
public String renderFooterCell()
```

Render the footer. This footer contains no text.

## Overrides

```
GridColumn.renderFooterCell()
```

## Returns

*the empty string*

# ExpressionColumn Class

*public class* **ExpressionColumn**

extends `FormattableColumn`

A column that can use expressions in addition to formatters to format the value of a data cell. The expression column enables combining data from different fields of the current data item or even from different expression contexts into a value for the cell. An expression column is not sortable or filterable.

Attribute Descriptions			
Attribute	Required	Runtime Expression Evaluation	Data Bindable
title	No	No	No
	The text that will be rendered in the header for this column.		
value	No	No	Yes
	The expression value that can contain expressions that reference data in the current data item. The result of the expression evaluation will be rendered in the data cell.		

## Related Topics

Grid

---

## Hierarchy

```

Object
  TagSupport
    BodyTagSupport
      AbstractBaseTag
        StructuredBaseTag
          GridComponent
            GridColumn
              FormattableColumn
                ExpressionColumn
  
```

## All Implemented Interfaces

BodyTag, Formattable, IterationTag, Serializable, Tag

---

## Field Summary

### Fields from `com.bea.wlw.netui.tags.databinding.base.StructuredBaseTag`

`EMPTY_STRING`

### Fields from `com.bea.wlw.netui.tags.AbstractBaseTag`

`ATTR_GENERAL`, `ATTR_GENERAL_EXPRESSION`, `ATTR_JAVASCRIPT`, `ATTR_STYLE`, `CLASS`, `ID`, `JAVASCRIPT_STATUS`, `NAME`, `NETUI_UNIQUE_CNT`, `ONCLICK`, `ONDBLCLICK`, `ONKEYDOWN`, `ONKEYPRESS`, `ONKEYUP`, `ONMOUSEDOWN`, `ONMOUSEMOVE`, `ONMOUSEOUT`, `ONMOUSEOVER`, `ONMOUSEUP`, `STYLE`, `TABINDEX`

### Fields from `javax.servlet.jsp.tagext.BodyTagSupport`

`bodyContent`

### Fields from `javax.servlet.jsp.tagext.TagSupport`

`id`, `pageContext`

### Fields from interface `javax.servlet.jsp.tagext.BodyTag`

`EVAL_BODY_BUFFERED`, `EVAL_BODY_TAG`

### Fields from interface `javax.servlet.jsp.tagext.IterationTag`

`EVAL_BODY_AGAIN`

### Fields from interface `javax.servlet.jsp.tagext.Tag`

`EVAL_BODY_INCLUDE`, `EVAL_PAGE`, `SKIP_BODY`, `SKIP_PAGE`

## Constructor Summary

*`ExpressionColumn()`*

## Method Summary

```
public
String getTagName()
    Get the name of this tag.

public
String getTitle()
```



Get the title rendered in the header cell of this column.

```
public
String getValue()
```

Get the value of this tag.

```
protected
```

```
void localRelease()
```

Reset all of the fields of this tag.

```
public
String renderDataCell()
```

Render a cell for the "current" data item.

```
public
String renderFooterCell()
```

Render the footer.

```
public
String renderHeaderCell()
```

Render the header cell for this column.

```
public
void setTitle(String title)
```

Set the title rendered in the header cell of this column.

```
public
void setValue(String value)
```

Set the value of this tag.

#### Methods from

**com.bea.wlw.netui.tags.databinding.grid.column.FormattableColumn**

addFormatter, format, renderStartTag

#### Methods from **com.bea.wlw.netui.tags.databinding.grid.column.GridColumn**

getColumns, getStyleClassPrefix, getValidContainerType, localRelease, prepare, renderCloseCellTag, renderDataCellTag, renderEndTag, renderFooterCellTag, renderHeaderCellTag, renderOpenCellTag, renderStartTag, setStyleClassPrefix

#### Methods from **com.bea.wlw.netui.tags.databinding.grid.GridComponent**

getCurrentRenderState, getGrid, getGridContext, getValidContainerType, localRelease, prepare

#### Methods from **com.bea.wlw.netui.tags.databinding.base.StructuredBaseTag**

addContent, doEndTag, doStartTag, getContent, getCurrentRenderState, getValidContainerType, localRelease, prepare, renderEndTag, renderStartTag, verifyAttributes, verifyStructure

#### Methods from **com.bea.wlw.netui.tags.AbstractBaseTag**

addTagIdMapping, evaluateAttributeToString, filter, filter,  
getJavaScriptUtils, getNearestForm, getScriptReporter, getUserLocale,  
localRelease, registerTagError, release, renderAttribute, rewriteName,  
write

### Methods from `javax.servlet.jsp.tagext.BodyTagSupport`

doAfterBody, doEndTag, doInitBody, doStartTag, getBodyContent,  
getPreviousOut, release, setBodyContent

### Methods from `javax.servlet.jsp.tagext.TagSupport`

doAfterBody, doEndTag, doStartTag, findAncestorWithClass, getId,  
getParent, getValue, getValues, release, removeValue, setId,  
setPageContext, setParent, setValue

### Methods from class `java.lang.Object`

clone, equals, finalize, getClass, hashCode, notify, notifyAll,  
toString, wait, wait, wait

### Methods from interface `javax.servlet.jsp.tagext.BodyTag`

doInitBody, setBodyContent

### Methods from interface `com.bea.wlw.netui.tags.html.Formattable`

addFormatter

### Methods from interface `javax.servlet.jsp.tagext.IterationTag`

doAfterBody

### Methods from interface `javax.servlet.jsp.tagext.Tag`

doEndTag, doStartTag, getParent, release, setPageContext, setParent

## Constructor Detail

### ExpressionColumn

```
public ExpressionColumn()
```

## Method Detail

## getTagName() Method

```
public String getTagName()
```

Get the name of this tag. This is used to identify the type of this tag for reporting tag errors.

### *Overrides*

```
AbstractBaseTag.getTagName()
```

### Returns

*a constant String representing the name of this tag.*

---

## getTitle() Method

```
public String getTitle()
```

Get the title rendered in the header cell of this column.

### Returns

*the title of the colum*

---

## getValue() Method

```
public String getValue()
```

Get the value of this tag.

### Returns

*the value of this tag*

---

## localRelease() Method

```
protected void localRelease()
```

Reset all of the fields of this tag.

### *Overrides*

```
FormattableColumn.localRelease()
```

---

## renderDataCell() Method

```
public String renderDataCell()  
    throws JspException
```

Render a cell for the "current" data item. The value attribute is first evaluated as an expression, then formatted using any contained Formatter tags, and is finally returned to be buffered. If errors occur during the process of expression evaluation on the value attribute, these errors are reported in the page.

### ***Overrides***

```
GridColumn.renderDataCell()
```

### **Returns**

*the String content that should be buffered for this tag*

### **Exceptions**

*JspException*

if an error occurs that can not be reported in the page

---

## **renderFooterCell() Method**

```
public String renderFooterCell()
```

Render the footer. This footer contains no text.

### ***Overrides***

```
GridColumn.renderFooterCell()
```

### **Returns**

*the empty string*

---

## **renderHeaderCell() Method**

```
public String renderHeaderCell()
```

Render the header cell for this column. The title attribute is rendered in the header of an ExpressionColumn.

### ***Overrides***

```
GridColumn.renderHeaderCell()
```

### **Returns**

*the String content that should be buffered for this tag*

---

## **setTitle(String) Method**

```
public void setTitle(String title)
```

Set the title rendered in the header cell of this column.

## Parameters

*title*

the title of the column

---

## setValue(String) Method

```
public void setValue(String value)
```

Set the value of this tag. The value may contain an expression(s) that will be evaluated before the data for each item in the Grid's data set is rendered. The databinding expression `container` may be used to access the "current" data item that is being rendered. For example, the expression `container.item` accesses the current data item.

## Parameters

*value*

the value of this tag. This attribute may contain expression(s).

# FormattableColumn Class

*public abstract class FormattableColumn*

extends GridColumn  
implements Formattable

The abstract base class of a GridColumn that can accept NetUI FormatTag tags. Formatter tags are used to provide a consistent appearance of data that is rendered in a cell in a GridColumn. The value of a data cell is formatted before it is rendered.

## Related Topics

Formattable

---

## Hierarchy

```

Object
  TagSupport
    BodyTagSupport
      AbstractBaseTag
        StructuredBaseTag
          GridComponent
            GridColumn
              FormattableColumn
  
```

## All Implemented Interfaces

BodyTag, Formattable, IterationTag, Serializable, Tag

## Direct Known Subclasses

ExpressionColumn, SortFilterColumn

---

## Field Summary

### Fields from com.bea.wlw.netui.tags.databinding.base.StructuredBaseTag

EMPTY\_STRING

### Fields from com.bea.wlw.netui.tags.AbstractBaseTag

ATTR\_GENERAL, ATTR\_GENERAL\_EXPRESSION, ATTR\_JAVASCRIPT, ATTR\_STYLE, CLASS, ID, JAVASCRIPT\_STATUS, NAME, NETUI\_UNIQUE\_CNT, ONCLICK, ONDBLCLICK, ONKEYDOWN, ONKEYPRESS, ONKEYUP, ONMOUSEDOWN, ONMOUSEMOVE,

ONMOUSEOUT, ONMOUSEOVER, ONMOUSEUP, STYLE, TABINDEX

#### Fields from `javax.servlet.jsp.tagext.BodyTagSupport`

`bodyContent`

#### Fields from `javax.servlet.jsp.tagext.TagSupport`

`id`, `pageContext`

#### Fields from interface `javax.servlet.jsp.tagext.BodyTag`

`EVAL_BODY_BUFFERED`, `EVAL_BODY_TAG`

#### Fields from interface `javax.servlet.jsp.tagext.IterationTag`

`EVAL_BODY_AGAIN`

#### Fields from interface `javax.servlet.jsp.tagext.Tag`

`EVAL_BODY_INCLUDE`, `EVAL_PAGE`, `SKIP_BODY`, `SKIP_PAGE`

## Constructor Summary

*`FormattableColumn()`*

## Method Summary

```

public
    void addFormatter(FormatTag.Formatter formatter)
        Add a FormatTag to this tag.

protected
    String format(Object value)
        Format the text of the column with all of the registered FormatTag tags.

protected
    void localRelease()
        Reset all of the fields of this tag.

protected
    int renderStartTag(int gridState)
        In order for a formatted column to pick-up the formatters, it should evaluate its
        body.
    
```

Methods from `com.bea.wlw.netui.tags.databinding.grid.column.GridColumn`

getColumns, getStyleClassPrefix, getValidContainerType, prepare, renderCloseCellTag, renderDataCell, renderDataCellTag, renderEndTag, renderFooterCell, renderFooterCellTag, renderHeaderCell, renderHeaderCellTag, renderOpenCellTag, setStyleClassPrefix

### Methods from **com.bea.wlw.netui.tags.databinding.grid.GridComponent**

getCurrentRenderState, getGrid, getGridContext, getValidContainerType, localRelease, prepare

### Methods from **com.bea.wlw.netui.tags.databinding.base.StructuredBaseTag**

addContent, doEndTag, doStartTag, getContent, getCurrentRenderState, getValidContainerType, localRelease, prepare, renderEndTag, renderStartTag, verifyAttributes, verifyStructure

### Methods from **com.bea.wlw.netui.tags.AbstractBaseTag**

addTagIdMapping, evaluateAttributeToString, filter, filter, getJavaScriptUtils, getNearestForm, getScriptReporter, getTagName, getUserLocale, localRelease, registerTagError, release, renderAttribute, rewriteName, write

### Methods from **javax.servlet.jsp.tagext.BodyTagSupport**

doAfterBody, doEndTag, doInitBody, doStartTag, getBodyContent, getPreviousOut, release, setBodyContent

### Methods from **javax.servlet.jsp.tagext.TagSupport**

doAfterBody, doEndTag, doStartTag, findAncestorWithClass, getId, getParent, getValue, getValues, release, removeValue, setId, setPageContext, setParent, setValue

### Methods from class **java.lang.Object**

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Methods from interface **javax.servlet.jsp.tagext.BodyTag**

doInitBody, setBodyContent

### Methods from interface **com.bea.wlw.netui.tags.html.Formatable**

addFormatter

### Methods from interface **javax.servlet.jsp.tagext.IterationTag**

doAfterBody



Methods from interface `javax.servlet.jsp.tagext.Tag`

`doEndTag`, `doStartTag`, `getParent`, `release`, `setPageContext`, `setParent`

## Constructor Detail

### FormattableColumn

```
public FormattableColumn()
```

## Method Detail

### addFormatter(FormatTag.Formatter) Method

```
public void addFormatter(FormatTag.Formatter formatter)
```

Add a `FormatTag` to this tag. The `Formatter` tags are used to format the value that will be written into this Grid cell. A `Formatter` tag registers itself with this `Formattable` tag using this method.

#### Parameters

*formatter*  
the `Formatter` object to add to the list of formatters

#### Related Topics

`Formattable`

---

### format(Object) Method

```
protected String format(Object value)
```

Format the text of the column with all of the registered `FormatTag` tags. Each of the formatters is called in order, chaining the value from the previous formatter into the next.

#### Parameters

*value*  
the object to format

#### Returns

*the result of applying all of the `Formatter` tags to the value `Object`.*

---

## localRelease() Method

```
protected void localRelease()
```

Reset all of the fields of this tag.

### Overrides

```
GridColumn.localRelease()
```

---

## renderStartTag(int) Method

```
protected int renderStartTag(int gridState)
```

In order for a formatted column to pick-up the formatters, it should evaluate its body. Child tags can override this behavior as necessary but should *always* evaluate the body in order to have the formatters applied. The body is evaluated as buffered so that garbage text that may be in the body of the tag is ignored.

### Overrides

```
GridColumn.renderStartTag(int)
```

### Parameters

*gridState*

the state of this column's Grid tag parent

### Returns

*EVAL\_BODY\_BUFFERED*

# GridColumn Class

*public abstract class* **GridColumn**

extends GridComponent

The abstract base class for all column tags that can be children of the Columns tag. This tag defines the basic lifecycle methods of a grid column:

- `GridColumn.renderHeaderCell()`
- `GridColumn.renderDataCell()`
- `GridColumn.renderFooterCell()`

These methods are called on a column tag as the Grid transitions between its rendering states. The `GridColumn.renderHeaderCell()` method is called in the Grid's `Grid.HEADER` state, and the `GridColumn.renderDataCell()` and `GridColumn.renderFooterCell()` methods are called during the Grid's `Grid.ROWS` and `Grid.FOOTER` lifecycle states respectively.

This tag extends `GridComponent` and overrides the valid container type for subclasses. `GridColumn` types must be nested within a `Columns` tag.

Attribute Descriptions			
Attribute	Required	Runtime Expression Evaluation	Data Bindable
<b>styleClassPrefix</b>	No	No	No
	A partial name of a CSS style class that can be used to stylize the header, data, and footer regions of a column. The rest of the actual CSS class rendered for these regions is created by appending <code>-td-header</code> , <code>-td-data</code> , and <code>-td-footer</code> , respectively to the value of the <code>styleClassPrefix</code> property.		

## Hierarchy

```

Object
  TagSupport
    BodyTagSupport
      AbstractBaseTag
        StructuredBaseTag
          GridComponent
            GridColumn
  
```

## All Implemented Interfaces

BodyTag, IterationTag, Serializable, Tag

*Direct Known Subclasses*FormattableColumn

---

## Field Summary

**Fields from `com.bea.wlw.netui.tags.databinding.base.StructuredBaseTag`**

EMPTY\_STRING

**Fields from `com.bea.wlw.netui.tags.AbstractBaseTag`**

ATTR\_GENERAL, ATTR\_GENERAL\_EXPRESSION, ATTR\_JAVASCRIPT, ATTR\_STYLE, CLASS, ID, JAVASCRIPT\_STATUS, NAME, NETUI\_UNIQUE\_CNT, ONCLICK, ONDBLCLICK, ONKEYDOWN, ONKEYPRESS, ONKEYUP, ONMOUSEDOWN, ONMOUSEMOVE, ONMOUSEOUT, ONMOUSEOVER, ONMOUSEUP, STYLE, TABINDEX

**Fields from `javax.servlet.jsp.tagext.BodyTagSupport`**

bodyContent

**Fields from `javax.servlet.jsp.tagext.TagSupport`**

id, pageContext

**Fields from interface `javax.servlet.jsp.tagext.BodyTag`**

EVAL\_BODY\_BUFFERED, EVAL\_BODY\_TAG

**Fields from interface `javax.servlet.jsp.tagext.IterationTag`**

EVAL\_BODY\_AGAIN

**Fields from interface `javax.servlet.jsp.tagext.Tag`**

EVAL\_BODY\_INCLUDE, EVAL\_PAGE, SKIP\_BODY, SKIP\_PAGE

## Constructor Summary

*GridColumn()*

## Method Summary

```

protected
    final getColumns()
        Columns        Get the Columns parent of this tag.
    public
        String getStyleClassPrefix()
            Get the CSS style class for this grid column.

protected
    Class getValidContainerType()
        Set the required parent type for this tag to the Columns tag.

protected
    void localRelease()
        Reset all of the fields of this tag.

protected
    void prepare()
        Implementation of the StructuredBaseTag.prepare() method.

protected
    String renderCloseCellTag()

protected
    abstract renderDataCell()
        String        Render the contents of this GridColumn for each dataset item into each cell in the
                       column.

protected
    String renderDataCellTag()

protected
    final int renderEndTag(int state)
        Implementation of the StructuredBaseTag.renderStartTag(int) method for all
        GridColumn subclasses.

protected
    abstract renderFooterCell()
        String        Render the contents of this GridColumn in the footer cell.

protected
    String renderFooterCellTag()

protected
    abstract renderHeaderCell()
        String        Render the contents of this GridColumn in the header cell.

protected
    String renderHeaderCellTag()

protected
    String renderOpenCellTag()

protected
    int renderStartTag(int gridState)
        Default implementation of the StructuredBaseTag.renderStartTag(int) method for all
        GridColumns that returns SKIP_BODY.

```

```
public setStyleClassPrefix(String styleClassPrefix)
void          Set the style class prefix that is used to render table cell style information for the
               header, data, and footer cells of this HTML table column.
```

**Methods from com.bea.wlw.netui.tags.databinding.grid.GridComponent**

```
getCurrentRenderState, getGrid, getGridContext,
```

**Methods from com.bea.wlw.netui.tags.databinding.base.StructuredBaseTag**

```
addContent, doEndTag, doStartTag, getContent, getCurrentRenderState,
getValidContainerType, localRelease, prepare, verifyAttributes,
verifyStructure
```

**Methods from com.bea.wlw.netui.tags.AbstractBaseTag**

```
addTagIdMapping, evaluateAttributeToString, filter, filter,
getJavaScriptUtils, getNearestForm, getScriptReporter, getTagName,
getUserLocale, localRelease, registerTagError, release, renderAttribute,
rewriteName, write
```

**Methods from javax.servlet.jsp.tagext.BodyTagSupport**

```
doAfterBody, doEndTag, doInitBody, doStartTag, getBodyContent,
getPreviousOut, release, setBodyContent
```

**Methods from javax.servlet.jsp.tagext.TagSupport**

```
doAfterBody, doEndTag, doStartTag, findAncestorWithClass, getId,
getParent, getValue, getValues, release, removeValue, setId,
setPageContext, setParent, setValue
```

**Methods from class java.lang.Object**

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll,
toString, wait, wait, wait
```

**Methods from interface javax.servlet.jsp.tagext.BodyTag**

```
doInitBody, setBodyContent
```

**Methods from interface javax.servlet.jsp.tagext.IterationTag**

```
doAfterBody
```

**Methods from interface javax.servlet.jsp.tagext.Tag**

```
doEndTag, doStartTag, getParent, release, setPageContext, setParent
```

## Constructor Detail

### GridColumn

```
public GridColumn()
```

## Method Detail

### getColumns() Method

```
protected final Columns getColumns()
```

Get the Columns parent of this tag.

---

### getStyleClassPrefix() Method

```
public String getStyleClassPrefix()
```

Get the CSS style class for this grid column. The value of this attribute is used as the prefix for the actual CSS class that is rendered for each part of the column, the header, data cells, and footer. The suffixes `-td-header`, `-td-data`, and `-td-footer` are appended to this style class prefix to create the style classes that should appear in a CSS.

#### Returns

*the CSS style class for this column*

---

### getValidContainerType() Method

```
protected Class getValidContainerType()
```

Set the required parent type for this tag to the Columns tag. A GridColumn used outside of this required parent tag will report an error.

#### Overrides

```
GridComponent.getValidContainerType()
```

#### Returns

*a class that is a valid container type for this tag*

---

## localRelease() Method

```
protected void localRelease()
```

Reset all of the fields of this tag.

### *Overrides*

```
GridComponent.localRelease()
```

---

## prepare() Method

```
protected void prepare()
```

Implementation of the StructuredBaseTag.prepare() method. This method finds the parent Columns tag and references it locally so that subclasses can access the Columns tag easily.

### *Overrides*

```
GridComponent.prepare()
```

---

## renderCloseCellTag() Method

```
protected String renderCloseCellTag()
```

---

## renderDataCell() Method

```
protected abstract String renderDataCell()  
    throws JspException
```

Render the contents of this GridColumn for each dataset item into each cell in the column.

### **Exceptions**

*JspException*  
if an error occurs that can not be reported in the page

---

## renderDataCellTag() Method

```
protected String renderDataCellTag()
```

---

## renderEndTag(int) Method

```
protected final int renderEndTag(int state)  
    throws JspException
```

Implementation of the StructuredBaseTag.renderStartTag(int) method for all GridColumn subclasses. This method performs the rendering lifecycle for all GridColumn instances; depending on the Grid's current render



state, specified with the `state` parameter, a different method is called on the subclasses to render a different part of a `GridColumn`.

Grid Lifecycle State	GridColumn Lifecycle Method
HEADER	<code>GridColumn.renderHeaderCell()</code>
ROWS	<code>GridColumn.renderDataCell()</code>
FOOTER	<code>GridColumn.renderFooterCell()</code>

Subclasses implements these methods in order to render their implementation specific contents into the content buffer of the parent `Columns` tag.

### Overrides

`StructuredBaseTag.renderEndTag(int)`

### Parameters

*state*  
the current rendering state of the containing Grid

### Returns

*EVAL\_PAGE*

### Exceptions

*JspException*  
if an error occurs that can not be reported on the page

---

## renderFooterCell() Method

```
protected abstract String renderFooterCell()
    throws JspException
```

Render the contents of this `GridColumn` in the footer cell. The footer cell is the last HTML cell in the column.

### Exceptions

*JspException*  
if an error occurs that can not be reported in the page

---

## renderFooterCellTag() Method

```
protected String renderFooterCellTag()
```

---

## renderHeaderCell() Method

```
protected abstract String renderHeaderCell()
    throws JspException
```

Render the contents of this `GridColumn` in the header cell. The header cell is the first HTML cell in the column.

### Exceptions

*JspException*

if an error occurs that can not be reported in the page

---

### renderHeaderCellTag() Method

```
protected String renderHeaderCellTag()
```

---

### renderOpenCellTag() Method

```
protected String renderOpenCellTag()
```

---

### renderStartTag(int) Method

```
protected int renderStartTag(int gridState)
```

Default implementation of the `StructuredBaseTag.renderStartTag(int)` method for all `GridColumn`s that returns `SKIP_BODY`. Subclasses that need to process their body, for example to read formatters or URL parameters should override this method.

#### Overrides

`StructuredBaseTag.renderStartTag(int)`

### Parameters

*gridState*

the current rendering state of the containing Grid

### Returns

`SKIP_BODY`;

---

### setStyleClassPrefix(String) Method

```
public void setStyleClassPrefix(String styleClassPrefix)
```

Set the style class prefix that is used to render table cell style information for the header, data, and footer cells of this HTML table column. The `styleClassPrefix` will be used in conjunction with a suffix to create style class names that are rendered inside of the `<td>` tags for each cell in this column. The value of this property overrides any value set on the `Columns.setStyleClassPrefix(String)` tag.

Assuming the value of this property is `foo`, the style class names rendered for each part of the column are:

Column Region	Style class
Header	foo-td-header
Data	foo-td-data
Footer	foo-td-footer

### Parameters

#### *styleClassPrefix*

the name of a style class that will be qualified with a suffix to name a CSS style class rendered for the header, data, and footer regions in a grid column.

# ImageColumn Class

*public class ImageColumn*

extends `SortFilterColumn`

A column that renders an image in HTML into each data cell in a column. The image that is rendered is specified with the `src` attribute and can be further parameterized with attributes to constrain its width, height, alternate text, and border. If the height and width attributes are not specified, the image is rendered at its actual size.

## Required resources:

The JavaScript file `_grid.js` must be included in any JSP page with a grid that uses a filter-enabled `ImageColumn`. The Page Flow Wizard adds the `_grid.js` file to a project's `/resources/grid` directory, when you build a page flow over a database control. The JSP should contain:

```
<script language="javascript" src="/resources/grid/_grid.js" type="text/javascript"></script>
```

Attribute Descriptions			
Attribute	Required	Runtime Expression Evaluation	Data Bindable
<b>alt</b>	No	Yes	No
	The alternate text displayed in the browser if the image cannot be displayed.		
<b>border</b>	No	Yes	No
	The number of pixels to display in a border around this image. The default border is zero.		
<b>height</b>	No	Yes	Yes
	The height of the image. If you omit the height and width attributes, the graphic's actual size will be used.		
<b>src</b>	Yes	Yes	Yes
	The source of the image to display.		
<b>width</b>	No	Yes	Yes
	The width of the image. If you omit the height and width attributes, the graphic's actual size will be used.		

## Related Topics

BinaryFlow Sample

Grid

---

***Hierarchy***

```

Object
  TagSupport
    BodyTagSupport
      AbstractBaseTag
        StructuredBaseTag
          GridComponent
            GridColumn
              FormattableColumn
                SortFilterColumn
                  ImageColumn

```

***All Implemented Interfaces***

```

BodyTag, Formattable, IterationTag, Serializable, Tag

```

***Direct Known Subclasses***

```

AnchorColumn

```

---

## Field

## Summary

**Fields from `com.bea.wlw.netui.tags.databinding.base.StructuredBaseTag`**

```

EMPTY_STRING

```

**Fields from `com.bea.wlw.netui.tags.AbstractBaseTag`**

```

ATTR_GENERAL, ATTR_GENERAL_EXPRESSION, ATTR_JAVASCRIPT, ATTR_STYLE,
CLASS, ID, JAVASCRIPT_STATUS, NAME, NETUI_UNIQUE_CNT, ONCLICK,
ONDBLCLICK, ONKEYDOWN, ONKEYPRESS, ONKEYUP, ONMOUSEDOWN, ONMOUSEMOVE,
ONMOUSEOUT, ONMOUSEOVER, ONMOUSEUP, STYLE, TABINDEX

```

**Fields from `javax.servlet.jsp.tagext.BodyTagSupport`**

```

bodyContent

```

**Fields from `javax.servlet.jsp.tagext.TagSupport`**

```

id, pageContext

```

**Fields from interface `javax.servlet.jsp.tagext.BodyTag`**

```

EVAL_BODY_BUFFERED, EVAL_BODY_TAG

```

**Fields from interface `javax.servlet.jsp.tagext.IterationTag`**

EVAL\_BODY\_AGAIN

### Fields from interface javax.servlet.jsp.tagext.Tag

EVAL\_BODY\_INCLUDE, EVAL\_PAGE, SKIP\_BODY, SKIP\_PAGE

## Constructor Summary

*ImageColumn()*

## Method Summary

```

public
String getAlt()
    Get the alternate text for the image.

public
    int getBorder()
        Get the width in pixels of the border to draw around the image.

public
String getHeight()
    Get the height of the image.

public
String getSrc()
    Get the source of the image to display.

public
String getTagName()
    Get the name of this tag.

public
String getWidth()
    Get the width of the image.

protected
    void localRelease()
        Reset all of the fields of this tag.

protected
    void prepare()
        Perform initialization for this tag.

public
String renderDataCell()
    Render the contents of this GridColumn for each dataset item into each cell in the
    column.

public
String renderFooterCell()
    Render the footer.

```

```

protected
    String renderImage()
protected
    int renderStartTag(int gridState)
        This tag only evaluates its body when the grid is rendering its ROWS.
public
    void setAlt(String alt)
        Set the alternate text displayed in the browser if the image can't be displayed.
public
    void setBorder(int border)
        Set an int representing the width of the border to draw around the image.
public
    void setHeight(String height)
        Set the height of the image to be displayed in pixels.
public
    void setSrc(String src)
        Set the source of the image to display.
public
    void setWidth(String width)
        Set the width of the image to be displayed in pixels.
protected
    void verifyAttributes()
        Ensure that the attributes set on the tag are valid.

```

**Methods from****com.bea.wlw.netui.tags.databinding.grid.column.SortFilterColumn**

canFilter, canSort, computeFilterAction, computeSortAction, getFilterAction, getName, getSortAction, getTitle, getTitleText, isFilterable, isSortable, renderHeaderCell, setFilterable, setFilterAction, setName, setSortable, setSortAction, setTitle,

**Methods from****com.bea.wlw.netui.tags.databinding.grid.column.FormattableColumn**

addFormatter, format, localRelease,

**Methods from com.bea.wlw.netui.tags.databinding.grid.column.GridColumn**

getColumns, getStyleClassPrefix, getValidContainerType, localRelease, prepare, renderCloseCellTag, renderDataCellTag, renderEndTag, renderFooterCellTag, renderHeaderCell, renderHeaderCellTag, renderOpenCellTag, renderStartTag, setStyleClassPrefix

**Methods from com.bea.wlw.netui.tags.databinding.grid.GridComponent**

getCurrentRenderState, getGrid, getGridContext, getValidContainerType, localRelease, prepare

**Methods from `com.bea.wlw.netui.tags.databinding.base.StructuredBaseTag`**

`addContent`, `doEndTag`, `doStartTag`, `getContent`, `getCurrentRenderState`, `getValidContainerType`, `localRelease`, `prepare`, `renderEndTag`, `renderStartTag`, `verifyAttributes`, `verifyStructure`

**Methods from `com.bea.wlw.netui.tags.AbstractBaseTag`**

`addTagIdMapping`, `evaluateAttributeToString`, `filter`, `filter`, `getJavaScriptUtils`, `getNearestForm`, `getScriptReporter`, `getUserLocale`, `localRelease`, `registerTagError`, `release`, `renderAttribute`, `rewriteName`, `write`

**Methods from `javax.servlet.jsp.tagext.BodyTagSupport`**

`doAfterBody`, `doEndTag`, `doInitBody`, `doStartTag`, `getBodyContent`, `getPreviousOut`, `release`, `setBodyContent`

**Methods from `javax.servlet.jsp.tagext.TagSupport`**

`doAfterBody`, `doEndTag`, `doStartTag`, `findAncestorWithClass`, `getId`, `getParent`, `getValue`, `getValues`, `release`, `removeValue`, `setId`, `setPageContext`, `setParent`, `setValue`

**Methods from class `java.lang.Object`**

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

**Methods from interface `javax.servlet.jsp.tagext.BodyTag`**

`doInitBody`, `setBodyContent`

**Methods from interface `com.bea.wlw.netui.tags.html.Formattable`**

`addFormatter`

**Methods from interface `javax.servlet.jsp.tagext.IterationTag`**

`doAfterBody`

**Methods from interface `javax.servlet.jsp.tagext.Tag`**

`doEndTag`, `doStartTag`, `getParent`, `release`, `setPageContext`, `setParent`

## Constructor Detail

### ImageColumn

```
public ImageColumn()
```



## Method Detail

### getAlt() Method

```
public String getAlt()
```

Get the alternate text for the image.

#### Returns

*the alternate text*

---

### getBorder() Method

```
public int getBorder()
```

Get the width in pixels of the border to draw around the image.

#### Returns

*the pixel width int; the default is zero.*

---

### getHeight() Method

```
public String getHeight()
```

Get the height of the image.

#### Returns

*the fixed height of the image*

---

### getSrc() Method

```
public String getSrc()
```

Get the source of the image to display.

#### Returns

*the source of an image*

---

## getTagName() Method

```
public String getTagName()
```

Get the name of this tag. This is used to identify the type of this tag for reporting tag errors.

### *Overrides*

```
AbstractBaseTag.getTagName()
```

### Returns

*a constant String representing the name of this tag.*

---

## getWidth() Method

```
public String getWidth()
```

Get the width of the image.

### Returns

*the fixed width of the image*

---

## localRelease() Method

```
protected void localRelease()
```

Reset all of the fields of this tag.

### *Overrides*

```
SortFilterColumn.localRelease()
```

---

## prepare() Method

```
protected void prepare()
```

Perform initialization for this tag.

### *Overrides*

```
SortFilterColumn.prepare()
```

---

## renderDataCell() Method

```
public String renderDataCell()  
    throws JspException
```

*Description copied from* `GridColumn.renderDataCell()`

Render the contents of this `GridColumn` for each dataset item into each cell in the column.

**Overrides**

`GridColumn.renderDataCell()`

**Exceptions**

*JspException*

if an error occurred that can not be reported in the page

---

**renderFooterCell() Method**

`public String renderFooterCell()`

Render the footer. This footer contains no text.

**Overrides**

`GridColumn.renderFooterCell()`

**Returns**

*the empty string*

---

**renderImage() Method**

`protected String renderImage()  
throws JspException`

**Exceptions**

*JspException*

---

**renderStartTag(int) Method**

`protected int renderStartTag(int gridState)`

This tag only evaluates its body when the grid is rendering its ROWS.

**Overrides**

`FormattableColumn.renderStartTag(int)`

**Parameters**

*gridState*

the state of this column's Grid tag parent

## Returns

*EVAL\_BODY\_BUFFERED*

---

## setAlt(String) Method

```
public void setAlt(String alt)
```

Set the alternate text displayed in the browser if the image can't be displayed.

### Parameters

*alt*

the alternate text

---

## setBorder(int) Method

```
public void setBorder(int border)
```

Set an int representing the width of the border to draw around the image.

### Parameters

*border*

boolean for image border drawing

---

## setHeight(String) Method

```
public void setHeight(String height)
```

Set the height of the image to be displayed in pixels.

### Parameters

*height*

the fixed height of the image

---

## setSrc(String) Method

```
public void setSrc(String src)
```

Set the source of the image to display.

**Parameters***src*the source of an image

---

**setWidth(String) Method**

```
public void setWidth(String width)
```

Set the width of the image to be displayed in pixels.

**Parameters***width*the fixed width of the image

---

**verifyAttributes() Method**

```
protected void verifyAttributes()  
    throws JspException
```

*Description copied from SortFilterColumn.verifyAttributes()*

Ensure that the attributes set on the tag are valid. The requirements are:

- If the column is determined to be filterable using the `SortFilterColumn.canFilter()` method, the `SortFilterColumn.computeFilterAction()` must return an action.
- If the column is determined to be sortable using the `SortFilterColumn.canSort()` method, the `SortFilterColumn.computeSortAction()` must return an action.

**Overrides**`SortFilterColumn.verifyAttributes()`**Exceptions***JspException*

if an error occurs that can not be reported in the page

## SortFilterColumn Class

*public abstract class SortFilterColumn*

extends `FormattableColumn`

The abstract base class of a grid column that is sortable and / or filterable. Depending on the type of the data set to which a grid tag is bound, the data set may be sortable and / or filterable. Columns of this type have attributes that determine these capabilities by using the attributes set on a `GridColumn` tag and the parent `Columns` tag.

If a particular `GridColumn` is determined to be either sortable or filterable, the header of the column is rendered with HTML anchor tags (links) that will parameterize the URL to preform the sort or filter.

For a sortable column, the title of the column will be rendered as a link which, if clicked on, will set a sort in the URL parameters that preserves all previous sorts / filters and changes or sets the direction of a sort on this column.

For a filterable column, a small filter icon will be rendered next to the column name that will display a UI which allows a filter, including the filter operation and filter value, to be set on the column.

In both the sort and filter case, the URL for a current page is changed in order to display the correct view on a data set on the next page.

### Required resources:

The JavaScript file `_grid.js` must be included in any page with a `a` if the Grid includes any `SortFilterColumn` that has filtering enabled.

Attribute Descriptions			
Attribute	Required	Runtime Expression Evaluation	Data Bindable
<b>filterable</b>	No	No	No
	A boolean that enables this column to be sortable. This attribute overrides the <code>Columns.setFilterable(boolean)</code> attribute of the <code>Columns</code> tag.		
<b>filterAction</b>	No	No	No
	The action that will be executed if this <code>GridColumn</code> is sorted. This attribute overrides the <code>Columns.setFilterAction(String)</code> attribute of the <code>Columns</code> tag.		

## Page Flow API Reference

name	No	No	No	
	When rendering in <code>Grid.ROWS</code> state, this is the name of the data from the current data item that is rendered for this row.			
sortable	No	No	No	
	A boolean that enables this column to be sortable. This attribute overrides the <code>Columns.setSortable(boolean)</code> attribute of the <code>Columns</code> tag.			
sortAction	No	No	No	
	The action that will be executed if this <code>GridColumn</code> is sorted. This attribute overrides the <code>Columns.setSortAction(String)</code> attribute of the <code>Columns</code> tag.			
title		No	No	Yes
	The title that will be rendered in the header of this tag			

### Related Topics

`SortFilterService`

---

### *Hierarchy*

```

Object
  TagSupport
    BodyTagSupport
      AbstractBaseTag
        StructuredBaseTag
          GridComponent
            GridColumn
              FormattableColumn
                SortFilterColumn
  
```

### *All Implemented Interfaces*

`BodyTag`, `Formattable`, `IterationTag`, `Serializable`, `Tag`

### *Direct Known Subclasses*

`BasicColumn`, `ImageColumn`

---

## Field Summary

### Fields from `com.bea.wlw.netui.tags.databinding.base.StructuredBaseTag`

`EMPTY_STRING`

### Fields from `com.bea.wlw.netui.tags.AbstractBaseTag`

`ATTR_GENERAL`, `ATTR_GENERAL_EXPRESSION`, `ATTR_JAVASCRIPT`, `ATTR_STYLE`, `CLASS`, `ID`, `JAVASCRIPT_STATUS`, `NAME`, `NETUI_UNIQUE_CNT`, `ONCLICK`, `ONDBLCLICK`, `ONKEYDOWN`, `ONKEYPRESS`, `ONKEYUP`, `ONMOUSEDOWN`, `ONMOUSEMOVE`, `ONMOUSEOUT`, `ONMOUSEOVER`, `ONMOUSEUP`, `STYLE`, `TABINDEX`

### Fields from `javax.servlet.jsp.tagext.BodyTagSupport`

`bodyContent`

### Fields from `javax.servlet.jsp.tagext.TagSupport`

`id`, `pageContext`

### Fields from interface `javax.servlet.jsp.tagext.BodyTag`

`EVAL_BODY_BUFFERED`, `EVAL_BODY_TAG`

### Fields from interface `javax.servlet.jsp.tagext.IterationTag`

`EVAL_BODY_AGAIN`

### Fields from interface `javax.servlet.jsp.tagext.Tag`

`EVAL_BODY_INCLUDE`, `EVAL_PAGE`, `SKIP_BODY`, `SKIP_PAGE`

## Constructor Summary

*`SortFilterColumn()`*

## Method Summary

```
public
boolean canFilter()
    Determine the absolute ability of a SortFilterColumn to be
    filtered.
```



```

    public canSort()
    boolean        Determine the absolute ability of a SortFilterColumn to be
                    filtered.

    public
    String computeFilterAction()
                    Get the absolute filter action for a SortFilterColumn.

    public
    String computeSortAction()
                    Get the absolute sort action for a SortFilterColumn.

    public
    String getFilterAction()
                    Get the action that will be invoked if this column is filtered.

    public
    String getName()
                    Get the name of the data that is rendered by this column.

    public
    String getSortAction()
                    Get the action that will be invoked if this column is sorted.

    public
    String getTitle()
                    Get the title rendered in the header cell of this column.

protected
    String getTitleText()

    public
    boolean isFilterable()
                    Get whether or not this column can be filtered.

    public
    boolean isSortable()
                    Get whether or not this column can be sorted.

protected
    void localRelease()
                    Reset all of the fields of this tag.

protected
    void prepare()
                    Implementation of the StructuredBaseTag.prepare() method.

    public
    String renderHeaderCell()
                    Render the header of a column that may be sortable or filterable.

    public
    void setFilterable(boolean filterable)
                    Set whether or not this column can be filtered.

    public
    void setFilterAction(String filterAction)
                    Set the action that will be invoked if this column is sorted.

    public
    void setName(String name)
                    Set the name of the data that is referenced by this column.

```

```

public setSortable(boolean sortable)
    void          Set whether or not this column can be sorted.
public
    void setSortAction(String sortAction)
        Set the action that will be invoked if this column is sorted.
public
    void setTitle(String title)
        Set the title rendered in the header cell of this column.
protected
    void verifyAttributes()
        Ensure that the attributes set on the tag are valid.

```

**Methods from****com.bea.wlw.netui.tags.databinding.grid.column.FormatTableColumn**

addFormatter, format, renderStartTag

**Methods from com.bea.wlw.netui.tags.databinding.grid.column.GridColumn**

getColumns, getStyleClassPrefix, getValidContainerType, localRelease, renderCloseCellTag, renderDataCell, renderDataCellTag, renderEndTag, renderFooterCell, renderFooterCellTag, renderHeaderCellTag, renderOpenCellTag, renderStartTag, setStyleClassPrefix

**Methods from com.bea.wlw.netui.tags.databinding.grid.GridComponent**

getCurrentRenderState, getGrid, getGridContext, getValidContainerType, localRelease, prepare

**Methods from com.bea.wlw.netui.tags.databinding.base.StructuredBaseTag**

addContent, doEndTag, doStartTag, getContent, getCurrentRenderState, getValidContainerType, localRelease, prepare, renderEndTag, renderStartTag, verifyStructure

**Methods from com.bea.wlw.netui.tags.AbstractBaseTag**

addTagIdMapping, evaluateAttributeToString, filter, filter, getJavaScriptUtils, getNearestForm, getScriptReporter, getTagName, getUserLocale, localRelease, registerTagError, release, renderAttribute, rewriteName, write

**Methods from javax.servlet.jsp.tagext.BodyTagSupport**

doAfterBody, doEndTag, doInitBody, doStartTag, getBodyContent, getPreviousOut, release, setBodyContent

**Methods from javax.servlet.jsp.tagext.TagSupport**

doAfterBody, doEndTag, doStartTag, findAncestorWithClass, getId, getParent, getValue, getValues, release, removeValue, setId,

`setPageContext, setParent, setValue`

#### Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

#### Methods from interface `javax.servlet.jsp.tagext.BodyTag`

`doInitBody, setBodyContent`

#### Methods from interface `com.bea.wlw.netui.tags.html.Formattable`

`addFormatter`

#### Methods from interface `javax.servlet.jsp.tagext.IterationTag`

`doAfterBody`

#### Methods from interface `javax.servlet.jsp.tagext.Tag`

`doEndTag, doStartTag, getParent, release, setPageContext, setParent`

## Constructor Detail

### SortFilterColumn

```
public SortFilterColumn()
```

## Method Detail

### canFilter() Method

```
public boolean canFilter()
```

Determine the absolute ability of a `SortFilterColumn` to be filtered. The capability of a column to be filterable is determined as a combination of the sortable attributes on this tag as well as the sortable attribute on the `Columns` tag. The rules for determining whether a column is filterable are:

- If the filterable attribute is unset, the `Columns` tag's filterable attribute is used.
- If the filterable attribute is used, the filterable attribute is used.

---

### canSort() Method

```
public boolean canSort()
```

Determine the absolute ability of a `SortFilterColumn` to be filtered. The capability of a column to be sortable is determined as a combination of the sortable attributes on this tag as well as the sortable attribute on the `Columns` tag. The rules for determining whether a column is sortable are:

- If the sortable attribute is unset, the `Columns` tag's sortable attribute is used.
  - If the sortable attribute is used, the sortable attribute is used.
- 

### **computeFilterAction() Method**

```
public String computeFilterAction()
```

Get the absolute filter action for a `SortFilterColumn`. This filter action is determined by combining the filter action on this tag as well as the filter action on the `Columns` tag. The rules for determining the filter action are:

- If the filter action on this column is set, this attribute is used.
  - Otherwise, the `Columns` tag's filter action attribute is used.
- 

### **computeSortAction() Method**

```
public String computeSortAction()
```

Get the absolute sort action for a `SortFilterColumn`. This sort action is determined by combining the sort action on this tag as well as the sort action on the `Columns` tag. The rules for determining the sort action are:

- If the sort action on this column is set, this attribute is used.
  - Otherwise, the `Columns` tag's sort action attribute is used.
- 

### **getFilterAction() Method**

```
public String getFilterAction()
```

Get the action that will be invoked if this column is filtered.

#### **Returns**

*the action that will be called to filter the column*

---

### **getName() Method**

```
public String getName()
```

Get the name of the data that is rendered by this column.

## Returns

*the name of the data to render*

---

## getSortAction() Method

```
public String getSortAction()
```

Get the action that will be invoked if this column is sorted.

## Returns

*the action that will be called to sort the column*

---

## getTitle() Method

```
public String getTitle()
```

Get the title rendered in the header cell of this column.

## Returns

*the title of the colum*

---

## getTitleText() Method

```
protected String getTitleText()
```

---

## isFilterable() Method

```
public boolean isFilterable()
```

Get whether or not this column can be filtered.

## Returns

*true if this column is filterable; false otherwise.*

---

## isSortable() Method

```
public boolean isSortable()
```

Get whether or not this column can be sorted.

**Returns**

*true if this column is sortable; false otherwise.*

---

**localRelease() Method**

```
protected void localRelease()
```

Reset all of the fields of this tag.

**Overrides**

```
FormattableColumn.localRelease()
```

---

**prepare() Method**

```
protected void prepare()
```

*Description copied from GridColumn.prepare()*

Implementation of the StructuredBaseTag.prepare() method. This method finds the parent Columns tag and references it locally so that subclasses can access the Columns tag easily.

**Overrides**

```
GridColumn.prepare()
```

---

**renderHeaderCell() Method**

```
public String renderHeaderCell()  
    throws JspException
```

Render the header of a column that may be sortable or filterable. Sorting and filtering on a Grid is performed through the use of a URL syntax that is interpreted into statements that can sort or filter the grid's dataset. The links to sort or filter a grid column are rendered in the header of the column. If a column is sortable, the title of the column will be rendered as a link that, when pressed, will sort the column in an order opposite the order of the current sort. If a column is filterable, an image anchor will be rendered as a link that, when pressed, will filter the column using the operation and values that are provided in the URL.

**Overrides**

```
GridColumn.renderHeaderCell()
```

**Returns**

*the String content of this header cell*

**Exceptions**

*JspException*

if an error occurs that can not be shown in the page

---

## setFilterable(boolean) Method

```
public void setFilterable(boolean filterable)
```

Set whether or not this column can be filtered.

### Parameters

*filterable*

a boolean that sets this capability. If `true` this column may be filterable; otherwise it is not.

---

## setFilterAction(String) Method

```
public void setFilterAction(String filterAction)
```

Set the action that will be invoked if this column is sorted. The use of this attribute requires the presence of a `PageFlowController` in the JSP's directory.

### Parameters

*filterAction*

the action that will be called to filter the column

---

## setName(String) Method

```
public void setName(String name)
```

Set the name of the data that is referenced by this column. Because data is extracted from the "current" data item by name, the name attribute is used to specify which data should be extracted for a given column. This attribute is usually used when rendering the value for a cell that is rendered in the Grid's `Grid.ROWS` render state.

### Parameters

*name*

the name of the data to extract from the current data item.

---

## setSortable(boolean) Method

```
public void setSortable(boolean sortable)
```

Set whether or not this column can be sorted.

### Parameters

*sortable*

a boolean that sets this capability. If `true` this column may be filterable; otherwise it is not.

## setSortAction(String) Method

```
public void setSortAction(String sortAction)
```

Set the action that will be invoked if this column is sorted. The use of this attribute requires the presence of a `PageFlowController` in the JSP's directory.

### Parameters

*sortAction*  
the action that will be called to sort the column

---

## setTitle(String) Method

```
public void setTitle(String title)
```

Set the title rendered in the header cell of this column.

### Parameters

*title*  
the title of the column

---

## verifyAttributes() Method

```
protected void verifyAttributes()  
    throws JspException
```

Ensure that the attributes set on the tag are valid. The requirements are:

- If the column is determined to be filterable using the `SortFilterColumn.canFilter()` method, the `SortFilterColumn.computeFilterAction()` must return an action.
- If the column is determined to be sortable using the `SortFilterColumn.canSort()` method, the `SortFilterColumn.computeSortAction()` must return an action.

### Overrides

`StructuredBaseTag.verifyAttributes()`

### Exceptions

*JspException*  
if an error occurs that can not be reported in the page



# **com.bea.wlw.netui.tags.databinding.grid.pager Package**

## **Class Summary**

Pager

A navigation pager for the grid.

# Pager Class

*public class **Pager***

extends `GridComponent`

A navigation pager for the grid. This feature allows the grid to render a subset of data on each page and renders the navigation links for moving between grid pages. The pager can be rendered before the grid table, after the grid table, or both by setting the `renderInHeader`, `renderInFooter`, or both of these attributes, respectively.

The rendered pager provides a reference for the current page and navigation links in the default format of:

Page # of # [Prev] [Next]

The "Page # of #" is computed using the current page, the page size, and the data set size. If there is no data, the pager will read "Page 1 of 1". The "Prev" and "Next" links are optional depending on the size of the data set, the current page, and the page size. The rules are:

- If on the first page, the "Prev" link is not shown.
- If on the last page, the "Next" link is not shown.
- If on a single page Grid, neither link is shown.

Attribute Descriptions			
Attribute	Required	Runtime Expression Evaluation	Data Bindable
<b>action</b>	No	No	No
	The action to use when paging between grid pages. This is rendered on the HTML anchor that is written for all paging links.		
<b>pageSize</b>	No	No	No
	The number of data items to render on a page of a Grid.		
<b>renderInFooter</b>	No	No	No
	A boolean that will render pager before the Grid starts the data table.		
<b>renderInHeader</b>	No	No	No
	A boolean that will render a pager after the Grid closes the data table.		

## Related Topics

Grid

---

***Hierarchy***

```

Object
  TagSupport
    BodyTagSupport
      AbstractBaseTag
        StructuredBaseTag
          GridComponent
            Pager

```

***All Implemented Interfaces***

```

BodyTag, IterationTag, Serializable, Tag

```

---

## Field Summary

**Fields from `com.bea.wlw.netui.tags.databinding.base.StructuredBaseTag`**

```

EMPTY_STRING

```

**Fields from `com.bea.wlw.netui.tags.AbstractBaseTag`**

```

ATTR_GENERAL, ATTR_GENERAL_EXPRESSION, ATTR_JAVASCRIPT, ATTR_STYLE,
CLASS, ID, JAVASCRIPT_STATUS, NAME, NETUI_UNIQUE_CNT, ONCLICK,
ONDBLCLICK, ONKEYDOWN, ONKEYPRESS, ONKEYUP, ONMOUSEDOWN, ONMOUSEMOVE,
ONMOUSEOUT, ONMOUSEOVER, ONMOUSEUP, STYLE, TABINDEX

```

**Fields from `javax.servlet.jsp.tagext.BodyTagSupport`**

```

bodyContent

```

**Fields from `javax.servlet.jsp.tagext.TagSupport`**

```

id, pageContext

```

**Fields from interface `javax.servlet.jsp.tagext.BodyTag`**

```

EVAL_BODY_BUFFERED, EVAL_BODY_TAG

```

**Fields from interface `javax.servlet.jsp.tagext.IterationTag`**

```

EVAL_BODY_AGAIN

```

**Fields from interface `javax.servlet.jsp.tagext.Tag`**

```

EVAL_BODY_INCLUDE, EVAL_PAGE, SKIP_BODY, SKIP_PAGE

```

## Constructor

## Summary

*Pager()*

## Method

## Summary

```

public
String getAction()
    Get the action that will be invoked if the containing Grid tag is paged.

public
int getPageSize()
    Get the number of data rows that will be displayed in a Grid page.

public
boolean getRenderInFooter()
    Get a boolean that describes whether the pager will render at the end of the Grid's
    table.

public
boolean getRenderInHeader()
    Get a boolean that describes whether the pager will render before the Grid's table.

public
String getTagName()
    Get the name of this tag.

protected
void localRelease()
    Reset all of the fields of this tag.

protected
int renderEndTag(int state)
    Render a pager for the containing Grid.

protected
int renderStartTag(int state)
    Render the start of the Pager.

public
void setAction(String action)
    Set the action that will be invoked if the containing Grid tag is paged.

public
void setPageSize(int pageSize)
    Set the number of data rows that will be displayed in a Grid.

public
void setRenderInFooter(boolean renderInFooter)
    Set a boolean that enables the Pager to render after the Grid has closed the Grid's
    table.

public
void setRenderInHeader(boolean renderInHeader)

```

Set a boolean that enables the Pager to render before the Grid has opened the Grid's table.

### Methods from `com.bea.wlw.netui.tags.databinding.grid.GridComponent`

`getCurrentRenderState`, `getGrid`, `getGridContext`, `getValidContainerType`, `prepare`

### Methods from `com.bea.wlw.netui.tags.databinding.base.StructuredBaseTag`

`addContent`, `doEndTag`, `doStartTag`, `getContent`, `getCurrentRenderState`, `getValidContainerType`, `localRelease`, `prepare`, `verifyAttributes`, `verifyStructure`

### Methods from `com.bea.wlw.netui.tags.AbstractBaseTag`

`addTagIdMapping`, `evaluateAttributeToString`, `filter`, `filter`, `getJavaScriptUtils`, `getNearestForm`, `getScriptReporter`, `getUserLocale`, `localRelease`, `registerTagError`, `release`, `renderAttribute`, `rewriteName`, `write`

### Methods from `javax.servlet.jsp.tagext.BodyTagSupport`

`doAfterBody`, `doEndTag`, `doInitBody`, `doStartTag`, `getBodyContent`, `getPreviousOut`, `release`, `setBodyContent`

### Methods from `javax.servlet.jsp.tagext.TagSupport`

`doAfterBody`, `doEndTag`, `doStartTag`, `findAncestorWithClass`, `getId`, `getParent`, `getValue`, `getValues`, `release`, `removeValue`, `setId`, `setPageContext`, `setParent`, `setValue`

### Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

### Methods from interface `javax.servlet.jsp.tagext.BodyTag`

`doInitBody`, `setBodyContent`

### Methods from interface `javax.servlet.jsp.tagext.IterationTag`

`doAfterBody`

### Methods from interface `javax.servlet.jsp.tagext.Tag`

`doEndTag`, `doStartTag`, `getParent`, `release`, `setPageContext`, `setParent`

## Constructor Detail

### Pager

```
public Pager()
```

## Method Detail

### getAction() Method

```
public String getAction()
```

Get the action that will be invoked if the containing Grid tag is paged.

#### Returns

*the name of the action to invoke in order to Page a grid*

---

### getPageSize() Method

```
public int getPageSize()
```

Get the number of data rows that will be displayed in a Grid page.

#### Returns

*the number of rows to render*

---

### getRenderInFooter() Method

```
public boolean getRenderInFooter()
```

Get a boolean that describes whether the pager will render at the end of the Grid's table.

#### Returns

*true if a pager will render at the end of the grid; false otherwise.*

---

### getRenderInHeader() Method

```
public boolean getRenderInHeader()
```

Get a boolean that describes whether the pager will render before the Grid's table.

**Returns**

*true if a pager will render before the grid; false otherwise.*

---

**getTagName() Method**

```
public String getTagName()
```

Get the name of this tag. This is used to identify the type of this tag for reporting tag errors.

**Overrides**

```
AbstractBaseTag.getTagName()
```

**Returns**

*a constant String representing the name of this tag.*

---

**localRelease() Method**

```
protected void localRelease()
```

Reset all of the fields of this tag.

**Overrides**

```
GridComponent.localRelease()
```

---

**renderEndTag(int) Method**

```
protected int renderEndTag(int state)
    throws JspException
```

Render a pager for the containing Grid. If the grid is rendered in either the header or footer, the URLs that are used to page to the next or previous pages are computed and rendered given the appropriate grid state. If the grid state is `Grid.HEAD_PAGER`, and the pager should be rendered in the header, the pager is rendered; if the grid state is `Grid.FOOT_PAGER`, and the pager should be rendered in the footer, the pager is rendered.

**Overrides**

```
StructuredBaseTag.renderEndTag(int)
```

**Parameters**

*state*

the containing Grid's current render state

**Returns**

*EVAL\_PAGE*

## Exceptions

*JspException*

---

## renderStartTag(int) Method

```
protected int renderStartTag(int state)
```

Render the start of the Pager.

### Overrides

StructuredBaseTag.renderStartTag(int)

### Parameters

*state*

the containing Grid's current render state

### Returns

*SKIP\_BODY*

---

## setAction(String) Method

```
public void setAction(String action)
```

Set the action that will be invoked if the containing Grid tag is paged. The use of this attribute is required by the tag, and a PageFlow must be present in the JSP's directory in order to page the grid.

### Parameters

*action*

the name of the action to invoke in order to Page a grid

---

## setPageSize(int) Method

```
public void setPageSize(int pageSize)
```

Set the number of data rows that will be displayed in a Grid. The default page size is 10 rows.

### Parameters

*pageSize*

the number of rows to render in a Grid page.

---



## setRenderInFooter(boolean) Method

```
public void setRenderInFooter(boolean renderInFooter)
```

Set a boolean that enables the Pager to render after the Grid has closed the Grid's table. The default value of this attribute is true.

### Parameters

*renderInFooter*

true to render the pager at the end of the grid; false othrewise.

---

## setRenderInHeader(boolean) Method

```
public void setRenderInHeader(boolean renderInHeader)
```

Set a boolean that enables the Pager to render before the Grid has opened the Grid's table. The default value of this attribute is false.

### Parameters

*renderInHeader*

true to render the pager before the grid; false otherwise.

# **com.bea.wlw.netui.tags.databinding.grid.style Package**

## **Class Summary**

GridStyle	A tag that allows parameterization of the style components of the HTML table that a Grid renders.
GridStyleContext	A JavaBean that contains the style information about a Grid.

# GridStyle Class

*public class GridStyle*

extends GridComponent

A tag that allows parameterization of the style components of the HTML table that a Grid renders. Styles are rendered in several locations in the table that the Grid renders. See the `Grid` tag description for the structure of the table that the Grid renders. The styles defined here are mapped to these regions as follows:

## Grid Region Name   HTML Element   Style Attribute

Table	table tag	tableClass
Header	table row tag	headerClass
Data Row	table row tag	rowClass
Footer	table row tag	footerClass

Attribute Descriptions			
Attribute	Required	Runtime Expression Evaluation	Data Bindable
styleClassPrefix	No	No	No
	The CSS style prefix that will be used to create the style class names for regions in the Grid.		

## Related Topics

<netui-data:grid> Tag Sample  
Grid

---

## Hierarchy

```

Object
  TagSupport
    BodyTagSupport
      AbstractBaseTag
        StructuredBaseTag
          GridComponent
            GridStyle
  
```

## All Implemented Interfaces

BodyTag, IterationTag, Serializable, Tag

---

## Field Summary

### Fields from `com.bea.wlw.netui.tags.databinding.base.StructuredBaseTag`

`EMPTY_STRING`

### Fields from `com.bea.wlw.netui.tags.AbstractBaseTag`

`ATTR_GENERAL`, `ATTR_GENERAL_EXPRESSION`, `ATTR_JAVASCRIPT`, `ATTR_STYLE`, `CLASS`, `ID`, `JAVASCRIPT_STATUS`, `NAME`, `NETUI_UNIQUE_CNT`, `ONCLICK`, `ONDBLCLICK`, `ONKEYDOWN`, `ONKEYPRESS`, `ONKEYUP`, `ONMOUSEDOWN`, `ONMOUSEMOVE`, `ONMOUSEOUT`, `ONMOUSEOVER`, `ONMOUSEUP`, `STYLE`, `TABINDEX`

### Fields from `javax.servlet.jsp.tagext.BodyTagSupport`

`bodyContent`

### Fields from `javax.servlet.jsp.tagext.TagSupport`

`id`, `pageContext`

### Fields from interface `javax.servlet.jsp.tagext.BodyTag`

`EVAL_BODY_BUFFERED`, `EVAL_BODY_TAG`

### Fields from interface `javax.servlet.jsp.tagext.IterationTag`

`EVAL_BODY_AGAIN`

### Fields from interface `javax.servlet.jsp.tagext.Tag`

`EVAL_BODY_INCLUDE`, `EVAL_PAGE`, `SKIP_BODY`, `SKIP_PAGE`

## Constructor Summary

*`GridStyle()`*

## Method Summary

```
public
String getStyleClassPrefix()
    Get the footer style class

public
String getTagName()
```

Get the name of this tag.

```
protected
    void localRelease()
        Reset all of the fields of this tag.

protected
    int renderEndTag(int gridState)
        Complete the tag lifecycle.

protected
    int renderStartTag(int gridState)
        Start the tag lifecycle.

public
    void setStyleClassPrefix(String styleClassPrefix)
        Set the footer style class.
```

#### Methods from **com.bea.wlw.netui.tags.databinding.grid.GridComponent**

getCurrentRenderState, getGrid, getGridContext, getValidContainerType, prepare

#### Methods from **com.bea.wlw.netui.tags.databinding.base.StructuredBaseTag**

addContent, doEndTag, doStartTag, getContent, getCurrentRenderState, getValidContainerType, localRelease, prepare, verifyAttributes, verifyStructure

#### Methods from **com.bea.wlw.netui.tags.AbstractBaseTag**

addTagIdMapping, evaluateAttributeToString, filter, filter, getJavaScriptUtils, getNearestForm, getScriptReporter, getUserLocale, localRelease, registerTagError, release, renderAttribute, rewriteName, write

#### Methods from **javax.servlet.jsp.tagext.BodyTagSupport**

doAfterBody, doEndTag, doInitBody, doStartTag, getBodyContent, getPreviousOut, release, setBodyContent

#### Methods from **javax.servlet.jsp.tagext.TagSupport**

doAfterBody, doEndTag, doStartTag, findAncestorWithClass, getId, getParent, getValue, getValues, release, removeValue, setId, setPageContext, setParent, setValue

#### Methods from class **java.lang.Object**

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

#### Methods from interface **javax.servlet.jsp.tagext.BodyTag**

doInitBody, setBodyContent

**Methods from interface `javax.servlet.jsp.tagext.IterationTag`**

`doAfterBody`

**Methods from interface `javax.servlet.jsp.tagext.Tag`**

`doEndTag`, `doStartTag`, `getParent`, `release`, `setPageContext`, `setParent`

## Constructor Detail

### GridStyle

```
public GridStyle()
```

## Method Detail

### getStyleClassPrefix() Method

```
public String getStyleClassPrefix()
```

Get the footer style class

#### Returns

*the footer style class or null if none is set*

---

### getTagName() Method

```
public String getTagName()
```

Get the name of this tag. This is used to identify the type of this tag for reporting tag errors.

#### Overrides

```
AbstractBaseTag.getTagName()
```

#### Returns

*a constant String representing the name of this tag.*

---

### localRelease() Method

```
protected void localRelease()
```

Reset all of the fields of this tag.

**Overrides**

`GridComponent.localRelease()`

---

**renderEndTag(int) Method**

`protected int renderEndTag(int gridState)`

Complete the tag lifecycle. Set the created `GridStyleContext` JavaBean on the Grid tag.

**Overrides**

`StructuredBaseTag.renderEndTag(int)`

**Parameters**

*gridState*

the current rendering state of the grid

**Returns**

`EVAL_PAGE`

---

**renderStartTag(int) Method**

`protected int renderStartTag(int gridState)`

Start the tag lifecycle.

**Overrides**

`StructuredBaseTag.renderStartTag(int)`

**Returns**

`SKIP_BODY`

---

**setStyleClassPrefix(String) Method**

`public void setStyleClassPrefix(String styleClassPrefix)`

Set the footer style class.

**Parameters**

*styleClassPrefix*

the footer style class

# GridStyleContext Class

*public class GridStyleContext*

extends AbstractStyleContext

A JavaBean that contains the style information about a Grid. Because a Grid rendering lifecycle makes several passes through the body and the GridStyle tag is created and destroyed several times during these cycles, a single JavaBean is created that contains the style information for a Grid. This JavaBean is set on the Grid using the `Grid.setGridStyle(GridStyleContext)` method.

---

## Hierarchy

```
Object
  AbstractStyleContext
    GridStyleContext
```

---

## Constructor Summary

*GridStyleContext()*

## Method Summary

```
public
String getStyleClassPrefix()
    Get the footer style class

public
String renderAlternatingRowStyle()
    A method to render an HTML table row tag with the appropriate row
    style / alternating row style information into the given
    UnsyncStringBuffer.

public
String renderFooterStyle()
    A method to render an HTML table row tag with the appropriate footer
    style information into the given UnsyncStringBuffer.

public
String renderHeaderStyle()
    A method to render an HTML table row tag with the appropriate header
    style information into the given UnsyncStringBuffer

public
String renderRowStyle()
```



```

public
String renderTableStyle()
public
    void setStyleClassPrefix(String styleClassPrefix)
        Set the footer style class.

```

**Methods from**

**com.bea.wlw.netui.tags.databinding.base.style.AbstractStyleContext**

getRowClass, getTableClass, renderStyle, setRowClass, setTableClass

**Methods from class java.lang.Object**

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructor Detail

### GridStyleContext

```
public GridStyleContext()
```

## Method Detail

### getStyleClassPrefix() Method

```
public String getStyleClassPrefix()
```

Get the footer style class

**Returns**

*the footer style class or null if none is set*

---

### renderAlternatingRowStyle() Method

```
public String renderAlternatingRowStyle()
```

A method to render an HTML table row tag with the appropriate row style / alternating row style information into the given UnsyncStringBuffer. If the alternating row style is null and a row style is set, the row style is rendered on this row.

---

## renderFooterStyle() Method

```
public String renderFooterStyle()
```

A method to render an HTML table row tag with the appropriate footer style information into the given `UnsyncStringBuffer`.

---

## renderHeaderStyle() Method

```
public String renderHeaderStyle()
```

A method to render an HTML table row tag with the appropriate header style information into the given `UnsyncStringBuffer`.

---

## renderRowStyle() Method

```
public String renderRowStyle()
```

### *Overrides*

`AbstractStyleContext.renderRowStyle()`

---

## renderTableStyle() Method

```
public String renderTableStyle()
```

### *Overrides*

`AbstractStyleContext.renderTableStyle()`

---

## setStyleClassPrefix(String) Method

```
public void setStyleClassPrefix(String styleClassPrefix)
```

Set the footer style class.

### Parameters

*styleClassPrefix*  
the footer style class

# com.bea.wlw.netui.tags.databinding.invoke Package

## Class Summary

AbstractCallMethod	An abstract base class for tags that are capable of reflectively invoking methods.
CallControl	A tag that is used to call a method on a control.
CallMethod	An abstract base class for tags that are capable of reflectively invoking methods.
CallPageFlow	A tag that is used to call a method on the current <code>PageFlowController</code> .
DeclareControl	Declare a control that is stored in the <code>PageContext</code> attribute map.
MethodParameter	A tag that is used to add an argument to a method that will be called on some object.

## Exception Summary

ObjectNotFoundException	An exception thrown when an object on which to invoke a method can not be found.
-------------------------	--

# AbstractCallMethod Class

*public abstract class AbstractCallMethod*

extends AbstractBaseTag

An abstract base class for tags that are capable of reflectively invoking methods. Specializations of this tag provide method implementations that locate the object on which to invoke the method and that handle any return value from the invoked method.

The CallMethod tag can have child tags of type MethodParameter; these tags must be in the same order as the parameter list in the method signature of the method that will be invoked. To invoke an overloaded method, the `property` must be set to the String name of the type to pass to the method. If the type attribute values on nested MethodParameter tags do not match any method signature, an error will be reported in the page.

Attribute Descriptions			
Attribute	Required	Runtime Expression Evaluation	Data Bindable
method	No	No	No
	The method to invoke on an object.		
failOnError	No	No	No
	A boolean that if <code>true</code> will cause the tag to report any method invocation errors in-line in the rendered page.		
resultId	No	No	No
	The PageContext attribute name where the result, if non-null, is stored.		

## Hierarchy

```

Object
  TagSupport
    BodyTagSupport
      AbstractBaseTag
        AbstractCallMethod
  
```

## All Implemented Interfaces

BodyTag, IterationTag, Serializable, Tag

## Direct Known Subclasses

CallControl, CallMethod

## Field Summary

### Fields from `com.bea.wlw.netui.tags.AbstractBaseTag`

`ATTR_GENERAL`, `ATTR_GENERAL_EXPRESSION`, `ATTR_JAVASCRIPT`, `ATTR_STYLE`, `CLASS`, `ID`, `JAVASCRIPT_STATUS`, `NAME`, `NETUI_UNIQUE_CNT`, `ONCLICK`, `ONDBLCLICK`, `ONKEYDOWN`, `ONKEYPRESS`, `ONKEYUP`, `ONMOUSEDOWN`, `ONMOUSEMOVE`, `ONMOUSEOUT`, `ONMOUSEOVER`, `ONMOUSEUP`, `STYLE`, `TABINDEX`

### Fields from `javax.servlet.jsp.tagext.BodyTagSupport`

`bodyContent`

### Fields from `javax.servlet.jsp.tagext.TagSupport`

`id`, `pageContext`

### Fields from interface `javax.servlet.jsp.tagext.BodyTag`

`EVAL_BODY_BUFFERED`, `EVAL_BODY_TAG`

### Fields from interface `javax.servlet.jsp.tagext.IterationTag`

`EVAL_BODY_AGAIN`

### Fields from interface `javax.servlet.jsp.tagext.Tag`

`EVAL_BODY_INCLUDE`, `EVAL_PAGE`, `SKIP_BODY`, `SKIP_PAGE`

## Constructor Summary

*`AbstractCallMethod()`*

## Method Summary

```
public
    void addParameter(String type, Object parameter)
        Add a paramter that will be passed as an argument to the method that will be invoked.

public
    int doEndTag()
        Reflectively invokes the method specified by the method attribute,
        AbstractCallMethod.getMethod().
```

```

public doStartTag()
    int          Causes the body of this tag to be rendered; only MethodParameter tags are
                  allowed to be contained inside of this tag.

protected
    Method findMethod(Object target, String methodName, boolean verifyTypes)
        The default findMethod implementation is an uncached search of all of the methods
        available on the Class of the target

    public
    boolean getFailOnError()
        Get whether or not errors invoking a method are reported in-line when the tag renders.

    public
    String getMethod()
        Gets the name of the method to invoke on the target object.

protected
    abstract getObjectName()
        String      Get the name of the object that is the target of the invocation.

protected
    List getParameterNodes()
        Internal, read-only property used by subclasses to get the list of parameters to be used
        when reflectively invoking a method.

    public
    String getResultId()
        Get the identifier at which the result of invoking the method can be found.

protected
    void handleReturnValue(Object result)
        A method that allows concrete classes to handle the result of the reflective invocation
        in an implementation specific way.

protected
    void localRelease()
        Reset all of the fields of this tag.

protected
    abstract resolveObject()
        Object      Resolve the object on which the method should be invoked.

    public
    void setFailOnError(boolean failOnError)
        Sets whether or not to report exceptions to the page when errors occur invoking a
        method on an object.

    public
    void setMethod(String method)
        Sets the name of a method to invoke on the target object.

    public
    void setResultId(String resultId)
        Sets the identifier at which the result of invoking the method will stored.

```

#### Methods from `com.bea.wlw.netui.tags.AbstractBaseTag`

`addTagIdMapping`, `evaluateAttributeToString`, `filter`, `filter`,  
`getJavaScriptUtils`, `getNearestForm`, `getScriptReporter`, `getTagName`,

`getUserLocale, registerTagError, release, renderAttribute, rewriteName, write`

### Methods from `javax.servlet.jsp.tagext.BodyTagSupport`

`doAfterBody, doInitBody, getBodyContent, getPreviousOut, release, setBodyContent`

### Methods from `javax.servlet.jsp.tagext.TagSupport`

`doAfterBody, doEndTag, doStartTag, findAncestorWithClass, getId, getParent, getValue, getValues, release, removeValue, setId, setPageContext, setParent, setValue`

### Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

### Methods from interface `javax.servlet.jsp.tagext.BodyTag`

`doInitBody, setBodyContent`

### Methods from interface `javax.servlet.jsp.tagext.IterationTag`

`doAfterBody`

### Methods from interface `javax.servlet.jsp.tagext.Tag`

`doEndTag, doStartTag, getParent, release, setPageContext, setParent`

## Constructor Detail

### AbstractCallMethod

```
public AbstractCallMethod()
```

## Method Detail

### addParameter(String, Object) Method

```
public void addParameter(String type,
                        Object parameter)
```

Add a paramter that will be passed as an argument to the method that will be invoked. This method is implemented to allow the the `MethodParameter` tags to register their parameters. This object is passed in the position that it appeared in the set of child `MethodParameter` tags.

## Parameters

*type*

a String of the type or class name of this parameter

*parameter*

an object that should be passed as an argument to the invoked method

## Related Topics

MethodParameter

---

## doEndTag() Method

```
public int doEndTag()  
    throws JspException
```

Reflectively invokes the method specified by the method attribute, `AbstractCallMethod.getMethod()`. The arguments passed to the method are taken from any nested `MethodParameter` tags. When the parameters which are added by the `MethodParameter` tags are `String` types, an attempt is made to convert each of these parameters into the type expected by the method. This conversion is done using the `TypeUtils.convertToObject(String, Class)` method. If a `String` can not be converted to the type expected by the method, an exception is thrown and the error is reported in the tag. Any return value that results from invoking the given method is passed to the subclass implementation of the method `AbstractCallMethod.handleReturnValue(Object)`.

### Overrides

```
BodyTagSupport.doEndTag()
```

## Returns

*EVAL\_PAGE* to continue evaluating the page

## Exceptions

*JspException*

if there are errors. All exceptions that may be thrown in the process of reflectively invoking the method and performing type conversion are reported as @see `javax.servlet.jsp.JspException`.

## Related Topics

`AbstractCallMethod.getMethod()`  
`AbstractCallMethod.handleReturnValue(Object)`  
`MethodParameter`  
`ObjectNotFoundException`  
`TypeUtils.convertToObject(String, Class)`  
`String`

---



## doStartTag() Method

```
public int doStartTag()
    throws JspException
```

Causes the body of this tag to be rendered; only `MethodParameter` tags are allowed to be contained inside of this tag. The body content is never rendered.

### Overrides

```
BodyTagSupport.doStartTag()
```

### Returns

*EVAL\_BODY\_BUFFERED*

### Exceptions

*JspException*

### Related Topics

`MethodParameter`

---

## findMethod(Object, String, boolean) Method

```
protected Method findMethod(Object target,
                             String methodName,
                             boolean verifyTypes)
```

The default `findMethod` implementation is an uncached search of all of the methods available on the Class of the target

### Parameters

*target*

the object from which to find the method

*methodName*

the name of the method to find

*verifyTypes*

a boolean that if true will match the type names in addition to the String method name

### Returns

*a Method object matching the methodName and types, if verifyTypes is true. null otherwise.*

---

## getFailOnError() Method

```
public boolean getFailOnError()
```

Get whether or not errors invoking a method are reported in-line when the tag renders.

### Returns

*a boolean that if true will report errors encountered during the tag's execution.*

---

## getMethod() Method

```
public String getMethod()
```

Gets the name of the method to invoke on the target object.

### Returns

*the name of the method to invoke*

---

## getObjectName() Method

```
protected abstract String getObjectNames()
```

Get the name of the object that is the target of the invocation. This is a generic method for this tag that enables more specific error reporting.

### Returns

*a name for the object on which the method will be invoked.*

---

## getParameterNodes() Method

```
protected List getParameterNodes()
```

Internal, read-only property used by subclasses to get the list of parameters to be used when reflectively invoking a method. If the method takes no parameters, this list will be of size zero.

### Returns

*the list of parameters*

---

## getResultId() Method

```
public String getResultId()
```

Get the identifier at which the result of invoking the method can be found.

## Returns

*a String where the result object can be found in the PageContext attribute map.*

---

## handleReturnValue(Object) Method

```
protected void handleReturnValue(Object result)
```

A method that allows concrete classes to handle the result of the reflective invocation in an implementation specific way.

The default behavior is to set the return value resulting from invoking the method in the PageContext attribute map of the current JSP page. The result is set as an attribute if the `result` is not null and the `CallMethod.getResultId()` String is not null. If the value returned from calling a method is null and the `CallMethod.getResultId()` is non-null, the `removeAttribute` is called to remove the attribute from the attribute map.

## Parameters

*result*

the object that was returned by calling the method on the object

---

## localRelease() Method

```
protected void localRelease()
```

Reset all of the fields of this tag.

## Overrides

```
AbstractBaseTag.localRelease()
```

---

## resolveObject() Method

```
protected abstract Object resolveObject()  
    throws ObjectNotFoundException
```

Resolve the object on which the method should be invoked. If there are errors resolving this object, this method will throw an `ObjectNotFoundException`. If the object is not found but no exception occurred, this method should return null.

## Returns

*the object on which to reflectively invoke the method.*

## Exceptions

*ObjectNotFoundException*

if an exception occurred attempting to resolve an object

---

## setFailOnError(boolean) Method

```
public void setFailOnError(boolean failOnError)
```

Sets whether or not to report exceptions to the page when errors occur invoking a method on an object.

### Parameters

*failOnError*

a boolean that defines whether or not exceptions should be thrown when invocation fails.

---

## setMethod(String) Method

```
public void setMethod(String method)
```

Sets the name of a method to invoke on the target object.

### Parameters

*method*

the name of the method to invoke

---

## setResultId(String) Method

```
public void setResultId(String resultId)
```

Sets the identifier at which the result of invoking the method will be stored.

### Parameters

*resultId*

a String that names an attribute in the PageContext's attribute map where any resulting object will be stored.

# CallControl Class

*public class* **CallControl**

extends `AbstractCallMethod`

A tag that is used to call a method on a control. The control on which to invoke the method is taken from the `PageContext` attribute map, and the given method is reflectively invoked using the parameters provided in any nested `MethodParameter` tags.

This tag extends the `CallMethod` tag, which defines additional attributes.

The `controlId` attribute is used to specify a `Control` in the `PageContext` attribute map; the method will be invoked on this `Control`. Any return value will be set in the `PageContext` attribute map under the `resultId` attribute that is provided on the `CallMethod` tag. In the case that the return value is a primitive type, the primitive type's object wrapper is set in the `PageContext` under the `resultId` key.

Attribute Descriptions			
Attribute	Required	Runtime Expression Evaluation	Data Bindable
<b>controlId</b>	No	No	Yes
	A <code>PageContext</code> attribute name whose value is the control to call.		
<b>failOnError</b>	No	No	No
	A boolean that if <code>true</code> will cause the tag to report any method invocation errors in-line in the rendered page.		
<b>method</b>	No	No	No
	Name of the method in the control that will be invoked.		
<b>resultId</b>	No	No	No
	The <code>PageContext</code> attribute name where the result, if non-null, is stored.		

## Related Topics

<netui-data:callControl> Tag Sample  
[MethodParameter](#)  
[CallMethod](#)  
[CallPageFlow](#)

---

## Hierarchy

```

Object
TagSupport
BodyTagSupport
AbstractBaseTag
AbstractCallMethod
CallControl

```

### *All Implemented Interfaces*

BodyTag, IterationTag, Serializable, Tag

---

## Field Summary

### Fields from `com.bea.wlw.netui.tags.AbstractBaseTag`

ATTR\_GENERAL, ATTR\_GENERAL\_EXPRESSION, ATTR\_JAVASCRIPT, ATTR\_STYLE, CLASS, ID, JAVASCRIPT\_STATUS, NAME, NETUI\_UNIQUE\_CNT, ONCLICK, ONDBLCLICK, ONKEYDOWN, ONKEYPRESS, ONKEYUP, ONMOUSEDOWN, ONMOUSEMOVE, ONMOUSEOUT, ONMOUSEOVER, ONMOUSEUP, STYLE, TABINDEX

### Fields from `javax.servlet.jsp.tagext.BodyTagSupport`

bodyContent

### Fields from `javax.servlet.jsp.tagext.TagSupport`

id, pageContext

### Fields from interface `javax.servlet.jsp.tagext.BodyTag`

EVAL\_BODY\_BUFFERED, EVAL\_BODY\_TAG

### Fields from interface `javax.servlet.jsp.tagext.IterationTag`

EVAL\_BODY\_AGAIN

### Fields from interface `javax.servlet.jsp.tagext.Tag`

EVAL\_BODY\_INCLUDE, EVAL\_PAGE, SKIP\_BODY, SKIP\_PAGE

## Constructor Summary

*CallControl()*

## Method Summary

```

    public
    String getControlId()
        Get the String identifier of the Control that on which a method will be invoked.

    protected
    String getObjectName()
        Get the name of the object that is the target of the invocation.

    public
    String getTagName()
        Get the name of this tag.

    protected
    void localRelease()
        Reset all of the fields of this tag.

    protected
    Object resolveObject()
        Get the Control whose method to invoke.

    public
    void setControlId(String controlId)
        Set the String which identifies the Control on which a method will be invoked.

```

### Methods from `com.bea.wlw.netui.tags.databinding.invoke.AbstractCallMethod`

```

addParameter, doEndTag, doStartTag, findMethod, getFailOnError,
getMethod, getParameterNodes, getResultId, handleReturnValue,
setFailOnError, setMethod, setResultId

```

### Methods from `com.bea.wlw.netui.tags.AbstractBaseTag`

```

addTagIdMapping, evaluateAttributeToString, filter, filter,
getJavaScriptUtils, getNearestForm, getScriptReporter, getUserLocale,
localRelease, registerTagError, release, renderAttribute, rewriteName,
write

```

### Methods from `javax.servlet.jsp.tagext.BodyTagSupport`

```

doAfterBody, doEndTag, doInitBody, doStartTag, getBodyContent,
getPreviousOut, release, setBodyContent

```

### Methods from `javax.servlet.jsp.tagext.TagSupport`

```

doAfterBody, doEndTag, doStartTag, findAncestorWithClass, getId,
getParent, getValue, getValues, release, removeValue, setId,
setPageContext, setParent, setValue

```

### Methods from class `java.lang.Object`

```

clone, equals, finalize, getClass, hashCode, notify, notifyAll,
toString, wait, wait, wait

```

**Methods from interface `javax.servlet.jsp.tagext.BodyTag`**

`doInitBody`, `setBodyContent`

**Methods from interface `javax.servlet.jsp.tagext.IterationTag`**

`doAfterBody`

**Methods from interface `javax.servlet.jsp.tagext.Tag`**

`doEndTag`, `doStartTag`, `getParent`, `release`, `setPageContext`, `setParent`

## Constructor Detail

### CallControl

```
public CallControl()
```

## Method Detail

### `getControlId()` Method

```
public String getControlId()
```

Get the String identifier of the Control that on which a method will be invoked.

#### Returns

*the String identifier that references the Control*

---

### `getObjectName()` Method

```
protected String getObjectName()
```

Get the name of the object that is the target of the invocation.

#### Overrides

```
AbstractCallMethod.getObjectName()
```

#### Returns

*a name for the object on which the method will be invoked.*

---



## getTagName() Method

```
public String getTagName()
```

Get the name of this tag. This is used to identify the type of this tag for reporting tag errors.

### *Overrides*

```
AbstractBaseTag.getTagName()
```

### Returns

*a constant String representing the name of this tag.*

---

## localRelease() Method

```
protected void localRelease()
```

Reset all of the fields of this tag.

### *Overrides*

```
AbstractCallMethod.localRelease()
```

---

## resolveObject() Method

```
protected Object resolveObject()  
    throws ObjectNotFoundException
```

Get the Control whose method to invoke. This is an implementation of the `CallMethod.resolveObject()` method that looks for a Control in the PageContext attribute map of the current JSP page.

### *Overrides*

```
AbstractCallMethod.resolveObject()
```

### Returns

*the Control on which to invoke the method*

### Exceptions

*ObjectNotFoundException*

---

## setControlId(String) Method

```
public void setControlId(String controlId)
```

Set the String which identifies the Control on which a method will be invoked.

## Parameters

*controlId*

the String identifier that references the Control

# CallMethod Class

*public class* **CallMethod**

extends **AbstractCallMethod**

An abstract base class for tags that are capable of reflectively invoking methods. Specializations of this tag provide method implementations that locate the object on which to invoke the method and that handle any return value from the invoked method.

The **CallMethod** tag can have child tags of type **MethodParameter**; these tags must be in the same order as the parameter list in the method signature of the method that will be invoked. To invoke an overloaded method, the `property` must be set to the **String** name of the type to pass to the method. If the type attribute values on nested **MethodParameter** tags do not match any method signature, an error will be reported in the page.

Attribute Descriptions			
Attribute	Required	Runtime Expression Evaluation	Data Bindable
<b>failOnError</b>	No	No	No
	A boolean that if <code>true</code> will cause the tag to report any method invocation errors in-line in the rendered page.		
<b>method</b>	No	No	No
	The method to invoke on an object.		
<b>object</b>	No	No	Yes
	References an object on which the method will be invoked.		
<b>resultId</b>	No	No	No
	The <b>PageContext</b> attribute name where the result, if non-null, is stored.		

## Related Topics

<netui-databinding:callMethod> Tag Sample

**MethodParameter**

**CallControl**

**CallPageFlow**

## Hierarchy

Object

TagSupport

BodyTagSupport

AbstractBaseTag  
AbstractCallMethod  
CallMethod

### *All Implemented Interfaces*

BodyTag, IterationTag, Serializable, Tag

### *Direct Known Subclasses*

CallPageFlow, ChoiceMethod

---

## Field Summary

### Fields from `com.bea.wlw.netui.tags.AbstractBaseTag`

ATTR\_GENERAL, ATTR\_GENERAL\_EXPRESSION, ATTR\_JAVASCRIPT, ATTR\_STYLE, CLASS, ID, JAVASCRIPT\_STATUS, NAME, NETUI\_UNIQUE\_CNT, ONCLICK, ONDBLCLICK, ONKEYDOWN, ONKEYPRESS, ONKEYUP, ONMOUSEDOWN, ONMOUSEMOVE, ONMOUSEOUT, ONMOUSEOVER, ONMOUSEUP, STYLE, TABINDEX

### Fields from `javax.servlet.jsp.tagext.BodyTagSupport`

bodyContent

### Fields from `javax.servlet.jsp.tagext.TagSupport`

id, pageContext

### Fields from interface `javax.servlet.jsp.tagext.BodyTag`

EVAL\_BODY\_BUFFERED, EVAL\_BODY\_TAG

### Fields from interface `javax.servlet.jsp.tagext.IterationTag`

EVAL\_BODY\_AGAIN

### Fields from interface `javax.servlet.jsp.tagext.Tag`

EVAL\_BODY\_INCLUDE, EVAL\_PAGE, SKIP\_BODY, SKIP\_PAGE

## Constructor Summary

*CallMethod()*

## Method Summary

```
protected
    final findMethod(Object target, String methodName, boolean verifyTypes)
        Method        The default findMethod implementation is an uncached search of all of the methods
                        available on the Class of the target

    public
        String getObject()
                        Get the object on which a method will be invoked.

protected
    String getObjectName()
                        Get the name of the object that is the target of the invocation.

    public
        String getTagName()
                        Get the name of this tag.

protected
    void localRelease()
                        Reset all of the fields of this tag.

protected
    Object resolveObject()
                        Resolve the object on which the method should be invoked.

    public
        void setObject(String object)
                        Set the object on which to invoke a method.
```

### Methods from **com.bea.wlw.netui.tags.databinding.invoke.AbstractCallMethod**

```
addParameter, doEndTag, doStartTag, getFailOnError, getMethod,
getParameterNodes, getResultId, handleReturnValue, setFailOnError,
setMethod, setResultId
```

### Methods from **com.bea.wlw.netui.tags.AbstractBaseTag**

```
addTagIdMapping, evaluateAttributeToString, filter, filter,
getJavaScriptUtils, getNearestForm, getScriptReporter, getUserLocale,
localRelease, registerTagError, release, renderAttribute, rewriteName,
write
```

### Methods from **javax.servlet.jsp.tagext.BodyTagSupport**

```
doAfterBody, doEndTag, doInitBody, doStartTag, getBodyContent,
getPreviousOut, release, setBodyContent
```

### Methods from **javax.servlet.jsp.tagext.TagSupport**

```
doAfterBody, doEndTag, doStartTag, findAncestorWithClass, getId,
getParent, getValue, getValues, release, removeValue, setId,
setPageContext, setParent, setValue
```

### Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

### Methods from interface `javax.servlet.jsp.tagext.BodyTag`

`doInitBody`, `setBodyContent`

### Methods from interface `javax.servlet.jsp.tagext.IterationTag`

`doAfterBody`

### Methods from interface `javax.servlet.jsp.tagext.Tag`

`doEndTag`, `doStartTag`, `getParent`, `release`, `setPageContext`, `setParent`

## Constructor Detail

### CallMethod

```
public CallMethod()
```

## Method Detail

### findMethod(Object, String, boolean) Method

```
protected final Method findMethod(Object target,  
                                   String methodName,  
                                   boolean verifyTypes)
```

*Description copied from* `AbstractCallMethod.findMethod(Object, String, boolean)`  
The default `findMethod` implementation is an uncached search of all of the methods available on the Class of the target

#### Overrides

`AbstractCallMethod.findMethod(Object, String, boolean)`

#### Parameters

*target*

the object from which to find the method

*methodName*

the name of the method to find

*verifyTypes*

a boolean that if true will match the type names in addition to the String method name

## Returns

*a Method object matching the methodName and types, if verifyTypes is true. null otherwise.*

---

## getObject() Method

```
public String getObject()
```

Get the object on which a method will be invoked.

---

## getObjectName() Method

```
protected String getObjectName()
```

*Description copied from AbstractCallMethod.getObjectName()*

Get the name of the object that is the target of the invocation. This is a generic method for this tag that enables more specific error reporting.

### Overrides

```
AbstractCallMethod.getObjectName()
```

## Returns

*a name for the object on which the method will be invoked.*

---

## getTagName() Method

```
public String getTagName()
```

Get the name of this tag. This is used to identify the type of this tag for reporting tag errors.

### Overrides

```
AbstractBaseTag.getTagName()
```

## Returns

*a constant String representing the name of this tag.*

---

## localRelease() Method

```
protected void localRelease()
```

Reset all of the fields of this tag.

**Overrides**

`AbstractCallMethod.localRelease()`

---

**resolveObject() Method**

`protected Object resolveObject()`  
`throws ObjectNotFoundException`

*Description copied from `AbstractCallMethod.resolveObject()`*

Resolve the object on which the method should be invoked. If there are errors resolving this object, this method will throw an `ObjectNotFoundException`. If the object is not found but no exception occurred, this method should return null.

**Overrides**

`AbstractCallMethod.resolveObject()`

**Returns**

*the object on which to reflectively invoke the method.*

**Exceptions**

*`ObjectNotFoundException`*

---

**setObject(String) Method**

`public void setObject(String object)`

Set the object on which to invoke a method.

**Parameters**

*object*  
the object on which to invoke a method



# CallPageFlow Class

*public class CallPageFlow*

extends CallMethod

A tag that is used to call a method on the current PageFlowController. The PageFlow on which to call the method is found given the location of the JSP page in the webapp; if no PageFlow is found in the current directory, an `ObjectNotFoundException` is thrown and the tag execution fails.

This tag extends the `CallMethod` tag, which defines additional attributes.

Once the method has been called, any return value is set in the `PageContext` attribute map under the `resultId` attribute that is provided on the `CallMethod` tag. In the case that the return value is a primitive type, the primitive type's object wrapper is set in the `PageContext` under the `resultId` key.

Attribute Descriptions			
Attribute	Required	Runtime Expression Evaluation	Data Bindable
<b>failOnError</b>	No	No	No
	A boolean that if <code>true</code> will cause the tag to report any method invocation errors in-line in the rendered page.		
<b>method</b>	Yes	No	No
	Name of the method in the page flow that will be invoked.		
<b>resultId</b>	No	No	No
	The <code>PageContext</code> attribute name where the result, if non-null, is stored.		

## Related Topics

<netui-data:callPageFlow> Tag Sample  
 MethodParameter  
 CallMethod  
 CallControl

---

## Hierarchy

```

Object
  TagSupport
    BodyTagSupport
      AbstractBaseTag
        AbstractCallMethod
          CallMethod
  
```

CallPageFlow

**All Implemented Interfaces**BodyTag, IterationTag, Serializable, Tag

---

**Field  
Summary****Fields from com.bea.wlw.netui.tags.AbstractBaseTag**

ATTR\_GENERAL, ATTR\_GENERAL\_EXPRESSION, ATTR\_JAVASCRIPT, ATTR\_STYLE, CLASS, ID, JAVASCRIPT\_STATUS, NAME, NETUI\_UNIQUE\_CNT, ONCLICK, ONDBLCLICK, ONKEYDOWN, ONKEYPRESS, ONKEYUP, ONMOUSEDOWN, ONMOUSEMOVE, ONMOUSEOUT, ONMOUSEOVER, ONMOUSEUP, STYLE, TABINDEX

**Fields from javax.servlet.jsp.tagext.BodyTagSupport**

bodyContent

**Fields from javax.servlet.jsp.tagext.TagSupport**

id, pageContext

**Fields from interface javax.servlet.jsp.tagext.BodyTag**

EVAL\_BODY\_BUFFERED, EVAL\_BODY\_TAG

**Fields from interface javax.servlet.jsp.tagext.IterationTag**

EVAL\_BODY\_AGAIN

**Fields from interface javax.servlet.jsp.tagext.Tag**

EVAL\_BODY\_INCLUDE, EVAL\_PAGE, SKIP\_BODY, SKIP\_PAGE

**Constructor  
Summary***CallPageFlow()***Method  
Summary**

protected

String *getObjectName()*

Get the name of the object that is the target of the invocation.

```

    public getTagName()
        String          Get the name of this tag.
protected
    Object resolveObject()
        Get the PageFlow for the using JSP's directory.

```

**Methods from `com.bea.wlw.netui.tags.databinding.invoke.CallMethod`**

`findMethod, getObject, localRelease, setObject`

**Methods from `com.bea.wlw.netui.tags.databinding.invoke.AbstractCallMethod`**

`addParameter, doEndTag, doStartTag, findMethod, getFailOnError, getMethod, getObject, getName, getParameterNodes, getResultId, handleReturnValue, localRelease, resolveObject, setFailOnError, setMethod, setResultId`

**Methods from `com.bea.wlw.netui.tags.AbstractBaseTag`**

`addTagIdMapping, evaluateAttributeToString, filter, filter, getJavaScriptUtils, getNearestForm, getScriptReporter, getTagName, getUserLocale, localRelease, registerTagError, release, renderAttribute, rewriteName, write`

**Methods from `javax.servlet.jsp.tagext.BodyTagSupport`**

`doAfterBody, doEndTag, doInitBody, doStartTag, getBodyContent, getPreviousOut, release, setBodyContent`

**Methods from `javax.servlet.jsp.tagext.TagSupport`**

`doAfterBody, doEndTag, doStartTag, findAncestorWithClass, getId, getParent, getValue, getValues, release, removeValue, setId, setPageContext, setParent, setValue`

**Methods from class `java.lang.Object`**

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

**Methods from interface `javax.servlet.jsp.tagext.BodyTag`**

`doInitBody, setBodyContent`

**Methods from interface `javax.servlet.jsp.tagext.IterationTag`**

`doAfterBody`

**Methods from interface `javax.servlet.jsp.tagext.Tag`**

`doEndTag, doStartTag, getParent, release, setPageContext, setParent`

## Constructor Detail

### CallPageFlow

```
public CallPageFlow()
```

## Method Detail

### getObjectName() Method

```
protected String getObjectName()
```

Get the name of the object that is the target of the invocation.

#### *Overrides*

```
CallMethod.getObjectName()
```

#### Returns

*a name for the object on which the method will be invoked.*

---

### getTagName() Method

```
public String getTagName()
```

Get the name of this tag. This is used to identify the type of this tag for reporting tag errors.

#### *Overrides*

```
CallMethod.getTagName()
```

#### Returns

*a constant String representing the name of this tag.*

---

### resolveObject() Method

```
protected Object resolveObject()  
    throws ObjectNotFoundException
```

Get the PageFlow for the using JSP's directory. This is an implementation of the `CallMethod.resolveObject()` method that finds the current PageFlow using the `PageFlowUtils.ensureCurrentPageFlow(HttpServletRequest, HttpServletResponse)` method.

### ***Overrides***

`CallMethod.resolveObject()`

### **Returns**

*the current PageFlow. If there is no current PageFlow, the `ObjectNotFoundException` will be thrown.*

### **Exceptions**

*`ObjectNotFoundException`*

# DeclareControl Class

*public class **DeclareControl***

extends `AbstractBaseTag`

Declare a control that is stored in the `PageContext` attribute map. The `declareControl` tag creates an instance of a `Control` that can be invoked by the `CallControl` tag and is valid for the duration of the page.

A control can be referenced from a control project or from the current webapp; if referencing a control in the former location, simply specify the `type` attribute as the class name of the `Control`'s public interface.

Two control types may be referenced by this tag, `.jcx` and `.jcs`. The default control type is `.jcx`, but `.jcs` `Control` instances that automatically create their own control interfaces may also be used.

The declared `Control` instance is stored in the `PageContext` attribute map under the key `controlId`.

Any errors reported by this tag during its lifetime, including those accessing the `Control`'s interface class or ensuring the runtime existence of the control, are reported in the JSP page.

Attribute Descriptions			
Attribute	Required	Runtime Expression Evaluation	Data Bindable
<b>controlId</b>	No	No	No
	The identifier in the <code>PageContext</code> where this control will be stored for use in the page.		
<b>type</b>	No	No	No
	The class name of the control to create.		

## Related Topics

<netui-data:callControl> Tag Sample  
[CallMethod](#)  
[MethodParameter](#)

---

## Hierarchy

```

Object
  TagSupport
    BodyTagSupport
      AbstractBaseTag
        DeclareControl
  
```

## All Implemented Interfaces

## Field Summary

### Fields from `com.bea.wlw.netui.tags.AbstractBaseTag`

ATTR\_GENERAL, ATTR\_GENERAL\_EXPRESSION, ATTR\_JAVASCRIPT, ATTR\_STYLE, CLASS, ID, JAVASCRIPT\_STATUS, NAME, NETUI\_UNIQUE\_CNT, ONCLICK, ONDBLCLICK, ONKEYDOWN, ONKEYPRESS, ONKEYUP, ONMOUSEDOWN, ONMOUSEMOVE, ONMOUSEOUT, ONMOUSEOVER, ONMOUSEUP, STYLE, TABINDEX

### Fields from `javax.servlet.jsp.tagext.BodyTagSupport`

bodyContent

### Fields from `javax.servlet.jsp.tagext.TagSupport`

id, pageContext

### Fields from interface `javax.servlet.jsp.tagext.BodyTag`

EVAL\_BODY\_BUFFERED, EVAL\_BODY\_TAG

### Fields from interface `javax.servlet.jsp.tagext.IterationTag`

EVAL\_BODY\_AGAIN

### Fields from interface `javax.servlet.jsp.tagext.Tag`

EVAL\_BODY\_INCLUDE, EVAL\_PAGE, SKIP\_BODY, SKIP\_PAGE

## Constructor Summary

*DeclareControl()*

## Method Summary

```
public
    int doEndTag()
        Creates an instance of a Control and places it in the PageContext attribute
        map.

public
    int doStartTag()
```

Start the execution of this tag.

```

public
String getControlId()
    Get the identifier where the Control is stored in the PageContext after it is
    created.

public
String getTagName()
    Get the name of this tag.

public
String getType()
    Get the type of Control to create.

protected
void localRelease()
    Reset all of the fields of this tag.

public
void setControlId(String controlId)
    Set the identifier where the Control is stored in the PageContext after it is created.

public
void setType(String type)
    Set the type of Control to create.

```

### Methods from **com.bea.wlw.netui.tags.AbstractBaseTag**

addTagIdMapping, evaluateAttributeToString, filter, filter,  
getJavaScriptUtils, getNearestForm, getScriptReporter, getUserLocale,  
registerTagError, release, renderAttribute, rewriteName, write

### Methods from **javax.servlet.jsp.tagext.BodyTagSupport**

doAfterBody, doInitBody, getBodyContent, getPreviousOut, release,  
setBodyContent

### Methods from **javax.servlet.jsp.tagext.TagSupport**

doAfterBody, doEndTag, doStartTag, findAncestorWithClass, getId,  
getParent, getValue, getValues, release, removeValue, setId,  
setPageContext, setParent, setValue

### Methods from class **java.lang.Object**

clone, equals, finalize, getClass, hashCode, notify, notifyAll,  
toString, wait, wait, wait

### Methods from interface **javax.servlet.jsp.tagext.BodyTag**

doInitBody, setBodyContent

### Methods from interface **javax.servlet.jsp.tagext.IterationTag**

doAfterBody



**Methods from interface `javax.servlet.jsp.tagext.Tag`**`doEndTag, doStartTag, getParent, release, setPageContext, setParent`

## Constructor Detail

### DeclareControl

```
public DeclareControl()
```

## Method Detail

### doEndTag() Method

```
public int doEndTag()  
    throws JspException
```

Creates an instance of a Control and places it in the `PageContext` attribute map. Any errors that occur while creating the Control are reported by the tag to the page. If the Control creation is successful, the control instance is stored in the `PageContext`'s attribute map under the key `controlId`.

**Overrides**

```
BodyTagSupport.doEndTag( )
```

**Returns**

*EVAL\_PAGE*

**Exceptions**

*JspException*  
if an error can not be reported in the page

---

### doStartTag() Method

```
public int doStartTag()
```

Start the execution of this tag. The body of this tag is not rendered.

**Overrides**

```
BodyTagSupport.doStartTag( )
```

**Returns***SKIP\_BODY*

---

**getControlId() Method**

```
public String getControlId()
```

Get the identifier where the Control is stored in the PageContext after it is created.

**Returns***the String identifier*

---

**getTagName() Method**

```
public String getTagName()
```

Get the name of this tag. This is used to identify the type of this tag for reporting tag errors.

**Overrides**

```
AbstractBaseTag.getTagName()
```

**Returns***a constant String representing the name of this tag.*

---

**getType() Method**

```
public String getType()
```

Get the type of Control to create.

**Returns***the type of Control to create*

---

**localRelease() Method**

```
protected void localRelease()
```

Reset all of the fields of this tag.

**Overrides**

```
AbstractBaseTag.localRelease()
```

---

## setControlId(String) Method

```
public void setControlId(String controlId)
```

Set the identifier where the Control is stored in the PageContext after it is created.

### Parameters

*controlId*  
the String identifier

---

## setType(String) Method

```
public void setType(String type)
```

Set the type of Control to create. This should be a String that represents the fully specified class name of the Control to create.

### Parameters

*type*  
the type of Control to create

## MethodParameter Class

*public class **MethodParameter***

extends `AbstractBaseTag`

A tag that is used to add an argument to a method that will be called on some object. This tag can be nested within tags that extend the `AbstractCallMethod` class. Those tags are:

- `<netui-data:callControl>`
- `<netui-data:callMethod>`
- `<netui-data:callPageFlow>`
- `<netui-data:choiceMethod>`

The `methodParameter` tags are used to parameterize the method that the `AbstractCallMethod` class will call; each `methodParameter` tag represents a single parameter. These tags are evaluated in order and the parameters they describe are passed in order.

Overloaded methods on an object can be invoked by setting the `type` attribute on each `methodParameter` tag that is embedded in a method invocation tag. The type name must exactly match the primitive type name or the fully qualified class name of the argument. The `methodParameter` tags must also be in the order that they will be passed to this method. The value of the `type` attribute must be an exact match of the type if it were printed after having been accessed through Java reflection.

In order to pass `null` as an argument to a method, the `null` attribute must be set on this tag. Either the `null` attribute or the `value` attribute must be set on this tag.

Attribute Descriptions			
Attribute	Required	Runtime Expression Evaluation	Data Bindable
<b>null</b>	No	No	No
	A boolean that, if <code>true</code> , will pass null to the method call.		
<b>type</b>	No	No	No
	The String of the type of this argument in the method signature. This name should match the primitive type name or fully qualified class name of the parameters on the signature of the method to which this parameter will be passed. For example:		
	<b>Method Signature</b>	<b>Argument Name</b>	<b>Type value</b>
	<code>addToPrice(int price)</code>	<code>price</code>	<code>int</code>
	<code>addToPrice(Integer price)</code>	<code>price</code>	<code>java.lang.Integer</code>

value	No	No	Yes
	The value to pass to the method call.		

**Related Topics**

[CallControl](#)  
[CallMethod](#)  
[CallPageFlow](#)  
[How Do I: Pass Request-time Data to a methodParameter?](#)

---

***Hierarchy***

```

Object
  TagSupport
    BodyTagSupport
      AbstractBaseTag
        MethodParameter
  
```

***All Implemented Interfaces***

BodyTag, IterationTag, Serializable, Tag

---

## Field

## Summary

**Fields from `com.bea.wlw.netui.tags.AbstractBaseTag`**

ATTR\_GENERAL, ATTR\_GENERAL\_EXPRESSION, ATTR\_JAVASCRIPT, ATTR\_STYLE, CLASS, ID, JAVASCRIPT\_STATUS, NAME, NETUI\_UNIQUE\_CNT, ONCLICK, ONDBLCLICK, ONKEYDOWN, ONKEYPRESS, ONKEYUP, ONMOUSEDOWN, ONMOUSEMOVE, ONMOUSEOUT, ONMOUSEOVER, ONMOUSEUP, STYLE, TABINDEX

**Fields from `javax.servlet.jsp.tagext.BodyTagSupport`**

bodyContent

**Fields from `javax.servlet.jsp.tagext.TagSupport`**

id, pageContext

**Fields from interface `javax.servlet.jsp.tagext.BodyTag`**

EVAL\_BODY\_BUFFERED, EVAL\_BODY\_TAG

**Fields from interface `javax.servlet.jsp.tagext.IterationTag`**

EVAL\_BODY\_AGAIN

**Fields from interface javax.servlet.jsp.tagext.Tag**

EVAL\_BODY\_INCLUDE, EVAL\_PAGE, SKIP\_BODY, SKIP\_PAGE

## Constructor Summary

*MethodParameter()*

## Method Summary

```

public
    int doEndTag()
        Prepare the value to pass up to the AbstractCallMethod type parent.

public
    int doStartTag()
        Start this tag's lifecycle.

public
    String getTagName()
        Get the name of this tag.

public
    String getType()
        Get the name of the type of this argument in the method signature.

public
    String getValue()
        Gets the String value of the parameter to pass to a method or an expression.

public
    boolean isNull()
        Gets a boolean that describes whether or not a null should be passed to the method that
        is to be called.

protected
    void localRelease()
        Reset all of the fields of this tag.

public
    void setNull(boolean isNull)
        Sets a boolean that describes that the parameter that should be passed to the method is
        null.

public
    void setType(String type)
        Set a String matching the type of this parameter on the method to invoke.

public
    void setValue(String value)
        Sets the value of the method parameter that will be passed to the method call.

```

### Methods from `com.bea.wlw.netui.tags.AbstractBaseTag`

`addTagIdMapping`, `evaluateAttributeToString`, `filter`, `filter`,  
`getJavaScriptUtils`, `getNearestForm`, `getScriptReporter`, `getUserLocale`,  
`registerTagError`, `release`, `renderAttribute`, `rewriteName`, `write`

### Methods from `javax.servlet.jsp.tagext.BodyTagSupport`

`doAfterBody`, `doInitBody`, `getBodyContent`, `getPreviousOut`, `release`,  
`setBodyContent`

### Methods from `javax.servlet.jsp.tagext.TagSupport`

`doAfterBody`, `doEndTag`, `doStartTag`, `findAncestorWithClass`, `getId`,  
`getParent`, `getValue`, `getValues`, `release`, `removeValue`, `setId`,  
`setPageContext`, `setParent`, `setValue`

### Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`,  
`toString`, `wait`, `wait`, `wait`

### Methods from interface `javax.servlet.jsp.tagext.BodyTag`

`doInitBody`, `setBodyContent`

### Methods from interface `javax.servlet.jsp.tagext.IterationTag`

`doAfterBody`

### Methods from interface `javax.servlet.jsp.tagext.Tag`

`doEndTag`, `doStartTag`, `getParent`, `release`, `setPageContext`, `setParent`

## Constructor Detail

### MethodParameter

```
public MethodParameter()
```

## Method Detail

### doEndTag() Method

```
public int doEndTag()  
    throws JspException
```

Prepare the value to pass up to the `AbstractCallMethod` type parent.

**Overrides**

`BodyTagSupport.doEndTag()`

**Returns**

*EVAL\_PAGE* to continue evaluating the page

**Exceptions**

*JspException*

---

**doStartTag() Method**

```
public int doStartTag()
    throws JspException
```

Start this tag's lifecycle. Verify that this tag is nested within a `AbstractCallMethod` tag and that one of the "null" and "value" attributes are set.

**Overrides**

`BodyTagSupport.doStartTag()`

**Returns**

*SKIP\_BODY*

**Exceptions**

*JspException*  
if an error occurs getting the parameter

---

**getTagName() Method**

```
public String getTagName()
```

Get the name of this tag. This is used to identify the type of this tag for reporting tag errors.

**Overrides**

`AbstractBaseTag.getTagName()`

**Returns**

*a constant String representing the name of this tag.*

---

**getType() Method**

```
public String getType()
```



Get the name of the type of this argument in the method signature.

### Returns

*the type name*

---

## getValue() Method

```
public String getValue()
```

Gets the String value of the parameter to pass to a method or an expression.

### Returns

*a String value or an expression*

---

## isNull() Method

```
public boolean isNull()
```

Gets a boolean that describes whether or not a null should be passed to the method that is to be called.

### Returns

*true if null will be passed; false otherwise*

---

## localRelease() Method

```
protected void localRelease()
```

Reset all of the fields of this tag.

### Overrides

```
AbstractBaseTag.localRelease()
```

---

## setNull(boolean) Method

```
public void setNull(boolean isNull)
```

Sets a boolean that describes that the parameter that should be passed to the method is null.

### Parameters

*isNull*

a value that describes whether or not this tag should pass null; if `true` null will be passed; otherwise the value from the value attribute will be passed.

## setType(String) Method

```
public void setType(String type)
```

Set a String matching the type of this parameter on the method to invoke.

This name should match the primitive type name or fully qualified class name of the parameters on the signature of the method to which this parameter will be passed.

For example:

Method Signature	Argument Name	Type value
addToPrice(int price)	price	int
addToPrice(Integer price)	price	java.lang.Integer

### Parameters

*type*

the type name

---

## setValue(String) Method

```
public void setValue(String value)
```

Sets the value of the method parameter that will be passed to the method call. This String can be an expression. If the value is not an expression that references an Object, the `AbstractCallMethod.doEndTag()` will attempt to convert the String to type that matches the position of the `MethodParameter` tag in the list of `MethodParameter` tags nested inside of an `AbstractCallMethod` tag.

### Parameters

*value*

a String value which may be an expression

# ObjectNotFoundException Class

*public class ObjectNotFoundException*

extends Exception

An exception thrown when an object on which to invoke a method can not be found. The `CallMethod.resolveObject()` call throws this exception.

---

## **Hierarchy**

```
Object
  Throwable
    Exception
      ObjectNotFoundException
```

## **All Implemented Interfaces**

Serializable

---

## **Constructor Summary**

### ***ObjectNotFoundException()***

Construct an ObjectNotFoundException.

### ***ObjectNotFoundException(String message, Throwable cause, String objectName)***

Construct an ObjectNotFoundException with the given message, cause, and objectName.

### ***ObjectNotFoundException(String message, Throwable cause)***

Construct an ObjectNotFoundException with the given message and cause.

### ***ObjectNotFoundException(String message)***

Construct an ObjectNotFoundException with the given message.

### ***ObjectNotFoundException(Throwable cause)***

Construct an ObjectNotFoundException with the given cause

## Method Summary

```
public
String getObjectName()
    Get the name of the object whose lookup
    failed.
```

### Methods from `java.lang.Throwable`

```
fillInStackTrace, getCause, getLocalizedMessage, getMessage,
getStackTrace, initCause, printStackTrace, printStackTrace,
printStackTrace, setStackTrace, toString
```

### Methods from class `java.lang.Object`

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll,
toString, wait, wait, wait
```

## Constructor Detail

### ObjectNotFoundException

```
public ObjectNotFoundException()

    Construct an ObjectNotFoundException.
```

---

### ObjectNotFoundException

```
public ObjectNotFoundException(String message,
                               Throwable cause,
                               String objectName)

    Construct an ObjectNotFoundException with the given message, cause, and objectName.
```

---

### ObjectNotFoundException

```
public ObjectNotFoundException(String message,
                               Throwable cause)

    Construct an ObjectNotFoundException with the given message and cause.
```

---

### ObjectNotFoundException

```
public ObjectNotFoundException(String message)

    Construct an ObjectNotFoundException with the given message.
```

## ObjectNotFoundException

```
public ObjectNotFoundException(Throwable cause)
```

Construct an ObjectNotFoundException with the given cause

## Method Detail

### getObjectName() Method

```
public String getObjectName()
```

Get the name of the object whose lookup failed.

#### Returns

*the String name*

# **com.bea.wlw.netui.tags.databinding.message Package**

## **Class Summary**

Message	Provides a message schema, which can be parameterized to construct customizable messages.
MessageArg	Provides a parameter value to a message schema.

# Message Class

*public class Message*

extends AbstractBaseTag

Attribute Descriptions			
Attribute	Required	Runtime Expression Evaluation	Data Bindable
<b>resultId</b>	Yes	No	No
	The Id of the resulting String.		
<b>value</b>	Yes	No	Yes
	A pageContext or request data binding expression.		

## Related Topics

<netui-data:message> Tag Sample

MessageArg

How Do I: Customize Message Formats in Page Flows?

## Hierarchy

```

Object
  TagSupport
    BodyTagSupport
      AbstractBaseTag
        Message
  
```

## All Implemented Interfaces

BodyTag, IterationTag, Serializable, Tag

## Field Summary

```

public
static MESSAGE_ARG_KEY
final String String
  
```

## Fields from com.bea.wlw.netui.tags.AbstractBaseTag

ATTR\_GENERAL, ATTR\_GENERAL\_EXPRESSION, ATTR\_JAVASCRIPT, ATTR\_STYLE, CLASS, ID, JAVASCRIPT\_STATUS, NAME, NETUI\_UNIQUE\_CNT, ONCLICK, ONDBLCLICK, ONKEYDOWN, ONKEYPRESS, ONKEYUP, ONMOUSEDOWN, ONMOUSEMOVE,

ONMOUSEOUT, ONMOUSEOVER, ONMOUSEUP, STYLE, TABINDEX

#### Fields from `javax.servlet.jsp.tagext.BodyTagSupport`

`bodyContent`

#### Fields from `javax.servlet.jsp.tagext.TagSupport`

`id`, `pageContext`

#### Fields from interface `javax.servlet.jsp.tagext.BodyTag`

`EVAL_BODY_BUFFERED`, `EVAL_BODY_TAG`

#### Fields from interface `javax.servlet.jsp.tagext.IterationTag`

`EVAL_BODY_AGAIN`

#### Fields from interface `javax.servlet.jsp.tagext.Tag`

`EVAL_BODY_INCLUDE`, `EVAL_PAGE`, `SKIP_BODY`, `SKIP_PAGE`

## Constructor Summary

*Message()*

## Method Summary

```
public
    void addMessageArgument(Object messageArgument)
public
    int doEndTag()
public
    int doStartTag()
public
    String getResultId()
public
    String getTagName()
public
    String getValue()
protected
    void localRelease()
```

This is a method that will initialize all temp state on a tag and should be called in the `doEndTag` method.



```

public
    void setResultId(String resultId)
public
    void setValue(String value)

```

#### Methods from `com.bea.wlw.netui.tags.AbstractBaseTag`

addTagIdMapping, evaluateAttributeToString, filter, filter,  
getJavaScriptUtils, getNearestForm, getScriptReporter, getUserLocale,  
registerTagError, release, renderAttribute, rewriteName, write

#### Methods from `javax.servlet.jsp.tagext.BodyTagSupport`

doAfterBody, doInitBody, getBodyContent, getPreviousOut, release,  
setBodyContent

#### Methods from `javax.servlet.jsp.tagext.TagSupport`

doAfterBody, doEndTag, doStartTag, findAncestorWithClass, getId,  
getParent, getValue, getValues, release, removeValue, setId,  
setPageContext, setParent, setValue

#### Methods from class `java.lang.Object`

clone, equals, finalize, getClass, hashCode, notify, notifyAll,  
toString, wait, wait, wait

#### Methods from interface `javax.servlet.jsp.tagext.BodyTag`

doInitBody, setBodyContent

#### Methods from interface `javax.servlet.jsp.tagext.IterationTag`

doAfterBody

#### Methods from interface `javax.servlet.jsp.tagext.Tag`

doEndTag, doStartTag, getParent, release, setPageContext, setParent

## Field Detail

### MESSAGE\_ARG\_KEY

```
public static final String MESSAGE_ARG_KEY
```

## Constructor Detail

### Message

```
public Message()
```

## Method Detail

### addMessageArgument(Object) Method

```
public void addMessageArgument(Object messageArgument)
```

---

### doEndTag() Method

```
public int doEndTag()  
    throws JspException
```

#### *Overrides*

```
BodyTagSupport.doEndTag( )
```

### Exceptions

*JspException*

---

### doStartTag() Method

```
public int doStartTag()
```

#### *Overrides*

```
BodyTagSupport.doStartTag( )
```

---

### getResultId() Method

```
public String getResultId()
```

---

### getTagName() Method

```
public String getTagName()
```

#### *Overrides*

```
AbstractBaseTag.getTagName( )
```

---

## getValue() Method

```
public String getValue()
```

---

## localRelease() Method

```
protected void localRelease()
```

*Description copied from AbstractBaseTag.localRelease()*

This is a method that will initialize all temp state on a tag and should be called in the doEndTag method.

### *Overrides*

```
AbstractBaseTag.localRelease()
```

---

## setResultId(String) Method

```
public void setResultId(String resultId)
```

---

## setValue(String) Method

```
public void setValue(String value)
```

# MessageArg Class

*public class MessageArg*

extends AbstractBaseTag

Attribute Descriptions			
Attribute	Required	Runtime Expression Evaluation	Data Bindable
value	Yes	No	No
	A String value for the argument.		

### Related Topics

Message  
How Do I: Customize Message Formats in Page Flows?

---

### Hierarchy

Object  
  TagSupport  
    BodyTagSupport  
      AbstractBaseTag  
        MessageArg

### All Implemented Interfaces

BodyTag, IterationTag, Serializable, Tag

---

## Field Summary

### Fields from com.bea.wlw.netui.tags.AbstractBaseTag

ATTR\_GENERAL, ATTR\_GENERAL\_EXPRESSION, ATTR\_JAVASCRIPT, ATTR\_STYLE, CLASS, ID, JAVASCRIPT\_STATUS, NAME, NETUI\_UNIQUE\_CNT, ONCLICK, ONDBLCLICK, ONKEYDOWN, ONKEYPRESS, ONKEYUP, ONMOUSEDOWN, ONMOUSEMOVE, ONMOUSEOUT, ONMOUSEOVER, ONMOUSEUP, STYLE, TABINDEX

### Fields from javax.servlet.jsp.tagext.BodyTagSupport

bodyContent

### Fields from javax.servlet.jsp.tagext.TagSupport

`id, pageContext`

#### Fields from interface `javax.servlet.jsp.tagext.BodyTag`

`EVAL_BODY_BUFFERED, EVAL_BODY_TAG`

#### Fields from interface `javax.servlet.jsp.tagext.IterationTag`

`EVAL_BODY_AGAIN`

#### Fields from interface `javax.servlet.jsp.tagext.Tag`

`EVAL_BODY_INCLUDE, EVAL_PAGE, SKIP_BODY, SKIP_PAGE`

## Constructor Summary

*MessageArg()*

## Method Summary

```

    public
        int doEndTag()
    public
        int doStartTag()
    public
        String getTagName()
    public
        String getValue()
protected
    void localRelease()
        This is a method that will initialize all temp state on a tag and should be called in the
        doEndTag method.
    public
        void setValue(String value)

```

#### Methods from `com.bea.wlw.netui.tags.AbstractBaseTag`

`addTagIdMapping, evaluateAttributeToString, filter, filter,`  
`getJavaScriptUtils, getNearestForm, getScriptReporter, getUserLocale,`  
`registerTagError, release, renderAttribute, rewriteName, write`

#### Methods from `javax.servlet.jsp.tagext.BodyTagSupport`

`doAfterBody, doInitBody, getBodyContent, getPreviousOut, release,`  
`setBodyContent`

#### Methods from `javax.servlet.jsp.tagext.TagSupport`

`doAfterBody`, `doEndTag`, `doStartTag`, `findAncestorWithClass`, `getId`,  
`getParent`, `getValue`, `getValues`, `release`, `removeValue`, `setId`,  
`setPageContext`, `setParent`, `setValue`

#### Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`,  
`toString`, `wait`, `wait`, `wait`

#### Methods from interface `javax.servlet.jsp.tagext.BodyTag`

`doInitBody`, `setBodyContent`

#### Methods from interface `javax.servlet.jsp.tagext.IterationTag`

`doAfterBody`

#### Methods from interface `javax.servlet.jsp.tagext.Tag`

`doEndTag`, `doStartTag`, `getParent`, `release`, `setPageContext`, `setParent`

## Constructor Detail

### MessageArg

```
public MessageArg()
```

## Method Detail

### doEndTag() Method

```
public int doEndTag()  
    throws JspException
```

#### *Overrides*

```
BodyTagSupport.doEndTag( )
```

### Exceptions

*JspException*

---

## doStartTag() Method

```
public int doStartTag()  
    throws JspException
```

### *Overrides*

```
    BodyTagSupport.doStartTag( )
```

### Exceptions

*JspException*

---

## getTagName() Method

```
public String getTagName()
```

### *Overrides*

```
    AbstractBaseTag.getTagName( )
```

---

## getValue() Method

```
public String getValue()
```

---

## localRelease() Method

```
protected void localRelease()
```

*Description copied from AbstractBaseTag.localRelease( )*

This is a method that will initialize all temp state on a tag and should be called in the doEndTag method.

### *Overrides*

```
    AbstractBaseTag.localRelease( )
```

---

## setValue(String) Method

```
public void setValue(String value)
```

# **com.bea.wlw.netui.tags.databinding.pageinput Package**

## **Class Summary**

### **DeclarePageInput**

The DeclarePageInput tag is used to declare variables that are passed to the page from the page flow.



# DeclarePageInput Class

*public class **DeclarePageInput***

extends `AbstractBaseTag`

The `DeclarePageInput` tag is used to declare variables that are passed to the page from the page flow. A page input variable has a lifetime of the page in which it is declared and can be referenced using the `pageInput` data binding context. The `name` attribute here is used as the identifier for the variable inside of this binding context, and the `type` attribute is used to identify the expected type of this variable.

Attribute Descriptions			
Attribute	Required	Runtime Expression Evaluation	Data Bindable
name	Yes	No	No
	The name of a variable that can be referenced in the page input data binding context.		
type	Yes	No	No
	The type of the variable referenced in the page input data binding context.		

## Related Topics

<netui-data:declarePageInput> Tag Sample

---

## Hierarchy

```

Object
  TagSupport
    BodyTagSupport
      AbstractBaseTag
        DeclarePageInput
  
```

## All Implemented Interfaces

`BodyTag`, `IterationTag`, `Serializable`, `Tag`

---

## Field Summary

Fields from `com.bea.wlw.netui.tags.AbstractBaseTag`

ATTR\_GENERAL, ATTR\_GENERAL\_EXPRESSION, ATTR\_JAVASCRIPT, ATTR\_STYLE, CLASS, ID, JAVASCRIPT\_STATUS, NAME, NETUI\_UNIQUE\_CNT, ONCLICK, ONDBLCLICK, ONKEYDOWN, ONKEYPRESS, ONKEYUP, ONMOUSEDOWN, ONMOUSEMOVE, ONMOUSEOUT, ONMOUSEOVER, ONMOUSEUP, STYLE, TABINDEX

### Fields from `javax.servlet.jsp.tagext.BodyTagSupport`

bodyContent

### Fields from `javax.servlet.jsp.tagext.TagSupport`

id, pageContext

### Fields from interface `javax.servlet.jsp.tagext.BodyTag`

EVAL\_BODY\_BUFFERED, EVAL\_BODY\_TAG

### Fields from interface `javax.servlet.jsp.tagext.IterationTag`

EVAL\_BODY\_AGAIN

### Fields from interface `javax.servlet.jsp.tagext.Tag`

EVAL\_BODY\_INCLUDE, EVAL\_PAGE, SKIP\_BODY, SKIP\_PAGE

## Constructor Summary

*DeclarePageInput()*

## Method Summary

```
public
    int doEndTag()
public
    int doStartTag()
public
    String getName()
        Get the name of the variable that can be referenced using the page input data binding
        context.
public
    String getTagName()
        Get the name of this tag.
public
    String getType()
        Get the type of the variable that referenced with the attribute on this tag.
```

```
protected localRelease()
    void          Reset all of the fields of this tag.
public
    void setName(String name)
        Set the name of a variable that can be referenced using the page input data binding
        context.
public
    void setType(String type)
        Set the type of the variable that referenced with the attribute on this tag.
```

#### Methods from `com.bea.wlw.netui.tags.AbstractBaseTag`

```
addTagIdMapping, evaluateAttributeToString, filter, filter,
getJavaScriptUtils, getNearestForm, getScriptReporter, getUserLocale,
registerTagError, release, renderAttribute, rewriteName, write
```

#### Methods from `javax.servlet.jsp.tagext.BodyTagSupport`

```
doAfterBody, doInitBody, getBodyContent, getPreviousOut, release,
setBodyContent
```

#### Methods from `javax.servlet.jsp.tagext.TagSupport`

```
doAfterBody, doEndTag, doStartTag, findAncestorWithClass, getId,
getParent, getValue, getValues, release, removeValue, setId,
setPageContext, setParent, setValue
```

#### Methods from class `java.lang.Object`

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll,
toString, wait, wait, wait
```

#### Methods from interface `javax.servlet.jsp.tagext.BodyTag`

```
doInitBody, setBodyContent
```

#### Methods from interface `javax.servlet.jsp.tagext.IterationTag`

```
doAfterBody
```

#### Methods from interface `javax.servlet.jsp.tagext.Tag`

```
doEndTag, doStartTag, getParent, release, setPageContext, setParent
```

## Constructor Detail

### DeclarePageInput

```
public DeclarePageInput()
```

## Method Detail

### doEndTag() Method

```
public int doEndTag()  
    throws JspException
```

#### *Overrides*

```
BodyTagSupport.doEndTag( )
```

#### Returns

*EVAL\_PAGE* as this tag always continues to evaluate the page

#### Exceptions

*JspException*

---

### doStartTag() Method

```
public int doStartTag()
```

#### *Overrides*

```
BodyTagSupport.doStartTag( )
```

#### Returns

*SKIP\_BODY* as this tag does not evaluate its body

---

### getName() Method

```
public String getName()
```

Get the name of the variable that can be referenced using the page input data binding context.

---

### getTagName() Method

```
public String getTagName()
```

Get the name of this tag. This is used to identify the type of this tag for reporting tag errors.

#### *Overrides*

```
AbstractBaseTag.getTagName( )
```

## Returns

*a constant String representing the name of this tag.*

---

## getType() Method

```
public String getType()
```

Get the type of the variable that referenced with the `type` attribute on this tag.

---

## localRelease() Method

```
protected void localRelease()
```

Reset all of the fields of this tag.

### Overrides

```
AbstractBaseTag.localRelease()
```

---

## setName(String) Method

```
public void setName(String name)
```

Set the name of a variable that can be referenced using the page input data binding context.

### Parameters

*name*

the name of the variable

---

## setType(String) Method

```
public void setType(String type)
```

Set the type of the variable that referenced with the `type` attribute on this tag.

### Parameters

*type*

the type of the variable that is referenced

# com.bea.wlw.netui.tags.databinding.repeater Package

## Class Summary

Repeater	The <netui-data:repeater> tag is a markup-generic tag that repeats over a data set.
RepeaterComponent	The base class for tags that are part of the Repeater tag set and participate in the structured nature of Repeater rendering.
RepeaterFooter	This tag is used to render the footer of a Repeater.
RepeaterHeader	This tag is used to render the header of a Repeater.
RepeaterItem	This tag is used to render each item in the data set.

## Repeater Class

*public class **Repeater***

extends DataSourceTag

implements DataAccessProvider

The <netui-data:repeater> tag is a markup-generic tag that repeats over a data set. The repeater tag set is used to render data from a data set into a page. The repeater itself does not render any markup. Instead, the markup from its contained tags is rendered to create the content generated by this tag. The tags in the repeater tag set are as follows:

Tag	Description
RepeaterHeader	Renders once in the Repeater . HEADER state.
RepeaterItem	Renders once in the Repeater . ITEM state.
RepeaterFooter	Renders once in the Repeater . FOOTER state.
Pad	Used to convert irregular data sets into regular data sets through padding or truncating the output.
Choice	Used to render different UI for a particular data item bsaed on the result of calling a "decision" method with the ChoiceMethod tag. This is allowed only within the RepeaterItem tag.
ChoiceMethod	Used to define the choices when rendering an item with a RepeaterItem tag. This is allowed only within the RepeaterItem tag.

The repeater can render in two modes; the first mode is a simple mode where the body of the repeater is rendered once for each item in the data set. In this case, none of the other tags above are present in the repeater body. For example, the following will render an unordered HTML list of items that are list items which contain the lastName, firstName of the current "customer" in the data set.

```
<ul>
<netui-data:repeater dataSource="{pageFlow.customers}">
  <li><netui:label value="{container.item.lastName}, {container.item.firstName}"/></li>
</netui-data:repeater>
</ul>
```

The second mode is a more structured mode of rendering where the tags above are used to delineate iteration boundaries on the body of a <netui-data:repeater> tag. In this case, if one of the above tags is present, any content directly in the body of the repeater is not rendered; rather, the content inside the structured tags of the repeater is rendered.

## Page Flow API Reference

For example, the following will render the same output as the example shown above, but it uses the structured tags for rendering the `{pageFlow.customers}` expression:

```
<netui-data:repeater dataSource="{pageFlow.customers}">
  <netui-data:repeaterHeader>
    <ul>
  </netui-data:repeaterHeader>
  <netui-data:repeaterItem>
    <li><netui:label value="{container.item.lastName}, {container.item.firstName}"/></li>
  </netui-data:repeaterItem>
  <netui-data:repeaterFooter>
    </ul>
  </netui-data:repeaterFooter>
</netui-data:repeater>
```

Attribute Descriptions			
Attribute	Required	Runtime Expression Evaluation	Data Bindable
defaultText	No	No	No
	The text that is rendered if the data set is null or contains zero items.		
ignoreNulls	No	No	No
	A boolean to set if rendering a sparse data set that can contain nulls which should be ignored.		

### Related Topics

<netui-data:repeater> Tag Sample  
Presenting Complex Data Sets in JSPs (Repeater Tags section)  
RepeaterHeader  
RepeaterItem  
RepeaterFooter

---

### Hierarchy

- Object
  - TagSupport
    - BodyTagSupport
      - AbstractBaseTag
        - DataSourceTag
          - Repeater

### All Implemented Interfaces

BodyTag, DataAccessProvider, IterationTag, Serializable, Tag

---



## Field Summary

```

    public
    static END
    final int
        A Repeater rendering state that signals the end of repeater rendering.

    public
    static FOOTER
    final int
        A Repeater rendering state that signals the rendering of the FOOTER.

    public
    static HEADER
    final int
        A Repeater rendering state that signals the rendering of the HEADER.

    public
    static INIT
    final int
        A Repeater rendering state that signals the beginning of repeater rendering.

    public
    static ITEM
    final int
        A Repeater rendering state that signals the rendering of the ITEM.

```

### Fields from `com.bea.wlw.netui.tags.AbstractBaseTag`

ATTR\_GENERAL, ATTR\_GENERAL\_EXPRESSION, ATTR\_JAVASCRIPT, ATTR\_STYLE, CLASS, ID, JAVASCRIPT\_STATUS, NAME, NETUI\_UNIQUE\_CNT, ONCLICK, ONDBLCLICK, ONKEYDOWN, ONKEYPRESS, ONKEYUP, ONMOUSEDOWN, ONMOUSEMOVE, ONMOUSEOUT, ONMOUSEOVER, ONMOUSEUP, STYLE, TABINDEX

### Fields from `javax.servlet.jsp.tagext.BodyTagSupport`

bodyContent

### Fields from `javax.servlet.jsp.tagext.TagSupport`

id, pageContext

### Fields from interface `javax.servlet.jsp.tagext.BodyTag`

EVAL\_BODY\_BUFFERED, EVAL\_BODY\_TAG

### Fields from interface `javax.servlet.jsp.tagext.IterationTag`

EVAL\_BODY\_AGAIN

### Fields from interface `javax.servlet.jsp.tagext.Tag`

EVAL\_BODY\_INCLUDE, EVAL\_PAGE, SKIP\_BODY, SKIP\_PAGE

## Constructor

### Summary

*Repeater()*

## Method

### Summary

```

public void
    addContent(String content)
        Add content to the content that is being buffered by this tag.

public int
    doAfterBody()

public int
    doEndTag()

public int
    doStartTag()

public int
    getCurrentIndex()
        Get the index of the current iteration through the body of this tag.

public Object
    getCurrentItem()
        Get the item that is currently being rendered by this repeating tag.

public Object
    getCurrentMetadata()
        Get the metadata for the current item.

public String
    getDefaultText()
        Get the defaultText attribute.

public boolean
    getIgnoreNulls()
        Get a boolean that describes whether the repeater should ignore null items
        encountered while iterating over a data set.

public
DataAccessProvider getProviderParent()
        Get the parent DataAccessProvider for this tag.

public int
    getRenderState()
        Get the current render state for the repeater.

public String
    getTagName()
        Get the name of this tag.

protected void
    localRelease()
        Reset all of the fields of this tag.

```

## Page Flow API Reference

```
public void registerChildTag(RepeaterComponent repeaterComponent)
public void
    setDefaultText(String defaultText)
    Set the text that will be rendered if the dataSource expression references a
    null object and the defaultText attribute is non-null.

public void
    setIgnoreNulls(boolean ignoreNulls)
    Set a boolean that describes whether the repeater should ignore null items
    encountered while iterating over a data set.

public void
    setPadContext(PadContext padContext)
```

### Methods from **com.bea.wlw.netui.tags.DataSourceTag**

getDataSource, setDataSource

### Methods from **com.bea.wlw.netui.tags.AbstractBaseTag**

addTagIdMapping, evaluateAttributeToString, filter, filter,  
getJavaScriptUtils, getNearestForm, getScriptReporter, getUserLocale,  
localRelease, registerTagError, release, renderAttribute, rewriteName,  
write

### Methods from **javax.servlet.jsp.tagext.BodyTagSupport**

doInitBody, getBodyContent, getPreviousOut, release, setBodyContent

### Methods from **javax.servlet.jsp.tagext.TagSupport**

doAfterBody, doEndTag, doStartTag, findAncestorWithClass, getId,  
getParent, getValue, getValues, release, removeValue, setId,  
setPageContext, setParent, setValue

### Methods from class **java.lang.Object**

clone, equals, finalize, getClass, hashCode, notify, notifyAll,  
toString, wait, wait, wait

### Methods from interface **javax.servlet.jsp.tagext.BodyTag**

doInitBody, setBodyContent

### Methods from interface **com.bea.wlw.netui.tags.DataAccessProvider**

getCurrentIndex, getCurrentItem, getCurrentMetadata, getDataSource,  
getProviderParent, setDataSource

### Methods from interface **javax.servlet.jsp.tagext.IterationTag**

doAfterBody

Methods from interface `javax.servlet.jsp.tagext.Tag`

`doEndTag`, `doStartTag`, `getParent`, `release`, `setPageContext`, `setParent`

## Field Detail

### END

```
public static final int END
```

A Repeater rendering state that signals the end of repeater rendering.

---

### FOOTER

```
public static final int FOOTER
```

A Repeater rendering state that signals the rendering of the FOOTER. The body renders in the FOOTER state once.

---

### HEADER

```
public static final int HEADER
```

A Repeater rendering state that signals the rendering of the HEADER. The body renders in the HEADER state once.

---

### INIT

```
public static final int INIT
```

A Repeater rendering state that signals the beginning of repeater rendering.

---

### ITEM

```
public static final int ITEM
```

A Repeater rendering state that signals the rendering of the ITEM. The body renders in the ITEM state once for each item in the data set.

## Constructor Detail

### Repeater

```
public Repeater()
```

## Method Detail

### addContent(String) Method

```
public void addContent(String content)
```

Add content to the content that is being buffered by this tag. All content written by the body of this tag is added to this buffer. The buffer is rendered at the end of the tag's lifecycle if no fatal errors have occurred during this tag's lifecycle.

#### Parameters

*content*

content that this tag should render.

---

### doAfterBody() Method

```
public int doAfterBody()
```

#### Overrides

BodyTagSupport.doAfterBody()

#### Returns

*SKIP\_BODY* if an error occurs or the data set has been rendered; *EVAL\_BODY\_AGAIN* otherwise

---

### doEndTag() Method

```
public int doEndTag()
    throws JspException
```

#### Overrides

BodyTagSupport.doEndTag()

#### Returns

*EVAL\_PAGE*

**Exceptions***JspException*if an error occurs that can not be reported in the page

---

**doStartTag() Method**

```
public int doStartTag()
    throws JspException
```

**Overrides**

BodyTagSupport.doStartTag()

**Returns***SKIP\_BODY* if an error occurs; *EVAL\_BODY\_BUFFERED* otherwise**Exceptions***JspException*if an error occurs that can not be reported in the page

---

**getCurrentIndex() Method**

```
public int getCurrentIndex()
```

Get the index of the current iteration through the body of this tag. This data can be accessed using the expression `container.index` on an attribute of a databindable NetUI tag that is contained within the repeating body of this tag. This expression is only valid when the dataset is being rendered.

**Returns***the integer index of the current data item in the data set***Related Topics**DataAccessProvider

---

**getCurrentItem() Method**

```
public Object getCurrentItem()
```

Get the item that is currently being rendered by this repeating tag. This can be accessed using the expression `expression.item` on an attribute of a databindable NetUI tag that is contained within the repeating body of this tag. The expression is only valid when the dataset is being rendered.

## Returns

*the current item in the data set*

## Related Topics

`DataProvider`

---

## `getCurrentMetadata()` Method

```
public Object getCurrentMetadata()
```

Get the metadata for the current item. This method is not supported by this tag.

## Returns

*the current metadata or null if no metadata can be found or metadata is not supported by a DataProvider implementation*

## Related Topics

`DataProvider`

---

## `getDefaultText()` Method

```
public String getDefaultText()
```

Get the defaultText attribute.

## Returns

*the default text or null if unset.*

---

## `getIgnoreNulls()` Method

```
public boolean getIgnoreNulls()
```

Get a boolean that describes whether the repeater should ignore null items encountered while iterating over a data set.

## Returns

*true if ignoring nulls; false otherwise*

---

## getProviderParent() Method

```
public DataAccessProvider getProviderParent()
```

Get the parent `DataAccessProvider` for this tag. If this tag is contained within a `DataAccessProvider`, the containing `DataAccessProvider` is available through the expression `container.container`. Any valid properties of the parent `DataAccessProvider` can be accessed through this expression. This method will return null if there is no parent `DataAccessProvider`.

### Returns

*a containing `DataAccessProvider` if one exists, null otherwise.*

### Related Topics

`DataAccessProvider`

---

## getRenderState() Method

```
public int getRenderState()
```

Get the current render state for the repeater. This tag is used by child tags to access the current location in the repeater's rendering lifecycle.

### Returns

*an integer that represents the current state of the grid; this is one of `Repeater.INIT`, `Repeater.HEADER`, `Repeater.ITEM`, `Repeater.FOOTER`, or `Repeater.END`.*

---

## getTagName() Method

```
public String getTagName()
```

Get the name of this tag. This is used to identify the type of this tag for reporting tag errors.

### Overrides

```
AbstractBaseTag.getTagName()
```

### Returns

*a constant `String` representing the name of this tag.*

---

## localRelease() Method

```
protected void localRelease()
```

Reset all of the fields of this tag.



**Overrides**

```
DataSourceTag.localRelease()
```

---

**registerChildTag(RepeaterComponent) Method**

```
public void registerChildTag(RepeaterComponent repeaterComponent)
```

**Parameters**

*repeaterComponent*

RepeaterComponent to register with the Repeater parent

---

**setDefaultText(String) Method**

```
public void setDefaultText(String defaultText)
```

Set the text that will be rendered if the dataSource expression references a null object and the defaultText attribute is non-null.

**Parameters**

*defaultText*

the default text

---

**setIgnoreNulls(boolean) Method**

```
public void setIgnoreNulls(boolean ignoreNulls)
```

Set a boolean that describes whether the repeater should ignore null items encountered while iterating over a data set.

**Parameters**

*ignoreNulls*

whether or not to ignore nulls

---

**setPadContext(PadContext) Method**

```
public void setPadContext(PadContext padContext)
```

# RepeaterComponent Class

*public abstract class **RepeaterComponent***

extends StructuredBaseTag

The base class for tags that are part of the Repeater tag set and participate in the structured nature of Repeater rendering. This class provides typed access to the Repeater tag and enforces the basic JSP tag parenting requirements of tags that can only be nested within the Repeater tag.

## Related Topics

StructuredBaseTag

---

## Hierarchy

```

Object
  TagSupport
    BodyTagSupport
      AbstractBaseTag
        StructuredBaseTag
          RepeaterComponent
  
```

## All Implemented Interfaces

BodyTag, IterationTag, Serializable, Tag

## Direct Known Subclasses

Choice, Pad, RepeaterFooter, RepeaterHeader, RepeaterItem

---

## Field Summary

```

protected
  static debug
final Debug          Debug
  
```

### Fields from com.bea.wlw.netui.tags.databinding.base.StructuredBaseTag

EMPTY\_STRING

### Fields from com.bea.wlw.netui.tags.AbstractBaseTag

ATTR\_GENERAL, ATTR\_GENERAL\_EXPRESSION, ATTR\_JAVASCRIPT, ATTR\_STYLE, CLASS, ID, JAVASCRIPT\_STATUS, NAME, NETUI\_UNIQUE\_CNT, ONCLICK, ONDBLCLICK, ONKEYDOWN, ONKEYPRESS, ONKEYUP, ONMOUSEDOWN, ONMOUSEMOVE,

ONMOUSEOUT, ONMOUSEOVER, ONMOUSEUP, STYLE, TABINDEX

#### Fields from `javax.servlet.jsp.tagext.BodyTagSupport`

`bodyContent`

#### Fields from `javax.servlet.jsp.tagext.TagSupport`

`id`, `pageContext`

#### Fields from interface `javax.servlet.jsp.tagext.BodyTag`

`EVAL_BODY_BUFFERED`, `EVAL_BODY_TAG`

#### Fields from interface `javax.servlet.jsp.tagext.IterationTag`

`EVAL_BODY_AGAIN`

#### Fields from interface `javax.servlet.jsp.tagext.Tag`

`EVAL_BODY_INCLUDE`, `EVAL_PAGE`, `SKIP_BODY`, `SKIP_PAGE`

## Constructor Summary

***RepeaterComponent()***

## Method Summary

```
public
    int doAfterBody()
        Default implementation of this JSP lifecycle method.

protected
    int getCurrentRenderState()
        Get the render state of the nearest structured tag.

protected
    final getRepeater()
        Repeater    Get the Repeater parent of this tag.

protected
    Class getValidContainerType()
        Default implementation of StructuredBaseTag.getValidContainerType() that enforces
        that the parent of this tag is the Repeater tag.

protected
    void localRelease()
        Reset all of the fields of this tag.
```

```
protected prepare()
    void          Prepare data that is provided to subclasses of this tag.
protected
    int renderEndTag(int state)
        Default implementation of StructuredBaseTag.renderEndTag(int).
```

#### Methods from **com.bea.wlw.netui.tags.databinding.base.StructuredBaseTag**

```
addContent, doEndTag, doStartTag, getContent, renderStartTag,
verifyAttributes, verifyStructure
```

#### Methods from **com.bea.wlw.netui.tags.AbstractBaseTag**

```
addTagIdMapping, evaluateAttributeToString, filter, filter,
getJavaScriptUtils, getNearestForm, getScriptReporter, getTagName,
getUserLocale, localRelease, registerTagError, release, renderAttribute,
rewriteName, write
```

#### Methods from **javax.servlet.jsp.tagext.BodyTagSupport**

```
doEndTag, doInitBody, doStartTag, getBodyContent, getPreviousOut,
release, setBodyContent
```

#### Methods from **javax.servlet.jsp.tagext.TagSupport**

```
doAfterBody, doEndTag, doStartTag, findAncestorWithClass, getId,
getParent, getValue, getValues, release, removeValue, setId,
setPageContext, setParent, setValue
```

#### Methods from class **java.lang.Object**

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll,
toString, wait, wait, wait
```

#### Methods from interface **javax.servlet.jsp.tagext.BodyTag**

```
doInitBody, setBodyContent
```

#### Methods from interface **javax.servlet.jsp.tagext.IterationTag**

```
doAfterBody
```

#### Methods from interface **javax.servlet.jsp.tagext.Tag**

```
doEndTag, doStartTag, getParent, release, setPageContext, setParent
```

## Field Detail

**debug**

```
protected static final Debug debug
```

**Constructor Detail****RepeaterComponent**

```
public RepeaterComponent()
```

**Method Detail****doAfterBody() Method**

```
public int doAfterBody()
    throws JspException
```

Default implementation of this JSP lifecycle method.

**Overrides**

```
BodyTagSupport.doAfterBody()
```

**Returns**

*SKIP\_BODY*

**Exceptions**

*JspException*

---

**getCurrentRenderState() Method**

```
protected int getCurrentRenderState()
```

Get the render state of the nearest structured tag. For subclasses of this tag, the nearest structured tag is the Repeater tag.

**Overrides**

```
StructuredBaseTag.getCurrentRenderState()
```

**Returns**

*the current render state of the Repeater*

---

## getRepeater() Method

```
protected final Repeater getRepeater()
```

Get the `Repeater` parent of this tag.

### Returns

*the Repeater parent of this tag*

---

## getValidContainerType() Method

```
protected Class getValidContainerType()
```

Default implementation of `StructuredBaseTag.getValidContainerType()` that enforces that the parent of this tag is the `Repeater` tag. Subclasses may override this method to further specialize the required container type.

### Overrides

`StructuredBaseTag.getValidContainerType()`

### Returns

*the valid container type of a RepeaterComponent; the Repeater.class*

---

## localRelease() Method

```
protected void localRelease()
```

Reset all of the fields of this tag.

### Overrides

`StructuredBaseTag.localRelease()`

---

## prepare() Method

```
protected void prepare()
```

Prepare data that is provided to subclasses of this tag.

### Overrides

`StructuredBaseTag.prepare()`

---

## renderEndTag(int) Method

```
protected int renderEndTag(int state)  
    throws JspException
```

Default implementation of StructuredBaseTag.renderEndTag(int). This returns EVAL\_PAGE.

### *Overrides*

StructuredBaseTag.renderEndTag(int)

### Parameters

*state*

the current render state of the Repeater

### Returns

*EVAL\_PAGE*

### Exceptions

*JspException*

if an error occurs that can not be reported in the page

# RepeaterFooter Class

*public class* **RepeaterFooter**

extends RepeaterComponent

This tag is used to render the footer of a Repeater. This header may contain any markup or tags that can be rendered to a page. In the case of closing a table, ordered list, or unordered list, the elements <table>, <ol>, and <ul> could respectively be rendered inside the body of this tag. There is no data item present at the time that the RepeaterFooter renders, so tags in the body can not reference the container databinding context to access a current item or index, though other databinding contexts are available.

The RepeaterFooter renders exactly once in a table during the Repeater tag's Repeater.FOOTER rendering state.

## Related Topics

<netui-data:repeater> Tag Sample  
Presenting Complex Data Sets in JSPs (Repeater Tags section)  
Repeater  
RepeaterHeader  
RepeaterItem

---

## Hierarchy

```
Object
  TagSupport
    BodyTagSupport
      AbstractBaseTag
        StructuredBaseTag
          RepeaterComponent
            RepeaterFooter
```

## All Implemented Interfaces

BodyTag, IterationTag, Serializable, Tag

---

## Field Summary

Fields from com.bea.wlw.netui.tags.databinding.repeater.RepeaterComponent

debug

Fields from com.bea.wlw.netui.tags.databinding.base.StructuredBaseTag



EMPTY\_STRING

#### Fields from `com.bea.wlw.netui.tags.AbstractBaseTag`

ATTR\_GENERAL, ATTR\_GENERAL\_EXPRESSION, ATTR\_JAVASCRIPT, ATTR\_STYLE, CLASS, ID, JAVASCRIPT\_STATUS, NAME, NETUI\_UNIQUE\_CNT, ONCLICK, ONDBLCLICK, ONKEYDOWN, ONKEYPRESS, ONKEYUP, ONMOUSEDOWN, ONMOUSEMOVE, ONMOUSEOUT, ONMOUSEOVER, ONMOUSEUP, STYLE, TABINDEX

#### Fields from `javax.servlet.jsp.tagext.BodyTagSupport`

bodyContent

#### Fields from `javax.servlet.jsp.tagext.TagSupport`

id, pageContext

#### Fields from interface `javax.servlet.jsp.tagext.BodyTag`

EVAL\_BODY\_BUFFERED, EVAL\_BODY\_TAG

#### Fields from interface `javax.servlet.jsp.tagext.IterationTag`

EVAL\_BODY\_AGAIN

#### Fields from interface `javax.servlet.jsp.tagext.Tag`

EVAL\_BODY\_INCLUDE, EVAL\_PAGE, SKIP\_BODY, SKIP\_PAGE

## Constructor Summary

*RepeaterFooter()*

## Method Summary

```
public
    int doAfterBody()
        Render the content of the body to the Repeater
        tag.

public
    String getTagName()
        Get the name of this tag.

protected
    void localRelease()
        Reset all of the fields of this tag.
```

```
public renderStartTag(int state)
    int          Render the start of this tag.
```

**Methods from com.bea.wlw.netui.tags.databinding.repeater.RepeaterComponent**

```
getCurrentRenderState, getRepeater, getValidContainerType, prepare,
renderEndTag
```

**Methods from com.bea.wlw.netui.tags.databinding.base.StructuredBaseTag**

```
addContent, doEndTag, doStartTag, getContent, getCurrentRenderState,
getValidContainerType, localRelease, prepare, renderEndTag,
verifyAttributes, verifyStructure
```

**Methods from com.bea.wlw.netui.tags.AbstractBaseTag**

```
addTagIdMapping, evaluateAttributeToString, filter, filter,
getJavaScriptUtils, getNearestForm, getScriptReporter, getUserLocale,
localRelease, registerTagError, release, renderAttribute, rewriteName,
write
```

**Methods from javax.servlet.jsp.tagext.BodyTagSupport**

```
doAfterBody, doEndTag, doInitBody, doStartTag, getBodyContent,
getPreviousOut, release, setBodyContent
```

**Methods from javax.servlet.jsp.tagext.TagSupport**

```
doAfterBody, doEndTag, doStartTag, findAncestorWithClass, getId,
getParent, getValue, getValues, release, removeValue, setId,
setPageContext, setParent, setValue
```

**Methods from class java.lang.Object**

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll,
toString, wait, wait, wait
```

**Methods from interface javax.servlet.jsp.tagext.BodyTag**

```
doInitBody, setBodyContent
```

**Methods from interface javax.servlet.jsp.tagext.IterationTag**

```
doAfterBody
```

**Methods from interface javax.servlet.jsp.tagext.Tag**

```
doEndTag, doStartTag, getParent, release, setPageContext, setParent
```

## Constructor Detail

### RepeaterFooter

```
public RepeaterFooter()
```

## Method Detail

### doAfterBody() Method

```
public int doAfterBody()
```

Render the content of the body to the Repeater tag.

#### *Overrides*

```
RepeaterComponent.doAfterBody()
```

#### Returns

*SKIP\_BODY*

---

### getTagName() Method

```
public String getTagName()
```

Get the name of this tag. This is used to identify the type of this tag for reporting tag errors.

#### *Overrides*

```
AbstractBaseTag.getTagName()
```

#### Returns

*a constant String representing the name of this tag.*

---

### localRelease() Method

```
protected void localRelease()
```

*Description copied from RepeaterComponent.localRelease()*

Reset all of the fields of this tag.

#### *Overrides*

```
RepeaterComponent.localRelease()
```

---

## renderStartTag(int) Method

```
public int renderStartTag(int state)
```

Render the start of this tag. If the repeater render state is `Repeater.FOOTER`, the body is evaluated. Otherwise, the body is skipped.

### *Overrides*

```
StructuredBaseTag.renderStartTag(int)
```

### **Returns**

*SKIP\_BODY unless the repeater render state is `Repeater.FOOTER`, then `EVAL_BODY_BUFFERED`*

# RepeaterHeader Class

*public class* **RepeaterHeader**

extends RepeaterComponent

This tag is used to render the header of a `Repeater`. This header may contain any markup or tags that can be rendered to a page. In the case of starting a table, ordered list, or unordered list, the HTML elements `<table>`, `<ol>`, and `<ul>` could respectively be rendered inside the body of this tag. There is no data item present at the time that the `RepeaterHeader` renders, so tags in the body can not reference the `container` databinding context to access a current item or index, though other databinding contexts are available.

The `RepeaterHeader` renders exactly once in a table during the `Repeater` tag's `Repeater.HEADER` rendering state.

## Related Topics

`<netui-data:repeater>` Tag Sample  
 Presenting Complex Data Sets in JSPs (Repeater Tags section)  
[Repeater](#)  
[RepeaterItem](#)  
[RepeaterFooter](#)

---

## Hierarchy

```

Object
  TagSupport
    BodyTagSupport
      AbstractBaseTag
        StructuredBaseTag
          RepeaterComponent
            RepeaterHeader
  
```

## All Implemented Interfaces

BodyTag, IterationTag, Serializable, Tag

---

## Field Summary

Fields from `com.bea.wlw.netui.tags.databinding.repeater.RepeaterComponent`

`debug`

Fields from `com.bea.wlw.netui.tags.databinding.base.StructuredBaseTag`

EMPTY\_STRING

#### Fields from `com.bea.wlw.netui.tags.AbstractBaseTag`

ATTR\_GENERAL, ATTR\_GENERAL\_EXPRESSION, ATTR\_JAVASCRIPT, ATTR\_STYLE, CLASS, ID, JAVASCRIPT\_STATUS, NAME, NETUI\_UNIQUE\_CNT, ONCLICK, ONDBLCLICK, ONKEYDOWN, ONKEYPRESS, ONKEYUP, ONMOUSEDOWN, ONMOUSEMOVE, ONMOUSEOUT, ONMOUSEOVER, ONMOUSEUP, STYLE, TABINDEX

#### Fields from `javax.servlet.jsp.tagext.BodyTagSupport`

bodyContent

#### Fields from `javax.servlet.jsp.tagext.TagSupport`

id, pageContext

#### Fields from interface `javax.servlet.jsp.tagext.BodyTag`

EVAL\_BODY\_BUFFERED, EVAL\_BODY\_TAG

#### Fields from interface `javax.servlet.jsp.tagext.IterationTag`

EVAL\_BODY\_AGAIN

#### Fields from interface `javax.servlet.jsp.tagext.Tag`

EVAL\_BODY\_INCLUDE, EVAL\_PAGE, SKIP\_BODY, SKIP\_PAGE

## Constructor Summary

*RepeaterHeader()*

## Method Summary

```

public
    int doAfterBody()
        Render the content of the body to the Repeater
        tag.

public
    String getTagName()
        Get the name of this tag.

protected
    void localRelease()
        Reset all of the fields of this tag.

```

```
public renderStartTag(int state)
    int          Render the start of this tag.
```

**Methods from com.bea.wlw.netui.tags.databinding.repeater.RepeaterComponent**

```
getCurrentRenderState, getRepeater, getValidContainerType, prepare,
renderEndTag
```

**Methods from com.bea.wlw.netui.tags.databinding.base.StructuredBaseTag**

```
addContent, doEndTag, doStartTag, getContent, getCurrentRenderState,
getValidContainerType, localRelease, prepare, renderEndTag,
verifyAttributes, verifyStructure
```

**Methods from com.bea.wlw.netui.tags.AbstractBaseTag**

```
addTagIdMapping, evaluateAttributeToString, filter, filter,
getJavaScriptUtils, getNearestForm, getScriptReporter, getUserLocale,
localRelease, registerTagError, release, renderAttribute, rewriteName,
write
```

**Methods from javax.servlet.jsp.tagext.BodyTagSupport**

```
doAfterBody, doEndTag, doInitBody, doStartTag, getBodyContent,
getPreviousOut, release, setBodyContent
```

**Methods from javax.servlet.jsp.tagext.TagSupport**

```
doAfterBody, doEndTag, doStartTag, findAncestorWithClass, getId,
getParent, getValue, getValues, release, removeValue, setId,
setPageContext, setParent, setValue
```

**Methods from class java.lang.Object**

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll,
toString, wait, wait, wait
```

**Methods from interface javax.servlet.jsp.tagext.BodyTag**

```
doInitBody, setBodyContent
```

**Methods from interface javax.servlet.jsp.tagext.IterationTag**

```
doAfterBody
```

**Methods from interface javax.servlet.jsp.tagext.Tag**

```
doEndTag, doStartTag, getParent, release, setPageContext, setParent
```

## Constructor Detail

### RepeaterHeader

```
public RepeaterHeader()
```

## Method Detail

### doAfterBody() Method

```
public int doAfterBody()
```

Render the content of the body to the Repeater tag.

#### *Overrides*

```
RepeaterComponent.doAfterBody()
```

#### Returns

*SKIP\_BODY*

---

### getTagName() Method

```
public String getTagName()
```

Get the name of this tag. This is used to identify the type of this tag for reporting tag errors.

#### *Overrides*

```
AbstractBaseTag.getTagName()
```

#### Returns

*a constant String representing the name of this tag.*

---

### localRelease() Method

```
protected void localRelease()
```

*Description copied from RepeaterComponent.localRelease()*

Reset all of the fields of this tag.

#### *Overrides*

```
RepeaterComponent.localRelease()
```

---



## renderStartTag(int) Method

```
public int renderStartTag(int state)
```

Render the start of this tag. If the repeater render state is `Repeater.HEADER`, the body is evaluated. Otherwise, the body is skipped.

### *Overrides*

`StructuredBaseTag.renderStartTag(int)`

### **Returns**

*SKIP\_BODY unless the repeater render state is `Repeater.HEADER`, then `EVAL_BODY_BUFFERED`*

# RepeaterItem Class

*public class* **RepeaterItem**

extends RepeaterComponent

This tag is used to render each item in the data set. The items in the data set are provided by this tag's Repeater parent and are available for databinding using the `container` databinding context. The current item is referenced using the `container.item` expression.

The RepeaterItem tag can only be used when directly contained by a Repeater tag, and a RepeaterItem is rendered during the Repeater's `Repeater.ITEM` rendering state.

By default, the RepeaterItem simply renders its body once for each of the items in the Repeater tag's data set.

A RepeaterItem has the ability to render different markup for each of the items that it renders. This is possible using the `ChoiceMethod` and `Choice` tags. These tags are used inside of the body to select which `Choice` tag to render and delineate each option that could be chosen.

## Related Topics

<netui-data:repeater> Tag Sample  
 Presenting Complex Data Sets in JSPs (Repeater Tags section)  
 Repeater  
 RepeaterHeader  
 RepeaterFooter  
 Choice

---

## Hierarchy

```

Object
  TagSupport
    BodyTagSupport
      AbstractBaseTag
        StructuredBaseTag
          RepeaterComponent
            RepeaterItem
  
```

## All Implemented Interfaces

BodyTag, IterationTag, Serializable, Tag

---

## Field Summary

```

public
static RENDER
  
```

`final int`      The RepeaterItem lifecycle state that runs through the body in order to render the body of the RepeaterItem.

#### Fields from `com.bea.wlw.netui.tags.databinding.repeater.RepeaterComponent`

`debug`

#### Fields from `com.bea.wlw.netui.tags.databinding.base.StructuredBaseTag`

`EMPTY_STRING`

#### Fields from `com.bea.wlw.netui.tags.AbstractBaseTag`

`ATTR_GENERAL`, `ATTR_GENERAL_EXPRESSION`, `ATTR_JAVASCRIPT`, `ATTR_STYLE`, `CLASS`, `ID`, `JAVASCRIPT_STATUS`, `NAME`, `NETUI_UNIQUE_CNT`, `ONCLICK`, `ONDBLCLICK`, `ONKEYDOWN`, `ONKEYPRESS`, `ONKEYUP`, `ONMOUSEDOWN`, `ONMOUSEMOVE`, `ONMOUSEOUT`, `ONMOUSEOVER`, `ONMOUSEUP`, `STYLE`, `TABINDEX`

#### Fields from `javax.servlet.jsp.tagext.BodyTagSupport`

`bodyContent`

#### Fields from `javax.servlet.jsp.tagext.TagSupport`

`id`, `pageContext`

#### Fields from interface `javax.servlet.jsp.tagext.BodyTag`

`EVAL_BODY_BUFFERED`, `EVAL_BODY_TAG`

#### Fields from interface `javax.servlet.jsp.tagext.IterationTag`

`EVAL_BODY_AGAIN`

#### Fields from interface `javax.servlet.jsp.tagext.Tag`

`EVAL_BODY_INCLUDE`, `EVAL_PAGE`, `SKIP_BODY`, `SKIP_PAGE`

## Constructor Summary

*`RepeaterItem()`*

## Method Summary

`public`  
    `int doAfterBody()`

Render the result of having evalutaed the body.

```

public
Object getChoiceValue()
    Get the value of a nested Choice tag whose value attribute corresponds to the value
    selected by executing the ChoiceMethod tag.

public
    int getRenderState()
        Get the current render state of the RepeaterItem.

public
String getTagName()
    Get the name of this tag.

public
boolean hasRenderedItem()
    Get a boolean that denotes that an item has been rendered for this RepeaterItem.
protected
    void localRelease()
        Reset all of the fields of this tag.

public
    int renderStartTag(int state)
        Render the start of this tag.

public
    void setChoiceValue(Object choiceValue)
        Set the value on a Choice tag whose value attribute matched the result of the result
        of invoking the ChoiceMethod.

public
    void setRenderedItem(boolean renderedItem)
        A boolean that denotes that an item has been rendered from the set of Choice tags
        that may be present in the body of a RepeaterItem tag.

public
boolean shouldRender(Object value, boolean defaultItem)
    The decision method that the Choice tags use to determine if the calling Choice tag
    is the one that hsould render.

public
    void stopRendering()

```

### Methods from `com.bea.wlw.netui.tags.databinding.repeater.RepeaterComponent`

`getCurrentRenderState`, `getRepeater`, `getValidContainerType`, `prepare`, `renderEndTag`

### Methods from `com.bea.wlw.netui.tags.databinding.base.StructuredBaseTag`

`addContent`, `doEndTag`, `doStartTag`, `getContent`, `getCurrentRenderState`, `getValidContainerType`, `localRelease`, `prepare`, `renderEndTag`, `verifyAttributes`, `verifyStructure`

### Methods from `com.bea.wlw.netui.tags.AbstractBaseTag`

`addTagIdMapping, evaluateAttributeToString, filter, filter,  
getJavaScriptUtils, getNearestForm, getScriptReporter, getUserLocale,  
localRelease, registerTagError, release, renderAttribute, rewriteName,  
write`

### Methods from `javax.servlet.jsp.tagext.BodyTagSupport`

`doAfterBody, doEndTag, doInitBody, doStartTag, getBodyContent,  
getPreviousOut, release, setBodyContent`

### Methods from `javax.servlet.jsp.tagext.TagSupport`

`doAfterBody, doEndTag, doStartTag, findAncestorWithClass, getId,  
getParent, getValue, getValues, release, removeValue, setId,  
setPageContext, setParent, setValue`

### Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll,  
toString, wait, wait, wait`

### Methods from interface `javax.servlet.jsp.tagext.BodyTag`

`doInitBody, setBodyContent`

### Methods from interface `javax.servlet.jsp.tagext.IterationTag`

`doAfterBody`

### Methods from interface `javax.servlet.jsp.tagext.Tag`

`doEndTag, doStartTag, getParent, release, setPageContext, setParent`

## Field Detail

### RENDER

```
public static final int RENDER
```

The RepeaterItem lifecycle state that runs through the body in order to render the body of the RepeaterItem.

## Constructor Detail

## RepeaterItem

```
public RepeaterItem()
```

## Method Detail

### doAfterBody() Method

```
public int doAfterBody()
    throws JspException
```

Render the result of having evalutaed the body. Generally, the body is evaluted once, but when `ChoiceMethod` and `Choice` tags are present, the body is evaluated twice; once to decide which `Choice` to render and once to actually render the body of that `Choice` tag.

#### Overrides

```
RepeaterComponent.doAfterBody()
```

#### Returns

*SKIP\_BODY* if the item has completed rendering; *EVAL\_BODY\_BUFFERED* if the body should render again.

#### Exceptions

*JspException*

---

### getChoiceValue() Method

```
public Object getChoiceValue()
```

Get the value of a nested `Choice` tag whose value attribute corresponds to the value selected by executing the `ChoiceMethod` tag. This value is used to determine which `Choice` tag should render its body.

#### Returns

*the value attribute from a nested Choice tag that matched the return value of the ChoiceMethod.*

---

### getRenderState() Method

```
public int getRenderState()
```

Get the current render state of the `RepeaterItem`.

**Returns**

*the RepeaterItem's render state.*

---

**getTagName() Method**

```
public String getTagName()
```

Get the name of this tag. This is used to identify the type of this tag for reporting tag errors.

**Overrides**

```
AbstractBaseTag.getTagName()
```

**Returns**

*a constant String representing the name of this tag.*

---

**hasRenderedItem() Method**

```
public boolean hasRenderedItem()
```

Get a boolean that denotes that an item has been rendered for this RepeaterItem. If there are no Choice tags, this will always return false and the body of the RepeaterItem will always render for each item.

**Returns**

*true if the Choice functionality is being used and a Choice has rendered; false otherwise.*

---

**localRelease() Method**

```
protected void localRelease()
```

Reset all of the fields of this tag.

**Overrides**

```
RepeaterComponent.localRelease()
```

---

**renderStartTag(int) Method**

```
public int renderStartTag(int state)
```

Render the start of this tag. If the repeater render state is Repeater.ITEM, the body is evaluated. Otherwise, the body is skipped.

**Overrides**

```
StructuredBaseTag.renderStartTag(int)
```

**Returns**

*SKIP\_BODY unless the repeater render state is `Repeater.ITEM`, then `EVAL_BODY_BUFFERED`*

---

**setChoiceValue(Object) Method**

```
public void setChoiceValue(Object choiceValue)
```

Set the value on a Choice tag whose value attribute matched the result of the result of invoking the ChoiceMethod.

**Parameters**

*choiceValue*  
the result of executing the ChoiceMethod

---

**setRenderedItem(boolean) Method**

```
public void setRenderedItem(boolean renderedItem)
```

A boolean that denotes that an item has been rendered from the set of Choice tags that may be present in the body of a RepeaterItem tag.

---

**shouldRender(Object, boolean) Method**

```
public boolean shouldRender(Object value,
                           boolean defaultItem)
```

The decision method that the Choice tags use to determine if the calling Choice tag is the one that should render. The rules for deciding if a Choice tag should render are as follows:

- If there was no value attribute on one of the nested Choice that matched the result of the ChoiceMethod's decision method and the calling Choice tag's `Choice.setDefault(boolean)` is true, this method returns true.
- If the value param matches the `RepeaterItem.getChoiceValue()` property, this method returns true.
- Otherwise, it returns false.

**Parameters**

*value*  
the `Choice.setValue(String)` attribute on the Choice tag that is asking if it should render its body.

*defaultItem*  
the `Choice.setDefault(boolean)` attribute on the Choice tag that is asking if it should render its body.



## Returns

*true if the Choice should render; false otherwise.*

---

## stopRendering() Method

```
public void stopRendering()
```

# **com.bea.wlw.netui.tags.databinding.repeater.choice Package**

## **Class Summary**

<code>Choice</code>	A tag that when nested in a <code>RepeaterItem</code> allows its body to be rendered conditionally.
<code>ChoiceMethod</code>	A <code>ChoiceMethod</code> tag is used in order to choose a particular <code>Choice</code> tag whose body should be rendered for the current data item in the <code>Repeater</code> .

# Choice Class

*public class* **Choice**

extends RepeaterComponent

A tag that when nested in a RepeaterItem allows its body to be rendered conditionally. The Choice tag must be used together with the ChoiceMethod tag in order to achieve conditional rendering of the body of a particular Choice tag. A single ChoiceMethod and one or more Choice tags should be nested within a RepeaterItem. The RepeaterItem then uses the ChoiceMethod to invoke a decision method that returns the value. The return value is then compared against the value attribute on each Choice tag in order to select the Choice whose body should be rendered. If no match is found but one of the Choice tags has its default attribute set, this "default" Choice will be rendered.

Attribute Descriptions			
Attribute	Required	Runtime Expression Evaluation	Data Bindable
default	No	No	No
	A boolean that if true will render this Choice's body if no Choice's value matches the result of the ChoiceMethod.		
value	No	No	Yes
	A String value that will be matched against the value returned from invoking the ChoiceMethod.		

## Related Topics

<netui-data:choice> Tag Sample

ChoiceMethod

Presenting Complex Data Sets in JSPs (Repeater Tags section)

## Hierarchy

```

Object
  TagSupport
    BodyTagSupport
      AbstractBaseTag
        StructuredBaseTag
          RepeaterComponent
            Choice
  
```

## All Implemented Interfaces

BodyTag, IterationTag, Serializable, Tag

## Field Summary

### Fields from `com.bea.wlw.netui.tags.databinding.repeater.RepeaterComponent`

`debug`

### Fields from `com.bea.wlw.netui.tags.databinding.base.StructuredBaseTag`

`EMPTY_STRING`

### Fields from `com.bea.wlw.netui.tags.AbstractBaseTag`

`ATTR_GENERAL`, `ATTR_GENERAL_EXPRESSION`, `ATTR_JAVASCRIPT`, `ATTR_STYLE`, `CLASS`, `ID`, `JAVASCRIPT_STATUS`, `NAME`, `NETUI_UNIQUE_CNT`, `ONCLICK`, `ONDBLCLICK`, `ONKEYDOWN`, `ONKEYPRESS`, `ONKEYUP`, `ONMOUSEDOWN`, `ONMOUSEMOVE`, `ONMOUSEOUT`, `ONMOUSEOVER`, `ONMOUSEUP`, `STYLE`, `TABINDEX`

### Fields from `javax.servlet.jsp.tagext.BodyTagSupport`

`bodyContent`

### Fields from `javax.servlet.jsp.tagext.TagSupport`

`id`, `pageContext`

### Fields from interface `javax.servlet.jsp.tagext.BodyTag`

`EVAL_BODY_BUFFERED`, `EVAL_BODY_TAG`

### Fields from interface `javax.servlet.jsp.tagext.IterationTag`

`EVAL_BODY_AGAIN`

### Fields from interface `javax.servlet.jsp.tagext.Tag`

`EVAL_BODY_INCLUDE`, `EVAL_PAGE`, `SKIP_BODY`, `SKIP_PAGE`

## Constructor Summary

*Choice()*

## Method Summary

```

public
    int doAfterBody()
        Handle the content that may have been rendered in the body by adding it to the
        RepeaterItem's content.

public
    Object getActualValue()

protected
    int getCurrentRenderState()
        Get the current render state of this tag's RepeaterItem parent.

public
    boolean getDefault()
        Get the boolean value of the default attribute.

public
    String getTagName()
        Get the name of this tag.

protected
    Class getValidContainerType()
        Get the valid container type of this tag, the RepeaterItem.

public
    String getValue()
        Get the String value that will be checked for a match in the ChoiceMethod.

protected
    void localRelease()
        Reset all of the fields of this tag.

protected
    void prepare()
        Prepare data that is provided to subclasses of this tag.

public
    int renderStartTag(int state)
        Conditionally render the body of this tag.

protected
    void reportErrors()
        This will report an expression error.

public
    void setDefault(boolean defaultItem)
        Sets a boolean value that, if true will mark a Choice tag as the "default" Choice if no
        other Choices match the value returned in invoking the ChoiceMethod.

public
    void setValue(String value)
        Set a String value that is used by the RepeaterItem to select this Choice tag if this
        value is returned by the ChoiceMethod.

public
    String toString()
        Get a debugging String that represents a PadContext.

```

**Methods from `com.bea.wlw.netui.tags.databinding.repeater.RepeaterComponent`**

`getRepeater`, `renderEndTag`

**Methods from `com.bea.wlw.netui.tags.databinding.base.StructuredBaseTag`**

`addContent`, `doEndTag`, `doStartTag`, `getContent`, `getCurrentRenderState`, `getValidContainerType`, `localRelease`, `prepare`, `renderEndTag`, `verifyAttributes`, `verifyStructure`

**Methods from `com.bea.wlw.netui.tags.AbstractBaseTag`**

`addTagIdMapping`, `evaluateAttributeToString`, `filter`, `filter`, `getJavaScriptUtils`, `getNearestForm`, `getScriptReporter`, `getUserLocale`, `localRelease`, `registerTagError`, `release`, `renderAttribute`, `rewriteName`, `write`

**Methods from `javax.servlet.jsp.tagext.BodyTagSupport`**

`doAfterBody`, `doEndTag`, `doInitBody`, `doStartTag`, `getBodyContent`, `getPreviousOut`, `release`, `setBodyContent`

**Methods from `javax.servlet.jsp.tagext.TagSupport`**

`doAfterBody`, `doEndTag`, `doStartTag`, `findAncestorWithClass`, `getId`, `getParent`, `getValue`, `getValues`, `release`, `removeValue`, `setId`, `setPageContext`, `setParent`, `setValue`

**Methods from class `java.lang.Object`**

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `wait`, `wait`, `wait`

**Methods from interface `javax.servlet.jsp.tagext.BodyTag`**

`doInitBody`, `setBodyContent`

**Methods from interface `javax.servlet.jsp.tagext.IterationTag`**

`doAfterBody`

**Methods from interface `javax.servlet.jsp.tagext.Tag`**

`doEndTag`, `doStartTag`, `getParent`, `release`, `setPageContext`, `setParent`

## Constructor Detail

## Choice

```
public Choice()
```

## Method Detail

### doAfterBody() Method

```
public int doAfterBody()
```

Handle the content that may have been rendered in the body by adding it to the `RepeaterItem`'s content.

#### *Overrides*

```
RepeaterComponent.doAfterBody()
```

#### Returns

*SKIP\_BODY*

---

### getActualValue() Method

```
public Object getActualValue()
```

---

### getCurrentRenderState() Method

```
protected int getCurrentRenderState()
```

Get the current render state of this tag's `RepeaterItem` parent.

#### *Overrides*

```
RepeaterComponent.getCurrentRenderState()
```

#### Returns

*the render state of the parent RepeaterItem*

---

### getDefault() Method

```
public boolean getDefault()
```

Get the boolean value of the default attribute.

**Returns**

*true if this Choice tag is the default choice in a Repeater body; false otherwise*

---

**getTagName() Method**

```
public String getTagName()
```

Get the name of this tag. This is used to identify the type of this tag for reporting tag errors.

**Overrides**

```
AbstractBaseTag.getTagName()
```

**Returns**

*a constant String representing the name of this tag.*

---

**getValidContainerType() Method**

```
protected Class getValidContainerType()
```

Get the valid container type of this tag, the RepeaterItem.

**Overrides**

```
RepeaterComponent.getValidContainerType()
```

**Returns**

*the valid container type*

---

**getValue() Method**

```
public String getValue()
```

Get the String value that will be checked for a match in the ChoiceMethod.

**Returns**

*the value that should be matched or null if unset*

---

**localRelease() Method**

```
protected void localRelease()
```

Reset all of the fields of this tag.



**Overrides**

```
RepeaterComponent.localRelease()
```

---

**prepare() Method**

```
protected void prepare()
```

Prepare data that is provided to subclasses of this tag.

**Overrides**

```
RepeaterComponent.prepare()
```

---

**renderStartTag(int) Method**

```
public int renderStartTag(int state)
    throws JspException
```

Conditionally render the body of this tag. The body is rendered if the `RepeaterItem.shouldRender(Object, boolean)` call returns true and the `RepeaterItem` is in its `RepeaterItem.RENDER` state. Otherwise, the body is skipped.

**Overrides**

```
StructuredBaseTag.renderStartTag(int)
```

**Parameters**

*state*  
the tag's current render state

**Returns**

*SKIP\_BODY* unless the repeater item should render; then return *EVAL\_BODY\_BUFFERED*

**Exceptions**

*JspException*

---

**reportErrors() Method**

```
protected void reportErrors()
    throws JspException
```

*Description copied from AbstractBaseTag.reportErrors()*  
This will report an expression error.

**Overrides**

```
AbstractBaseTag.reportErrors()
```

## Exceptions

*JspException*

---

## setDefault(boolean) Method

```
public void setDefault(boolean defaultItem)
```

Sets a boolean value that, if `true` will mark a Choice tag as the "default" Choice if no other Choices match the value returned in invoking the ChoiceMethod.

### Parameters

*defaultItem*

a boolean that denotes this item as the "default" to render

---

## setValue(String) Method

```
public void setValue(String value)
```

Set a String value that is used by the RepeaterItem to select this Choice tag if this value is returned by the ChoiceMethod. The RepeaterItem selects a Choice based on the value returned by running the ChoiceMethod tag; this value is compared against the value attribute here. If a match is found in performing this comparison, the matching Choice tag is rendered in the RepeaterItem.

### Parameters

*value*

the value that should be matched in order to render this Choice

---

## toString() Method

```
public String toString()
```

Get a debugging String that represents a PadContext.

### Overrides

`Object.toString()`

### Returns

*a String representation of the PadContext*

# ChoiceMethod Class

*public class ChoiceMethod*

extends CallMethod

A ChoiceMethod tag is used in order to choose a particular Choice tag whose body should be rendered for the current data item in the Repeater. The ChoiceMethod tag invokes a method on an object and passes parameters from its nested MethodParameter tags. The value of each of these parameters may be taken from the "current" data item using the `container.item` databinding syntax. The result of evaluating this method is set on the RepeaterItem through the `RepeaterItem.setChoiceValue(Object)` call. The Choice tags then reach to their RepeaterItem to determine if the body of a particular Choice tag should be rendered.

Attribute Descriptions			
Attribute	Required	Runtime Expression Evaluation	Data Bindable
failOnError	No	No	No
	A boolean that if <code>true</code> will cause the tag to fail if any errors are reported invoking the method.		
method	No	No	No
	Name of the method in the page flow that will be invoked.		
object	Yes	No	Yes
	References an object with an expression on which a choice method should be invoked.		
resultId	No	No	No
	The <code>PageContext</code> attribute name where the result, if non-null, is stored.		

Additional attributes can be found on the CallMethod tag.

## Related Topics

<netui-data:choice> Tag Sample

Choice

Presenting Complex Data Sets in JSPs (Repeater Tags section)

## Hierarchy

Object

TagSupport

BodyTagSupport

AbstractBaseTag  
 AbstractCallMethod  
 CallMethod  
 ChoiceMethod

### *All Implemented Interfaces*

BodyTag, IterationTag, Serializable, Tag

---

## Field Summary

### Fields from `com.bea.wlw.netui.tags.AbstractBaseTag`

ATTR\_GENERAL, ATTR\_GENERAL\_EXPRESSION, ATTR\_JAVASCRIPT, ATTR\_STYLE,  
 CLASS, ID, JAVASCRIPT\_STATUS, NAME, NETUI\_UNIQUE\_CNT, ONCLICK,  
 ONDBLCLICK, ONKEYDOWN, ONKEYPRESS, ONKEYUP, ONMOUSEDOWN, ONMOUSEMOVE,  
 ONMOUSEOUT, ONMOUSEOVER, ONMOUSEUP, STYLE, TABINDEX

### Fields from `javax.servlet.jsp.tagext.BodyTagSupport`

bodyContent

### Fields from `javax.servlet.jsp.tagext.TagSupport`

id, pageContext

### Fields from interface `javax.servlet.jsp.tagext.BodyTag`

EVAL\_BODY\_BUFFERED, EVAL\_BODY\_TAG

### Fields from interface `javax.servlet.jsp.tagext.IterationTag`

EVAL\_BODY\_AGAIN

### Fields from interface `javax.servlet.jsp.tagext.Tag`

EVAL\_BODY\_INCLUDE, EVAL\_PAGE, SKIP\_BODY, SKIP\_PAGE

## Constructor Summary

*ChoiceMethod()*

## Method Summary

```

public
    int doEndTag()
        End this tag's rendering lifecycle.

public
    int doStartTag()
        Register this tag with its RepeaterItem parent.

public
    String getObject()
        Get the expression that references an object on which to invoke the decision
        method.

protected
    String getObjectName()
        Get the name of the object that is the target of the invocation.

public
    String getTagName()
        Get the name of this tag.

protected
    void handleReturnValue(Object result)
        Handle any value that may be returned from the decision method.

protected
    void localRelease()
        Reset all of the fields of this tag.

protected
    void reportErrors()
        This will report an expression error.

protected
    Object resolveObject()
        Get the object on which an expression the choice method should be executed.

public
    void setObject(String object)
        Set the object on which the decision method should be invoked.

```

### Methods from `com.bea.wlw.netui.tags.databinding.invoke.CallMethod`

`findMethod`,

### Methods from `com.bea.wlw.netui.tags.databinding.invoke.AbstractCallMethod`

`addParameter`, `findMethod`, `getFailOnError`, `getMethod`, `getObjectName`,  
`getParameterNodes`, `getResultId`, `localRelease`, `resolveObject`,  
`setFailOnError`, `setMethod`, `getResultId`

### Methods from `com.bea.wlw.netui.tags.AbstractBaseTag`

`addTagIdMapping`, `evaluateAttributeToString`, `filter`, `filter`,  
`getJavaScriptUtils`, `getNearestForm`, `getScriptReporter`, `getTagName`,

`getUserLocale, localRelease, registerTagError, release, renderAttribute, rewriteName, write`

### Methods from `javax.servlet.jsp.tagext.BodyTagSupport`

`doAfterBody, doEndTag, doInitBody, doStartTag, getBodyContent, getPreviousOut, release, setBodyContent`

### Methods from `javax.servlet.jsp.tagext.TagSupport`

`doAfterBody, doEndTag, doStartTag, findAncestorWithClass, getId, getParent, getValue, getValues, release, removeValue, setId, setPageContext, setParent, setValue`

### Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

### Methods from interface `javax.servlet.jsp.tagext.BodyTag`

`doInitBody, setBodyContent`

### Methods from interface `javax.servlet.jsp.tagext.IterationTag`

`doAfterBody`

### Methods from interface `javax.servlet.jsp.tagext.Tag`

`doEndTag, doStartTag, getParent, release, setPageContext, setParent`

## Constructor Detail

### ChoiceMethod

```
public ChoiceMethod()
```

## Method Detail

### doEndTag() Method

```
public int doEndTag()  
    throws JspException
```

End this tag's rendering lifecycle.

### *Overrides*

```
AbstractCallMethod.doEndTag()
```

### Returns

*EVAL\_PAGE*

### Exceptions

*JspException*

if an error is reported that can not be reported in the page

---

## doStartTag() Method

```
public int doStartTag()
    throws JspException
```

Register this tag with its `RepeaterItem` parent. The `RepeaterItem` will adjust its rendering lifecycle based on the presence of a `ChoiceMethod` tag.

### Overrides

```
AbstractCallMethod.doStartTag()
```

### Returns

*EVAL\_BODY\_BUFFERED*; if an error occurs, return *SKIP\_BODY*

### Exceptions

*JspException*

if an error is reported that can not be reported in the page

---

## getObject() Method

```
public String getObject()
```

Get the expression that references an object on which to invoke the decision method.

### Overrides

```
CallMethod.getObject()
```

### Returns

*the value of the object attribute or null if unset*

---

## getObjectName() Method

```
protected String getObjectName()
```

Get the name of the object that is the target of the invocation.

**Overrides**

`CallMethod.getObjectName()`

**Returns**

*a name for the object on which the method will be invoked.*

---

## getTagName() Method

`public String getTagName()`

Get the name of this tag. This is used to identify the type of this tag for reporting tag errors.

**Overrides**

`CallMethod.getTagName()`

**Returns**

*a constant String representing the name of this tag.*

---

## handleReturnValue(Object) Method

`protected void handleReturnValue(Object result)`

Handle any value that may be returned from the decision method. The result of invoking the decision method is set on the `RepeaterItem` so that it can be accessed by the `Choice` tags in a repeater tag's `RepeaterItem` tag.

**Overrides**

`AbstractCallMethod.handleReturnValue(Object)`

**Parameters**

*result*

the object that resulted from invoking decision method

---

## localRelease() Method

`protected void localRelease()`

Reset all of the fields of this tag.

**Overrides**

`CallMethod.localRelease()`

---



## reportErrors() Method

```
protected void reportErrors()
    throws JspException
```

*Description copied from AbstractBaseTag.reportErrors()*  
This will report an expression error.

### Overrides

```
AbstractBaseTag.reportErrors()
```

### Exceptions

*JspException*

---

## resolveObject() Method

```
protected Object resolveObject()
    throws ObjectNotFoundException
```

Get the object on which an expression the choice method should be executed.

### Overrides

```
CallMethod.resolveObject()
```

### Returns

*the object on which to invoke the decision method*

### Exceptions

*ObjectNotFoundException*

---

## setObject(String) Method

```
public void setObject(String object)
```

Set the object on which the decision method should be invoked. The value of this attribute should be an expression

### Overrides

```
CallMethod.setObject(String)
```

### Parameters

*object*

the object on which to invoke a method

## Returns

*the value of the object attribute or null if unset*

# **com.bea.wlw.netui.tags.databinding.repeater.pad Package**

## **Class Summary**

**Pad**

A tag that affects the number of items that are rendered in a `Repeater`.

**PadContext** A `JavaBean` that encapsulates the data needed to pad a `Repeater` with text.

## Pad Class

*public class* **Pad**

extends RepeaterComponent

A tag that affects the number of items that are rendered in a `Repeater`. The `Pad` tag has the ability to turn an irregular data set in the `Repeater` into a regular dataset through the use of the three attributes on the `Pad` tag. The `Pad` tag can truncate the rendering of a data set if enough items have been rendered or can pad a default item text onto the end of a data set that is not big enough. If the `padText` attribute is unset, the body of the `Pad` tag is used as the `padText`.

Note, the `container` data binding context should not be used inside of the `Pad`'s `padText` attribute as binding to both the `item` and `index` could product unexpected results.

Attribute Descriptions			
Attribute	Required	Runtime Expression Evaluation	Data Bindable
<b>maxRepeat</b>	No	No	Yes
	The maximum number of times a <code>RepeaterItem</code> will render.		
<b>minRepeat</b>	No	No	Yes
	The minimum number of times a <code>RepeaterItem</code> will render.		
<b>padText</b>	No	No	Yes
	The text to use when padding a <code>Repeater</code> that has rendered too few items.		

### Related Topics

<netui-data:pad> Tag Sample  
Repeater

---

### Hierarchy

```

Object
  TagSupport
    BodyTagSupport
      AbstractBaseTag
        StructuredBaseTag
          RepeaterComponent
            Pad
  
```

### All Implemented Interfaces

## Field Summary

### Fields from `com.bea.wlw.netui.tags.databinding.repeater.RepeaterComponent`

`debug`

### Fields from `com.bea.wlw.netui.tags.databinding.base.StructuredBaseTag`

`EMPTY_STRING`

### Fields from `com.bea.wlw.netui.tags.AbstractBaseTag`

`ATTR_GENERAL`, `ATTR_GENERAL_EXPRESSION`, `ATTR_JAVASCRIPT`, `ATTR_STYLE`, `CLASS`, `ID`, `JAVASCRIPT_STATUS`, `NAME`, `NETUI_UNIQUE_CNT`, `ONCLICK`, `ONDBLCLICK`, `ONKEYDOWN`, `ONKEYPRESS`, `ONKEYUP`, `ONMOUSEDOWN`, `ONMOUSEMOVE`, `ONMOUSEOUT`, `ONMOUSEOVER`, `ONMOUSEUP`, `STYLE`, `TABINDEX`

### Fields from `javax.servlet.jsp.tagext.BodyTagSupport`

`bodyContent`

### Fields from `javax.servlet.jsp.tagext.TagSupport`

`id`, `pageContext`

### Fields from interface `javax.servlet.jsp.tagext.BodyTag`

`EVAL_BODY_BUFFERED`, `EVAL_BODY_TAG`

### Fields from interface `javax.servlet.jsp.tagext.IterationTag`

`EVAL_BODY_AGAIN`

### Fields from interface `javax.servlet.jsp.tagext.Tag`

`EVAL_BODY_INCLUDE`, `EVAL_PAGE`, `SKIP_BODY`, `SKIP_PAGE`

## Constructor Summary

*Pad()*

## Method Summary

```

public
    int doAfterBody()
        Set the pad text from the content that was rendered in the body.

public
    String getMaxRepeat()
        Get the maximum number of times that the RepeaterItem will be rendered.

public
    String getMinRepeat()
        Get the minimum number of items to render.

public
    String getPadText()
        Get the text that is used to pad the items in a Repeater.

public
    String getTagName()
        Get the name of this tag.

protected
    void localRelease()
        Reset all of the fields of this tag.

public
    int renderEndTag(int state)
        Complete rendering the body of this tag.

public
    int renderStartTag(int state)
        Render the body of this tag and buffer the output.

public
    void setMaxRepeat(String maxRepeatString)
        Set the maximum number of items that will be rendered by a Repeater.

public
    void setMinRepeat(String minRepeatString)
        Set the minimum number of items that will be rendered by a Repeater.

public
    void setPadText(String padText)
        Set the text that will be used when padding a Repeater that renders its body too few
        times.

protected
    void verifyAttributes()
        Verify that the attributes set on the tag are correct.

```

**Methods from `com.bea.wlw.netui.tags.databinding.repeater.RepeaterComponent`**

`getCurrentRenderState`, `getRepeater`, `getValidContainerType`, `prepare`,

**Methods from `com.bea.wlw.netui.tags.databinding.base.StructuredBaseTag`**

`addContent`, `doEndTag`, `doStartTag`, `getContent`, `getCurrentRenderState`, `getValidContainerType`, `localRelease`, `prepare`, `renderEndTag`, `verifyStructure`

### Methods from `com.bea.wlw.netui.tags.AbstractBaseTag`

`addTagIdMapping`, `evaluateAttributeToString`, `filter`, `filter`, `getJavaScriptUtils`, `getNearestForm`, `getScriptReporter`, `getUserLocale`, `localRelease`, `registerTagError`, `release`, `renderAttribute`, `rewriteName`, `write`

### Methods from `javax.servlet.jsp.tagext.BodyTagSupport`

`doAfterBody`, `doEndTag`, `doInitBody`, `doStartTag`, `getBodyContent`, `getPreviousOut`, `release`, `setBodyContent`

### Methods from `javax.servlet.jsp.tagext.TagSupport`

`doAfterBody`, `doEndTag`, `doStartTag`, `findAncestorWithClass`, `getId`, `getParent`, `getValue`, `getValues`, `release`, `removeValue`, `setId`, `setPageContext`, `setParent`, `setValue`

### Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

### Methods from interface `javax.servlet.jsp.tagext.BodyTag`

`doInitBody`, `setBodyContent`

### Methods from interface `javax.servlet.jsp.tagext.IterationTag`

`doAfterBody`

### Methods from interface `javax.servlet.jsp.tagext.Tag`

`doEndTag`, `doStartTag`, `getParent`, `release`, `setPageContext`, `setParent`

## Constructor Detail

### Pad

```
public Pad()
```

## Method Detail

### doAfterBody() Method

```
public int doAfterBody()
```

Set the pad text from the content that was rendered in the body. This value is only set when the `padText` property has not been set.

#### *Overrides*

```
RepeaterComponent.doAfterBody()
```

#### Returns

*SKIP\_BODY*

---

### getMaxRepeat() Method

```
public String getMaxRepeat()
```

Get the maximum number of times that the `RepeaterItem` will be rendered.

#### Returns

*an integer or an expression that references an integer for the maximum number of items to render or null if unset*

---

### getMinRepeat() Method

```
public String getMinRepeat()
```

Get the minimum number of items to render.

#### Returns

*an integer or an expression that references an integer for the minimum number of items to render*

---

### getPadText() Method

```
public String getPadText()
```

Get the text that is used to pad the items in a `Repeater`.

#### Returns

*the text that is used to pad a repeater's content*



## getTagName() Method

```
public String getTagName()
```

Get the name of this tag. This is used to identify the type of this tag for reporting tag errors.

### *Overrides*

```
AbstractBaseTag.getTagName()
```

### Returns

*a constant String representing the name of this tag.*

---

## localRelease() Method

```
protected void localRelease()
```

Reset all of the fields of this tag.

### *Overrides*

```
RepeaterComponent.localRelease()
```

---

## renderEndTag(int) Method

```
public int renderEndTag(int state)
    throws JspException
```

Complete rendering the body of this tag. If the padText property was unset, the body of the tag is used as the pad text.

### *Overrides*

```
RepeaterComponent.renderEndTag(int)
```

### Parameters

*state*

the current render state of the Repeater

### Returns

*EVAL\_PAGE*

### Exceptions

*JspException*

if an error occurs that can not be reported on the page

---

## renderStartTag(int) Method

```
public int renderStartTag(int state)
    throws JspException
```

Render the body of this tag and buffer the output. If the padText attribute is unset, the body content is used instead.

### Overrides

StructuredBaseTag.renderStartTag(int)

### Exceptions

*JspException*

---

## setMaxRepeat(String) Method

```
public void setMaxRepeat(String maxRepeatString)
```

Set the maximum number of items that will be rendered by a Repeater. If the size of the data set is greater than this, only this many items will be rendered. This value can be a String of an integer or can be an expression that references an integer.

### Parameters

*maxRepeatString*

an integer or an expression that references an integer for the maximum number of items to render

---

## setMinRepeat(String) Method

```
public void setMinRepeat(String minRepeatString)
```

Set the minimum number of items that will be rendered by a Repeater. If the size of the data set is smaller than this, the data set will be padded with value of the pad text attribute.

### Parameters

*minRepeatString*

an integer or an expression that references an integer for the minimum number of items to render

---

## setPadText(String) Method

```
public void setPadText(String padText)
```

Set the text that will be used when padding a Repeater that renders its body too few times.

## Parameters

*padText*

the text that is used to pad a repeater's content

---

## verifyAttributes() Method

```
protected void verifyAttributes()  
    throws JspException
```

Verify that the attributes set on the tag are correct. Check:

- The minRepeat value can be converted into an integer
- The maxRepeat value can be converted into an integer
- The minRepeat value is less than or equal to the maxRepeat value

### Overrides

StructuredBaseTag.verifyAttributes()

## Exceptions

*JspException*

if an error occurs that can not be reported on the page

# PadContext Class

*public class PadContext*

extends Object

A JavaBean that encapsulates the data needed to pad a Repeater with text. The PadContext is needed if the given constraints, the minRepeat and maxRepeat attributes, require that the data rendered in the repeater be padded.

## Hierarchy

```

Object
  PadContext
  
```

## Field Summary

## Constructor Summary

*PadContext(String padText, int minRepeat, int maxRepeat)*

## Method Summary

```

public
boolean checkMaxRepeat(int currCount)

public
boolean checkMinRepeat(int currCount)

public
int getMaxRepeat()
    Get the maximum number of times to render items in the repeater's body.

public
int getMinRepeat()
    Get the minimum number of times to render an item in the repeater's
    body.

public
String getPadText()
    Get the text to use when padding the Repeater

public
  
```

String **toString()**

Get a debugging String that represents a PadContext.

### Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `wait`, `wait`, `wait`

## Field Detail

## Constructor Detail

### PadContext

```
public PadContext(String padText,
                  int minRepeat,
                  int maxRepeat)
```

## Method Detail

### checkMaxRepeat(int) Method

```
public boolean checkMaxRepeat(int currCount)
```

#### Parameters

*currCount*  
the count of the number of items rendered so far

#### Returns

*true if the maximum number of items have been rendered; false otherwise*

---

### checkMinRepeat(int) Method

```
public boolean checkMinRepeat(int currCount)
```

#### Parameters

*currCount*  
the count of the number of items rendered.

#### Returns

*true if the minimum number of items have been rendered; false otherwise*

---

## getMaxRepeat() Method

```
public int getMaxRepeat()
```

Get the maximum number of times to render items in the repeater's body.

### Returns

*the maximum number of items to render items in the repeater's body*

---

## getMinRepeat() Method

```
public int getMinRepeat()
```

Get the minimum number of times to render an item in the repeater's body.

### Returns

*the minimum number of items that must be rendered in the repeater's body*

---

## getPadText() Method

```
public String getPadText()
```

Get the text to use when padding the Repeater

### Returns

*the pad text*

---

## toString() Method

```
public String toString()
```

Get a debugging String that represents a PadContext.

### Overrides

```
Object.toString()
```

### Returns

*a String representation of the PadContext*

# **com.bea.wlw.netui.tags.databinding.script Package**

## **Class Summary**

### **GetData**

This tag evaluates an expression and places the result of expression evaluation in a JSP's `PageContext`.

# GetData Class

*public class* **GetData**

extends `AbstractBaseTag`

This tag evaluates an expression and places the result of expression evaluation in a JSP's `PageContext`. This can be used to extract objects from forms, page flows, and other objects that can be databound; once in the JSP, scriptlet can be written using these objects by accessing them through the `PageContext.getAttribute(String)` method.

Attribute Descriptions			
Attribute	Required	Runtime Expression Evaluation	Data Bindable
<b>resultId</b>	Yes	No	Yes
	The String identifier under which the Object resulting from expression evaluation will be stored.		
<b>value</b>	Yes	No	Yes
	The expression that references the data to store in the <code>PageContext</code> attribute map.		

## Related Topics

<netui-data:getData> Tag Sample  
 Using Data Binding in Page Flows (`PageContext` section)  
[PageContext](#)

---

## Hierarchy

```

Object
  TagSupport
    BodyTagSupport
      AbstractBaseTag
        GetData
  
```

## All Implemented Interfaces

`BodyTag`, `IterationTag`, `Serializable`, `Tag`

---

## Field Summary



### Fields from `com.bea.wlw.netui.tags.AbstractBaseTag`

ATTR\_GENERAL, ATTR\_GENERAL\_EXPRESSION, ATTR\_JAVASCRIPT, ATTR\_STYLE, CLASS, ID, JAVASCRIPT\_STATUS, NAME, NETUI\_UNIQUE\_CNT, ONCLICK, ONDBLCLICK, ONKEYDOWN, ONKEYPRESS, ONKEYUP, ONMOUSEDOWN, ONMOUSEMOVE, ONMOUSEOUT, ONMOUSEOVER, ONMOUSEUP, STYLE, TABINDEX

### Fields from `javax.servlet.jsp.tagext.BodyTagSupport`

bodyContent

### Fields from `javax.servlet.jsp.tagext.TagSupport`

id, pageContext

### Fields from interface `javax.servlet.jsp.tagext.BodyTag`

EVAL\_BODY\_BUFFERED, EVAL\_BODY\_TAG

### Fields from interface `javax.servlet.jsp.tagext.IterationTag`

EVAL\_BODY\_AGAIN

### Fields from interface `javax.servlet.jsp.tagext.Tag`

EVAL\_BODY\_INCLUDE, EVAL\_PAGE, SKIP\_BODY, SKIP\_PAGE

## Constructor Summary

*GetData()*

## Method Summary

```
public
    int doEndTag()
        Evaluate the expression at the value attribute and store the result in the
        PageContext under the attribute key resultId.

public
    int doStartTag()
        Start the tag evaluation.

public
    String getResultId()
        Get the key under which the result of evaluating the expression will be stored
        in the PageContext.

public
    String getTagName()
```

Get the name of this tag.

```
public
String getValue()
```

Get the expression to evaluate.

```
protected
```

```
void localRelease()
```

Reset all of the fields of this tag.

```
public
```

```
void setResultId(String resultId)
```

Set the String key under which the result of evaluating an expression will be stored in the PageContext.

```
public
```

```
void setValue(String value)
```

Set the expression to evaluate.

### Methods from **com.bea.wlw.netui.tags.AbstractBaseTag**

addTagIdMapping, evaluateAttributeToString, filter, filter,  
getJavaScriptUtils, getNearestForm, getScriptReporter, getUserLocale,  
registerTagError, release, renderAttribute, rewriteName, write

### Methods from **javax.servlet.jsp.tagext.BodyTagSupport**

doAfterBody, doInitBody, getBodyContent, getPreviousOut, release,  
setBodyContent

### Methods from **javax.servlet.jsp.tagext.TagSupport**

doAfterBody, doEndTag, doStartTag, findAncestorWithClass, getId,  
getParent, getValue, getValues, release, removeValue, setId,  
setPageContext, setParent, setValue

### Methods from class **java.lang.Object**

clone, equals, finalize, getClass, hashCode, notify, notifyAll,  
toString, wait, wait, wait

### Methods from interface **javax.servlet.jsp.tagext.BodyTag**

doInitBody, setBodyContent

### Methods from interface **javax.servlet.jsp.tagext.IterationTag**

doAfterBody

### Methods from interface **javax.servlet.jsp.tagext.Tag**

doEndTag, doStartTag, getParent, release, setPageContext, setParent

## Constructor Detail

### GetData

```
public GetData()
```

## Method Detail

### doEndTag() Method

```
public int doEndTag()  
    throws JspException
```

Evaluate the expression at the `value` attribute and store the result in the `PageContext` under the attribute key `resultId`. If an existing key in the `PageContext`'s attribute map exists, a warning will be written to the log file. If errors occur during expression evaluation, they will be reported in the JSP page. If the value returned by the expression is null, an attribute named `resultId` will be removed from the `PageContext`'s attribute map.

#### *Overrides*

```
BodyTagSupport.doEndTag( )
```

#### Returns

*EVAL\_PAGE*

#### Exceptions

*JspException*

---

### doStartTag() Method

```
public int doStartTag()
```

Start the tag evaluation. This tag ignores its body content.

#### *Overrides*

```
BodyTagSupport.doStartTag( )
```

#### Returns

*SKIP\_BODY*

---

## getResultId() Method

```
public String getResultId()
```

Get the key under which the result of evaluating the expression will be stored in the PageContext.

### Returns

*the String key*

---

## getTagName() Method

```
public String getTagName()
```

Get the name of this tag. This is used to identify the type of this tag for reporting tag errors.

### Overrides

```
AbstractBaseTag.getTagName()
```

### Returns

*a constant String representing the name of this tag.*

---

## getValue() Method

```
public String getValue()
```

Get the expression to evaluate.

### Returns

*the expression to evaluate*

---

## localRelease() Method

```
protected void localRelease()
```

Reset all of the fields of this tag.

### Overrides

```
AbstractBaseTag.localRelease()
```

---

## setResultId(String) Method

```
public void setResultId(String resultId)
```

Set the String key under which the result of evaluating an expression will be stored in the PageContext.

### Parameters

*resultId*  
the String key

---

### setValue(String) Method

```
public void setValue(String value)
```

Set the expression to evaluate.

### Parameters

*value*  
the expression to evaluate

# com.bea.wlw.netui.tags.html Package

## Interface Summary

**Formattable** Interface that indicates whether or not a tag's output can be formatted with by a `FormatTag`.

**URLParams**

## Class Summary

<b>Anchor</b>	Generates a hyperlink to a specified URI.
<b>Attribute</b>	Add an attribute to the parent tag which is being rendered
<b>Base</b>	Provides the base for every URL on the page.
<b>BindingUpdateErrors</b>	Renders the set of error messages found during the process of resolving data binding expressions ( <code>{pageFlow.firstname}</code> , <code>{request.firstname}</code> , etc.).
<b>Button</b>	Generates a button on the page with the specified attributes.
<b>CheckBox</b>	Generates a checkbox which binds to a form bean property or databound expression.
<b>CheckBoxGroup</b>	Groups a collection of <code>CheckBoxOptions</code> , and handles databinding of their values.
<b>CheckBoxOption</b>	A checkbox whose state is determined by its enclosing <code>CheckBoxGroup</code> .
<b>Content</b>	The <code>Content</code> tag is used to display text, or the result of an expression, to the page.
<b>Error</b>	Renders an error message with a given error key value if that key can be found in the <code>ActionErrors</code> registered in the <code>pageContext</code> at <code>org.apache.struts.action.Action.ERROR_KEY</code> .
<b>Errors</b>	Renders the set of error messages found in the <code>ActionErrors</code> registered in the <code>pageContext</code> at <code>org.apache.struts.action.Action.ERROR_KEY</code> .
<b>Exceptions</b>	Renders formatted exception data, as found in the <code>Request</code> with the key: <code>org.apache.struts.action.Action.EXCEPTION_KEY</code> .
<b>FileUpload</b>	Renders an input tag with <code>type="file"</code> , with the given attributes.
<b>Form</b>	This tag represents an input form, associated with a bean whose properties correspond to the various fields of the form.
<b>FormatDate</b>	A formatter used to format dates.
<b>FormatNumber</b>	A formatter used to format numbers.
<b>FormatString</b>	A formatter used to format strings.
<b>FormatTag</b>	Abstract base class for formatting tags.
<b>GetNetuiTagName</b>	
<b>Hidden</b>	Generates a hidden tag with a given value.
<b>Html</b>	Generates the html element and performs error handling within its body.
<b>Image</b>	Generates an image with the specified attributes.
<b>ImageAnchor</b>	Generates a hyperlink with a clickable image.
<b>ImageButton</b>	Generates an <code>&lt;input type="image"&gt;</code> tag with the specified attributes.
<b>Label</b>	Generates styled text span based on an expression or literal value.
<b>Node</b>	Instantiates a <code>TreeNode</code> object that will get added to the parent tag (either a <code>Tree</code> or another <code>Node</code> ).

## Page Flow API Reference

Parameter	Writes a URL parameter to a URL on its parent tag.
ParameterMap	Writes each in a map of URL parameters to a URL on its parent tag.
RadioButtonGroup	Groups a collection of RadioButtonOptions, and handles databinding of their values.
RadioButtonOption	A radio button whose state is determined by its enclosing RadioButtonGroup.
RewriteName	Allow a tag name, typically either an id or name attribute, to participate in URL rewriting.
RewriteURL	Allow a name, typically either an id or name attribute, to participate in URL rewriting.
ScriptContainer	Acts as a container that will bundle up JavaScript created by other NetUI tags, and output it within a single <script> tag.
Select	Renders a select containing a set of SelectOptions.
SelectOption	An option whose state is determined by its enclosing SelectOption.
TextArea	Renders a databound TextArea with the given attributes.
TextBox	Renders a databound TextBox with the given attributes.
Tree	Netui tag that renders a tree control represented by a set of TreeNode objects.
TreeNode	An individual node of a Tree, and rendered by that Tree instance.

# Anchor Class

*public class **Anchor***

extends AbstractBaseTag

implements IAttributeConsumer, URLParams

Generates a hyperlink to a specified URI. Also adds support for URL re-writing and JavaScript-based form submission. An anchor must have one of five attributes to correctly create the hyperlink:

- **action** – an action invoked by clicking the hyperlink.
- **forward** – a forward to redirect to.
- **href** – a URL to go to
- **linkName** – an internal place in the page to move to
- **page** – The module-relative page URL to which this hyperlink will be rendered.

Attribute Descriptions			
Attribute	Required	Runtime Expression Evaluation	Data Bindable
<b>accessKey</b>	No	No	No
	Define the keyboard navigation key for the element.		
<b>action</b>	No	No	Read Only
	An action invoked by clicking the link.		
<b>formSubmit</b>	No	No	No
	True if clicking this link should submit its enclosing form.		
<b>forward</b>	No	No	No
	The logical forward name from which to retrieve the hyperlink URI.		
<b>href</b>	No	No	Read Only
	A URL destination to go to.		
<b>id</b>	No	No	No
	<b>id is deprecated, use tagId.</b> The id attribute of the generated Anchor. You should use the tagId which will encode the id correctly for containers such as the Portal which needs to rewrite id's to make them unique.		
<b>linkName</b>	No	No	No
	An internal place in the page to move to.		



## Page Flow API Reference

<b>location</b>	No	No	No
	Location within the URI to visit		
<b>page</b>	No	No	No
	<b>page is deprecated</b> The module–relative page URL to which this hyperlink will be rendered.		
<b>tagId</b>	No	No	No
	The id of this anchor. The real id written to the HTML stream may be changed by the container (Portal), but may be looked up with the 'tagId' value through the JavaScript method <code>getNetuiTagName( tag, tagId )</code> method written by the <code>&lt;netui:html&gt;</code> tag.		
<b>target</b>	No	No	No
	The window target		

### Related Topics

`<netui:anchor>` Tag Sample

How Do I: Define an Action That Forwards Users to Another Page?  
Form

### *Hierarchy*

```

Object
  TagSupport
    BodyTagSupport
      AbstractBaseTag
        Anchor
  
```

### *All Implemented Interfaces*

BodyTag, IAttributeConsumer, IterationTag, Serializable, Tag, URLParams

### *Direct Known Subclasses*

ImageAnchor

## Field Summary

```

    public
    static HREF
final String
protected String
    id
String
  
```

**Fields from `com.bea.wlw.netui.tags.AbstractBaseTag`**

ATTR\_GENERAL, ATTR\_GENERAL\_EXPRESSION, ATTR\_JAVASCRIPT, ATTR\_STYLE, CLASS, ID, JAVASCRIPT\_STATUS, NAME, NETUI\_UNIQUE\_CNT, ONCLICK, ONDBLCLICK, ONKEYDOWN, ONKEYPRESS, ONKEYUP, ONMOUSEDOWN, ONMOUSEMOVE, ONMOUSEOUT, ONMOUSEOVER, ONMOUSEUP, STYLE, TABINDEX

**Fields from `javax.servlet.jsp.tagext.BodyTagSupport`**

bodyContent

**Fields from `javax.servlet.jsp.tagext.TagSupport`**

id, pageContext

**Fields from interface `javax.servlet.jsp.tagext.BodyTag`**

EVAL\_BODY\_BUFFERED, EVAL\_BODY\_TAG

**Fields from interface `javax.servlet.jsp.tagext.IterationTag`**

EVAL\_BODY\_AGAIN

**Fields from interface `javax.servlet.jsp.tagext.Tag`**

EVAL\_BODY\_INCLUDE, EVAL\_PAGE, SKIP\_BODY, SKIP\_PAGE

## Constructor

### Summary

*Anchor()*

## Method

### Summary

```
public
    void addParameter(String name, Object value)
        Adds a URL parameter to the generated hyperlink.
```

```
protected
    void createAnchorAttributes(UnsyncStringBuffer results)
        Method which will write additional attributes into the anchor being created by the
        createAnchorBeginTag method.
```

```
protected
    boolean createAnchorBeginTag(UnsyncStringBuffer results, UnsyncStringBuffer
        script, boolean encodeAction)
        This method will take all of the internal variables and create an anchor.
```

```

public doAfterBody()
    int          Save the body content of the Anchor.

public
    int doEndTag()
        Render the hyperlink.

public
    int doStartTag()
        Prepare the hyperlink for rendering

public
String getAccessKey()
    Gets the accessKey attribute value.

public
String getAction()
    Return the action of the Anchor.

public
boolean getFormSubmit()
    Return the formSubmit indicator.

public
String getForward()
    Return the forward name of the Anchor.

public
String getHref()
    Return the href of the Anchor.

public
String getId()
    Return the ID associated with this tag.

public
String getLinkName()
    Return the link name of the Anchor.

public
String getLocation()
    Returns the anchor to be added to the end of the generated hyperlink.

public
String getOnBlur()
    Gets the onBlur javascript event.

public
String getOnClick()
    Gets the onClick javascript event.

public
String getOnDblClick()
    Gets the onDblClick javascript event.

public
String getOnFocus()
    Gets the onFocus javascript event.

public
String getOnKeyDown()
    Gets the onKeyDown javascript event.

```

```

public getOnKeyPress()
String      Gets the onKeyPress javascript event.

public
String getOnKeyUp()
           Gets the onKeyUp javascript event.

public
String getOnMouseDown()
           Gets the onMouseDown javascript event.

public
String getOnMouseMove()
           Gets the onMouseMove javascript event.

public
String getOnMouseOut()
           Gets the onMouseOut javascript event.

public
String getOnMouseOver()
           Gets the onMouseOver javascript event.

public
String getOnMouseUp()
           Gets the onMouseUp javascript event.

public
String getPage()
           Returns the module–relative page URL to which this hyperlink will be rendered.

protected
String getRealFormName()
           This will get the real name of the form.

public
String getStyle()
           Gets the style of the rendered html tag.

public
String getStyleClass()
           Gets the style class of the rendered html tag.

public
String getTabIndex()
           Gets the tabIndex of the rendered html tag.

public
String getTagId()
           Return the ID of the tag.

public
String getTagName()
           Returns the name of the Tag.

public
String getTarget()
           Returns the window target.

protected
void localRelease()
           Release any acquired resources.

```

```

public setAccessKey(String accessKey)
    void          Sets the accessKey attribute value.
public
    void setAction(String action)
        Set the name of the action for the Anchor.
public
    void setAttribute(String name, String value)
        Set an attribute value.
public
    void setFormSubmit(boolean formSubmit)
        Sets the formSubmit indicator.
public
    void setForward(String forward)
        Sets the forward name of the Anchor.
public
    void setHref(String href)
        Sets the href of the Anchor.
public
    void setId(String id)
        Set the ID attribute of the tag.
public
    void setLinkName(String linkName)
        Sets the link name of the Anchor.
public
    void setLocation(String location)
        Sets the anchor to be added to the end of the generated hyperlink.
public
    void setOnBlur(String onblur)
        Sets the onBlur javascript event.
public
    void setOnClick(String onclick)
        Sets the onClick javascript event.
public
    void setOnDbClick(String ondblclick)
        Sets the onDbClick javascript event.
public
    void setOnFocus(String onfocus)
        Sets the onFocus javascript event.
public
    void setOnKeyDown(String onkeydown)
        Sets the onKeyDown javascript event.
public
    void setOnKeyPress(String onkeypress)
        Sets the onKeyPress javascript event.
public
    void setOnKeyUp(String onkeyup)
        Sets the onKeyUp javascript event.

```

```

public setOnMouseDown(String onmousedown)
    void          Sets the onMouseDown javascript event.
public
    void setOnMouseMove(String onmousemove)
        Sets the onMouseMove javascript event.
public
    void setOnMouseOut(String onmouseout)
        Sets the onMouseOut javascript event.
public
    void setOnMouseOver(String onmouseover)
        Sets the onMouseOver javascript event.
public
    void setOnMouseUp(String onmouseup)
        Sets the onMouseUp javascript event.
public
    void setPage(String page)
        Sets the module–relative page URL to which this hyperlink will be rendered.
public
    void setStyle(String style)
        Sets the style of the rendered html tag.
public
    void setStyleClass(String styleClass)
        Sets the style class of the rendered html tag.
public
    void setTabIndex(String tabIndex)
        Sets the tabIndex of the rendered html tag.
public
    void setTagId(String tagId)
        Set the ID of the tag.
public
    void setTarget(String target)
        Sets the window target.

```

### Methods from **com.bea.wlw.netui.tags.AbstractBaseTag**

```

addTagIdMapping, evaluateAttributeToString, filter, filter,
getJavaScriptUtils, getNearestForm, getScriptReporter, getUserLocale,
registerTagError, release, renderAttribute, rewriteName, write

```

### Methods from **javax.servlet.jsp.tagext.BodyTagSupport**

```

doInitBody, getBodyContent, getPreviousOut, release, setBodyContent

```

### Methods from **javax.servlet.jsp.tagext.TagSupport**

```

doAfterBody, doEndTag, doStartTag, findAncestorWithClass, getParent,
getValue, getValues, release, removeValue, setPageContext, setParent,
setValue

```

### Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

### Methods from interface `javax.servlet.jsp.tagext.BodyTag`

`doInitBody`, `setBodyContent`

### Methods from interface `com.bea.wlw.netui.tags.IAttributeConsumer`

`setAttribute`

### Methods from interface `javax.servlet.jsp.tagext.IterationTag`

`doAfterBody`

### Methods from interface `javax.servlet.jsp.tagext.Tag`

`doEndTag`, `doStartTag`, `getParent`, `release`, `setPageContext`, `setParent`

### Methods from interface `com.bea.wlw.netui.tags.html.URLParams`

`addParameter`

## Field Detail

### HREF

```
public static final String HREF
```

---

### id

```
protected String id
```

## Constructor Detail

### Anchor

```
public Anchor()
```

## Method Detail

### addParameter(String, Object) Method

```
public void addParameter(String name,
                        Object value)
    throws JspException
```

Adds a URL parameter to the generated hyperlink.

#### Parameters

*name*

– the name of the parameter to be added.

*value*

– the value of the parameter to be added (a String or String[]).

#### Exceptions

*JspException*

---

### createAnchorAttributes(UnsyncStringBuffer) Method

```
protected void createAnchorAttributes(UnsyncStringBuffer results)
```

Method which will write additional attributes into the anchor being created by the createAnchorBeginTag method. This allows subclasses to override which sets of attributes are attached to the <a> tag.

---

### createAnchorBeginTag(UnsyncStringBuffer, UnsyncStringBuffer, boolean) Method

```
protected boolean createAnchorBeginTag(UnsyncStringBuffer results,
                                       UnsyncStringBuffer script,
                                       boolean encodeAction)
    throws JspException
```

This method will take all of the internal variables and create an anchor.

#### Exceptions

*JspException*

---



## doAfterBody() Method

```
public int doAfterBody()  
    throws JspException
```

Save the body content of the Anchor.

### *Overrides*

```
BodyTagSupport.doAfterBody()
```

### Exceptions

*JspException*  
if a JSP exception has occurred

---

## doEndTag() Method

```
public int doEndTag()  
    throws JspException
```

Render the hyperlink.

### *Overrides*

```
BodyTagSupport.doEndTag()
```

### Exceptions

*JspException*  
if a JSP exception has occurred

---

## doStartTag() Method

```
public int doStartTag()  
    throws JspException
```

Prepare the hyperlink for rendering

### *Overrides*

```
BodyTagSupport.doStartTag()
```

### Exceptions

*JspException*  
if a JSP exception has occurred

---

## **getAccessKey() Method**

```
public String getAccessKey()
```

Gets the accessKey attribute value.

### **Returns**

*the accessKey value.*

---

## **getAction() Method**

```
public String getAction()
```

Return the action of the Anchor.

### **Returns**

*a String representing the action name of the Anchor.*

---

## **getFormSubmit() Method**

```
public boolean getFormSubmit()
```

Return the formSubmit indicator.

### **Returns**

*a String representing whether or not the enclosing Form should be submitted.*

---

## **getForward() Method**

```
public String getForward()
```

Return the forward name of the Anchor.

### **Returns**

*the Anchor's logical forward name*

---

## **getHref() Method**

```
public String getHref()
```

Return the href of the Anchor.

**Returns**

*the hyperlink URI*

---

**getId() Method**

```
public String getId()
```

Return the ID associated with this tag.

**Overrides**

```
TagSupport.getId()
```

**Returns**

*the ID.*

---

**getLinkName() Method**

```
public String getLinkName()
```

Return the link name of the Anchor.

**Returns**

*the linkName*

---

**getLocation() Method**

```
public String getLocation()
```

Returns the anchor to be added to the end of the generated hyperlink.

**Returns**

*the name of the location anchor.*

---

**getOnBlur() Method**

```
public String getOnBlur()
```

Gets the onBlur javascript event.

**Returns**

*the onBlur event.*

---

**getOnClick() Method**

```
public String getOnClick()
```

Gets the onClick javascript event.

**Returns**

*the onClick event.*

---

**getOnDbClick() Method**

```
public String getOnDbClick()
```

Gets the onDbClick javascript event.

**Returns**

*the onDbClick event.*

---

**getOnFocus() Method**

```
public String getOnFocus()
```

Gets the onFocus javascript event.

**Returns**

*the onFocus event.*

---

**getOnKeyDown() Method**

```
public String getOnKeyDown()
```

Gets the onKeyDown javascript event.

**Returns**

*the onKeyDown event.*

---

## getOnKeyPress() Method

```
public String getOnKeyPress()
```

Gets the onKeyPress javascript event.

### Returns

*the onKeyPress event.*

---

## getOnKeyUp() Method

```
public String getOnKeyUp()
```

Gets the onKeyUp javascript event.

### Returns

*the onKeyUp event.*

---

## getOnMouseDown() Method

```
public String getOnMouseDown()
```

Gets the onMouseDown javascript event.

### Returns

*the onMouseDown event.*

---

## getOnMouseMove() Method

```
public String getOnMouseMove()
```

Gets the onMouseMove javascript event.

### Returns

*the onMouseMove event.*

---

## getOnMouseOut() Method

```
public String getOnMouseOut()
```

Gets the onMouseOut javascript event.

**Returns**

*the onMouseOut event.*

---

**getOnMouseOver() Method**

```
public String getOnMouseOver()
```

Gets the onMouseOver javascript event.

**Returns**

*the onMouseOver event.*

---

**getOnMouseUp() Method**

```
public String getOnMouseUp()
```

Gets the onMouseUp javascript event.

**Returns**

*the onMouseUp event.*

---

**getPage() Method**

```
public String getPage()
```

Returns the module–relative page URL to which this hyperlink will be rendered.

**Returns**

*the page URL.*

---

**getRealFormName() Method**

```
protected String getRealFormName()
```

This will get the real name of the form. This is set in the id attribute.

**Returns**

*The String real name of the containing form.*

---

## getStyle() Method

```
public String getStyle()
```

Gets the style of the rendered html tag.

### Returns

*the style.*

---

## getStyleClass() Method

```
public String getStyleClass()
```

Gets the style class of the rendered html tag.

### Returns

*the style class.*

---

## getTabIndex() Method

```
public String getTabIndex()
```

Gets the tabIndex of the rendered html tag.

### Returns

*the tabIndex.*

---

## getTagId() Method

```
public String getTagId()
```

Return the ID of the tag. The id may be rewritten by the container (such as a portal) to make sure it is unique. JavaScript may lookup the actual id of the element by looking it up in the netui\_names table written into the HTML.

### Returns

*the tagId.*

---

## getTagName() Method

```
public String getTagName()
```

Returns the name of the Tag.

**Overrides**

`AbstractBaseTag.getTagName()`

---

**getTarget() Method**

`public String getTarget()`

Returns the window target.

**Returns**

*the window target.*

---

**localRelease() Method**

`protected void localRelease()`

Release any acquired resources.

**Overrides**

`AbstractBaseTag.localRelease()`

---

**setAccessKey(String) Method**

`public void setAccessKey(String accessKey)`

Sets the accessKey attribute value. This should key value of the keyboard navigation key. It is recommended not to use the following values because there are often used by browsers A, C, E, F, G, H, V, left arrow, and right arrow.

**Parameters**

*accessKey*  
– the accessKey value.

---

**setAction(String) Method**

`public void setAction(String action)`

Set the name of the action for the Anchor.



**Parameters***action*

– the name of the action to set for the Anchor.

**setAttribute(String, String) Method**

```
public void setAttribute(String name,
                        String value)
    throws JspException
```

Set an attribute value. The *name* represents the name of the attribute. The *value* represents the value and may contain a netui expression. This method may result in errors being generated. This requires that the tag buffer its body and write attributes in the end tag. For the anchor tag it is not legal to set the *id*, *name*, or *href* attributes with this method.

**Parameters***name*

The name of the attribute. This value may not be null or the empty string.

*value*

The value of the attribute. This may contain a netui expression.

**Exceptions***JspException*A *JspException* may be thrown if there is an error setting the attribute.**setFormSubmit(boolean) Method**

```
public void setFormSubmit(boolean formSubmit)
```

Sets the *formSubmit* indicator.

**Parameters***formSubmit*

– whether or not the enclosing Form should be submitted.

**setForward(String) Method**

```
public void setForward(String forward)
```

Sets the forward name of the Anchor.

## Parameters

*forward*

– the logical forward name from which to retrieve the hyperlink URI.

---

## setHref(String) Method

```
public void setHref(String href)
```

Sets the href of the Anchor.

## Parameters

*href*

– the hyperlink URI for the Anchor.

---

## setId(String) Method

```
public void setId(String id)
```

Set the ID attribute of the tag.

## Overrides

```
TagSupport.setId(String)
```

## Parameters

*id*

– the ID.

---

## setLinkName(String) Method

```
public void setLinkName(String linkName)
```

Sets the link name of the Anchor.

## Parameters

*linkName*

– the link name for the Anchor.

---

## setLocation(String) Method

```
public void setLocation(String location)
```

Sets the anchor to be added to the end of the generated hyperlink.

## Parameters

*location*

– the name of the location anchor.

---

## setOnBlur(String) Method

```
public void setOnBlur(String onblur)
```

Sets the onBlur javascript event.

## Parameters

*onblur*

– the onBlur event.

---

## setOnClick(String) Method

```
public void setOnClick(String onclick)
```

Sets the onClick javascript event.

## Parameters

*onclick*

– the onClick event.

---

## setOnDbClick(String) Method

```
public void setOnDbClick(String ondblclick)
```

Sets the onDbClick javascript event.

## Parameters

*ondblclick*

– the onDbClick event.

---

## setOnFocus(String) Method

```
public void setOnFocus(String onfocus)
```

Sets the onFocus javascript event.

## Parameters

*onfocus*

– the onFocus event.

---

## setOnKeyDown(String) Method

```
public void setOnKeyDown(String onkeydown)
```

Sets the onKeyDown javascript event.

## Parameters

*onkeydown*

– the onKeyDown event.

---

## setOnKeyPress(String) Method

```
public void setOnKeyPress(String onkeypress)
```

Sets the onKeyPress javascript event.

## Parameters

*onkeypress*

– the onKeyPress event.

---

## setOnKeyUp(String) Method

```
public void setOnKeyUp(String onkeyup)
```

Sets the onKeyUp javascript event.

## Parameters

*onkeyup*

– the onKeyUp event.

---

## setOnMouseDown(String) Method

```
public void setOnMouseDown(String onmousedown)
```

Sets the onMouseDown javascript event.

**Parameters**

*onmousedown*

– the onMouseDown event.

---

**setOnMouseMove(String) Method**

```
public void setOnMouseMove(String onmousemove)
```

Sets the onMouseMove javascript event.

**Parameters**

*onmousemove*

– the onMouseMove event.

---

**setOnMouseOut(String) Method**

```
public void setOnMouseOut(String onmouseout)
```

Sets the onMouseOut javascript event.

**Parameters**

*onmouseout*

– the onMouseOut event.

---

**setOnMouseOver(String) Method**

```
public void setOnMouseOver(String onmouseover)
```

Sets the onMouseOver javascript event.

**Parameters**

*onmouseover*

– the onMouseOver event.

---

**setOnMouseUp(String) Method**

```
public void setOnMouseUp(String onmouseup)
```

Sets the onMouseUp javascript event.

## Parameters

*onmouseup*

– the onMouseUp event.

---

## setPage(String) Method

```
public void setPage(String page)
```

Sets the module–relative page URL to which this hyperlink will be rendered.

## Parameters

*page*

– the page URL.

---

## setStyle(String) Method

```
public void setStyle(String style)
```

Sets the style of the rendered html tag.

## Parameters

*style*

– the html style.

---

## setStyleClass(String) Method

```
public void setStyleClass(String styleClass)
```

Sets the style class of the rendered html tag.

## Parameters

*styleClass*

– the html style class.

---

## setTabIndex(String) Method

```
public void setTabIndex(String tabIndex)
```

Sets the tabIndex of the rendered html tag.

## Parameters

*tabindex*

– the tab index.

---

## setTagId(String) Method

```
public void setTagId(String tagId)
```

Set the ID of the tag.

## Parameters

*tagId*

– the tagId.

---

## setTarget(String) Method

```
public void setTarget(String target)
```

Sets the window target.

## Parameters

*target*

– the window target.

---

# Attribute Class

*public class Attribute*

extends AbstractBaseTag

Add an attribute to the parent tag which is being rendered

## Related Topics

Parameter

---

### *Hierarchy*

```
Object
  TagSupport
    BodyTagSupport
      AbstractBaseTag
        Attribute
```

### *All Implemented Interfaces*

BodyTag, IterationTag, Serializable, Tag

---

## Field Summary

### Fields from com.bea.wlw.netui.tags.AbstractBaseTag

ATTR\_GENERAL, ATTR\_GENERAL\_EXPRESSION, ATTR\_JAVASCRIPT, ATTR\_STYLE, CLASS, ID, JAVASCRIPT\_STATUS, NAME, NETUI\_UNIQUE\_CNT, ONCLICK, ONDBLCLICK, ONKEYDOWN, ONKEYPRESS, ONKEYUP, ONMOUSEDOWN, ONMOUSEMOVE, ONMOUSEOUT, ONMOUSEOVER, ONMOUSEUP, STYLE, TABINDEX

### Fields from javax.servlet.jsp.tagext.BodyTagSupport

bodyContent

### Fields from javax.servlet.jsp.tagext.TagSupport

id, pageContext

### Fields from interface javax.servlet.jsp.tagext.BodyTag

EVAL\_BODY\_BUFFERED, EVAL\_BODY\_TAG



**Fields from interface javax.servlet.jsp.tagext.IterationTag**

EVAL\_BODY\_AGAIN

**Fields from interface javax.servlet.jsp.tagext.Tag**

EVAL\_BODY\_INCLUDE, EVAL\_PAGE, SKIP\_BODY, SKIP\_PAGE

## Constructor Summary

*Attribute()*

## Method Summary

```

    public
        int doStartTag()
            Add the URL parameter to the Parameter's parent.

    public
        String getName()
            Return the value of the name attribute.

    public
        String getTagName()
            Return the name of the Tag.

    public
        String getValue()
            Return the value of the value attribute

protected
    void localRelease()
        This is a method that will initialize all temp state on a tag and should be called in the
        doEndTag method.

    public
        void setName(String name)
            Sets the name attribute.

    public
        void setValue(String value)
            Sets the value attribute.

```

**Methods from com.bea.wlw.netui.tags.AbstractBaseTag**

```

addTagIdMapping, evaluateAttributeToString, filter, filter,
getJavaScriptUtils, getNearestForm, getScriptReporter, getUserLocale,
registerTagError, release, renderAttribute, rewriteName, write

```

**Methods from javax.servlet.jsp.tagext.BodyTagSupport**

`doAfterBody`, `doEndTag`, `doInitBody`, `getBodyContent`, `getPreviousOut`, `release`, `setBodyContent`

### Methods from `javax.servlet.jsp.tagext.TagSupport`

`doAfterBody`, `doEndTag`, `doStartTag`, `findAncestorWithClass`, `getId`, `getParent`, `getValue`, `getValues`, `release`, `removeValue`, `setId`, `setPageContext`, `setParent`, `setValue`

### Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

### Methods from interface `javax.servlet.jsp.tagext.BodyTag`

`doInitBody`, `setBodyContent`

### Methods from interface `javax.servlet.jsp.tagext.IterationTag`

`doAfterBody`

### Methods from interface `javax.servlet.jsp.tagext.Tag`

`doEndTag`, `doStartTag`, `getParent`, `release`, `setPageContext`, `setParent`

## Constructor Detail

### Attribute

```
public Attribute()
```

## Method Detail

### `doStartTag()` Method

```
public int doStartTag()  
    throws JspException
```

Add the URL parameter to the Parameter's parent.

#### *Overrides*

```
BodyTagSupport.doStartTag()
```

## Exceptions

*JspException*

if a JSP exception has occurred

---

## getName() Method

```
public String getName()
```

Return the value of the name attribute.

### Returns

*the value of the attribute name.*

---

## getTagName() Method

```
public String getTagName()
```

Return the name of the Tag.

### Overrides

```
AbstractBaseTag.getTagName()
```

---

## getValue() Method

```
public String getValue()
```

Return the value of the value attribute

### Returns

*the value of the attribute value.*

---

## localRelease() Method

```
protected void localRelease()
```

*Description copied from* `AbstractBaseTag.localRelease()`

This is a method that will initialize all temp state on a tag and should be called in the doEndTag method.

### Overrides

```
AbstractBaseTag.localRelease()
```

---

## **setName(String) Method**

```
public void setName(String name)
```

Sets the name attribute.

### **Parameters**

*name*

– the value of the name attribute.

---

## **setValue(String) Method**

```
public void setValue(String value)
```

Sets the value attribute.

### **Parameters**

*value*

– the value of the name attribute.

# Base Class

*public class* **Base**

extends AbstractBaseTag  
implements IAttributeConsumer

Provides the base for every URL on the page.

Attribute Descriptions			
Attribute	Required	Runtime Expression Evaluation	Data Bindable
target	No	No	No
	The default target window.		

*Hierarchy*

Object  
TagSupport  
BodyTagSupport  
AbstractBaseTag  
Base

*All Implemented Interfaces*

BodyTag, IAttributeConsumer, IterationTag, Serializable, Tag

## Field Summary

public  
static

**HREF**

final String

String

protected

String

**target**

The default target window for every link that does not have an explicit target reference

public  
static

**TARGET**

final String

String

**Fields from com.bea.wlw.netui.tags.AbstractBaseTag**

ATTR\_GENERAL, ATTR\_GENERAL\_EXPRESSION, ATTR\_JAVASCRIPT, ATTR\_STYLE, CLASS, ID, JAVASCRIPT\_STATUS, NAME, NETUI\_UNIQUE\_CNT, ONCLICK,

ONDBLCLICK, ONKEYDOWN, ONKEYPRESS, ONKEYUP, ONMOUSEDOWN, ONMOUSEMOVE, ONMOUSEOUT, ONMOUSEOVER, ONMOUSEUP, STYLE, TABINDEX

### Fields from `javax.servlet.jsp.tagext.BodyTagSupport`

`bodyContent`

### Fields from `javax.servlet.jsp.tagext.TagSupport`

`id`, `pageContext`

### Fields from interface `javax.servlet.jsp.tagext.BodyTag`

`EVAL_BODY_BUFFERED`, `EVAL_BODY_TAG`

### Fields from interface `javax.servlet.jsp.tagext.IterationTag`

`EVAL_BODY_AGAIN`

### Fields from interface `javax.servlet.jsp.tagext.Tag`

`EVAL_BODY_INCLUDE`, `EVAL_PAGE`, `SKIP_BODY`, `SKIP_PAGE`

## Constructor Summary

*Base()*

## Method Summary

```
public
    int doAfterBody()
public
    int doEndTag()
        Render the HTML base tag.
public
    int doStartTag()
public
    String getTagName()
        Returns the name of the Tag.
public
    String getTarget()
        Return the target of the base.
protected
    void localRelease()
```

## Page Flow API Reference

This is a method that will initialize all temp state on a tag and should be called in the `doEndTag` method.

```
public
    void setAttribute(String name, String value)
        Set an attribute value.

public
    void setTarget(String target)
        Set the default window target.
```

### Methods from `com.bea.wlw.netui.tags.AbstractBaseTag`

`addTagIdMapping`, `evaluateAttributeToString`, `filter`, `filter`,  
`getJavaScriptUtils`, `getNearestForm`, `getScriptReporter`, `getUserLocale`,  
`registerTagError`, `release`, `renderAttribute`, `rewriteName`, `write`

### Methods from `javax.servlet.jsp.tagext.BodyTagSupport`

`doInitBody`, `getBodyContent`, `getPreviousOut`, `release`, `setBodyContent`

### Methods from `javax.servlet.jsp.tagext.TagSupport`

`doAfterBody`, `doEndTag`, `doStartTag`, `findAncestorWithClass`, `getId`,  
`getParent`, `getValue`, `getValues`, `release`, `removeValue`, `setId`,  
`setPageContext`, `setParent`, `setValue`

### Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`,  
`toString`, `wait`, `wait`, `wait`

### Methods from interface `javax.servlet.jsp.tagext.BodyTag`

`doInitBody`, `setBodyContent`

### Methods from interface `com.bea.wlw.netui.tags.IAttributeConsumer`

`setAttribute`

### Methods from interface `javax.servlet.jsp.tagext.IterationTag`

`doAfterBody`

### Methods from interface `javax.servlet.jsp.tagext.Tag`

`doEndTag`, `doStartTag`, `getParent`, `release`, `setPageContext`, `setParent`

## Field Detail

## HREF

```
public static final String HREF
```

---

## target

```
protected String target
```

The default target window for every link that does not have an explicit target reference

---

## TARGET

```
public static final String TARGET
```

## Constructor Detail

### Base

```
public Base()
```

## Method Detail

### doAfterBody() Method

```
public int doAfterBody()
    throws JspException
```

#### *Overrides*

```
BodyTagSupport.doAfterBody()
```

### Exceptions

*JspException*  
if a JSP exception has occurred

---

### doEndTag() Method

```
public int doEndTag()
    throws JspException
```

Render the HTML base tag.



**Overrides**

`BodyTagSupport.doEndTag()`

**Exceptions**

*JspException*

if a JSP exception has occurred

---

**doStartTag() Method**

```
public int doStartTag()
    throws JspException
```

**Overrides**

`BodyTagSupport.doStartTag()`

**Exceptions**

*JspException*

if a JSP exception has occurred

---

**getTagName() Method**

```
public String getTagName()
```

Returns the name of the Tag.

**Overrides**

`AbstractBaseTag.getTagName()`

---

**getTarget() Method**

```
public String getTarget()
```

Return the target of the base.

**Returns**

*the target.*

---

**localRelease() Method**

```
protected void localRelease()
```

*Description copied from AbstractBaseTag.localRelease()*

This is a method that will initialize all temp state on a tag and should be called in the doEndTag method.

**Overrides**

`AbstractBaseTag.localRelease()`

---

**setAttribute(String, String) Method**

```
public void setAttribute(String name,  
                        String value)  
    throws JspException
```

Set an attribute value. The `name` represents the name of the attribute. The `value` represents the value and may contain a netui expression. This method may result in errors being generated. This requires that the tag buffer its body and write attributes in the end tag. For the base tag it is not legal to set the `id`, `name`, or `href` attributes with this method.

**Parameters**

*name*

The name of the attribute. This value may not be null or the empty string.

*value*

The value of the attribute. This may contain a netui expression.

**Exceptions**

*JspException*

A `JspException` may be thrown if there is an error setting the attribute.

---

**setTarget(String) Method**

```
public void setTarget(String target)
```

Set the default window target.

**Parameters**

*target*

– the window target.

# BindingUpdateErrors Class

*public class* **BindingUpdateErrors**

extends AbstractBaseTag

Renders the set of error messages found during the process of resolving data binding expressions ({pageFlow.firstname}, {request.firstname}, etc.).

These errors are treated as warnings. By default, this tag is only active in Development mode and the warning are not displayed in Production mode. The tag is intended for development use.

Attribute Descriptions			
Attribute	Required	Runtime Expression Evaluation	Data Bindable
<b>alwaysReport</b>	No	No	No
	This attribute will override the default behavior of only showing errors in iterative development mode, so they are also shown in production mode.		
<b>expression</b>	No	No	No
	This attribute will display errors only for the specified expression.		

## Related Topics

<netui:bindingUpdateErrors> Tag Sample

---

## Hierarchy

```

Object
  TagSupport
    BodyTagSupport
      AbstractBaseTag
        BindingUpdateErrors
  
```

## All Implemented Interfaces

BodyTag, IterationTag, Serializable, Tag

---

## Field Summary

**Fields from `com.bea.wlw.netui.tags.AbstractBaseTag`**

ATTR\_GENERAL, ATTR\_GENERAL\_EXPRESSION, ATTR\_JAVASCRIPT, ATTR\_STYLE, CLASS, ID, JAVASCRIPT\_STATUS, NAME, NETUI\_UNIQUE\_CNT, ONCLICK, ONDBLCLICK, ONKEYDOWN, ONKEYPRESS, ONKEYUP, ONMOUSEDOWN, ONMOUSEMOVE, ONMOUSEOUT, ONMOUSEOVER, ONMOUSEUP, STYLE, TABINDEX

**Fields from `javax.servlet.jsp.tagext.BodyTagSupport`**

bodyContent

**Fields from `javax.servlet.jsp.tagext.TagSupport`**

id, pageContext

**Fields from interface `javax.servlet.jsp.tagext.BodyTag`**

EVAL\_BODY\_BUFFERED, EVAL\_BODY\_TAG

**Fields from interface `javax.servlet.jsp.tagext.IterationTag`**

EVAL\_BODY\_AGAIN

**Fields from interface `javax.servlet.jsp.tagext.Tag`**

EVAL\_BODY\_INCLUDE, EVAL\_PAGE, SKIP\_BODY, SKIP\_PAGE

## Constructor

### Summary

*BindingUpdateErrors()*

## Method

### Summary

```
public
    int doStartTag()
        Render the specified error messages if there are any.

public
    String getExpression()
        Returns the expression to match for binding errors.

public
    String getTagName()
        Return the name of the Tag.

public
    boolean isAlwaysReport()
        Returns true if errors are reported in production mode.
```

```

protected localRelease()
    void          Release any acquired resources.
public
    void setAlwaysReport(boolean alwaysReport)
        Set the value which will override the default behavior of not showing errors in
        production mode.
public
    void setExpression(String expression)
        Sets the expression to match for binding errors.

```

#### Methods from `com.bea.wlw.netui.tags.AbstractBaseTag`

```

addTagIdMapping, evaluateAttributeToString, filter, filter,
getJavaScriptUtils, getNearestForm, getScriptReporter, getUserLocale,
registerTagError, release, renderAttribute, rewriteName, write

```

#### Methods from `javax.servlet.jsp.tagext.BodyTagSupport`

```

doAfterBody, doEndTag, doInitBody, getBodyContent, getPreviousOut,
release, setBodyContent

```

#### Methods from `javax.servlet.jsp.tagext.TagSupport`

```

doAfterBody, doEndTag, doStartTag, findAncestorWithClass, getId,
getParent, getValue, getValues, release, removeValue, setId,
setPageContext, setParent, setValue

```

#### Methods from class `java.lang.Object`

```

clone, equals, finalize, getClass, hashCode, notify, notifyAll,
toString, wait, wait, wait

```

#### Methods from interface `javax.servlet.jsp.tagext.BodyTag`

```

doInitBody, setBodyContent

```

#### Methods from interface `javax.servlet.jsp.tagext.IterationTag`

```

doAfterBody

```

#### Methods from interface `javax.servlet.jsp.tagext.Tag`

```

doEndTag, doStartTag, getParent, release, setPageContext, setParent

```

## Constructor Detail

### BindingUpdateErrors

```

public BindingUpdateErrors()

```

## Method Detail

### doStartTag() Method

```
public int doStartTag()
    throws JspException
```

Render the specified error messages if there are any.

#### *Overrides*

```
BodyTagSupport.doStartTag( )
```

### Exceptions

*JspException*  
if a JSP exception has occurred

---

### getExpression() Method

```
public String getExpression()
```

Returns the expression to match for binding errors. If an expression is set, only binding errors for that expression will be displayed. Otherwise, all errors will be displayed.

#### **Returns**

*the expression being match*

---

### getTagName() Method

```
public String getTagName()
```

Return the name of the Tag.

#### *Overrides*

```
AbstractBaseTag.getTagName( )
```

---

### isAlwaysReport() Method

```
public boolean isAlwaysReport()
```

Returns `true` if errors are reported in production mode. The default behavior is to only display errors in interactive development.

## Returns

*the boolean value indicating if errors are reported in production mode.*

---

## localRelease() Method

```
protected void localRelease()
```

Release any acquired resources.

### Overrides

```
AbstractBaseTag.localRelease()
```

---

## setAlwaysReport(boolean) Method

```
public void setAlwaysReport(boolean alwaysReport)
```

Set the value which will override the default behavior of not showing errors in production mode.

### Parameters

*alwaysReport*

a boolean that if `true` will cause the errors to always be displayed. The default is `false`

---

## setExpression(String) Method

```
public void setExpression(String expression)
```

Sets the expression to match for binding errors. If an expression is set, only binding errors for that expression will be displayed. Otherwise, all errors will be displayed.

### Parameters

*expression*

The expression to match against.

# Button Class

*public class* **Button**

extends AbstractBaseTag  
implements IAttributeConsumer, URLParams

Generates a button on the page with the specified attributes. The <netui:button> tag must be enclosed in <netui:form...> ... </netui:form> tags. You can specify the action that the form will raise on the <netui:form> tag.

Attribute Descriptions			
Attribute	Required	Runtime Expression Evaluation	Data Bindable
<b>accessKey</b>	No	No	No
	Define the keyboard navigation key for the element.		
<b>action</b>	No	No	No
	The overriding action caused by pressing this button.		
<b>alt</b>	No	No	Read Only
	Define the alt attribute of the <Input> element.		
<b>disabled</b>	No	No	Read Only
	A boolean value which if true will cause the Button to be disabled.		
<b>tagId</b>	No	No	No
	The ID of this button used by the enclosing html tag to get the tag's real ID attribute for javascript and focus purposes. The real ID attribute id is generated based upon this name.		
<b>type</b>	No	No	No
	The type of button this is (defaults to 'submit').		
<b>value</b>	No	No	Read Only
	The label of this button.		

## Related Topics

<netui:form> Tag Sample  
Form

---

## Hierarchy



```

Object
TagSupport
BodyTagSupport
AbstractBaseTag
Button

```

### *All Implemented Interfaces*

BodyTag, IAttributeConsumer, IterationTag, Serializable, Tag, URLParams

---

## Field Summary

```

protected String
    action
        The action of this button.

    public
    static ACTION_OVERRIDE
final String
    String

protected String
    buttonValue
        String

protected String
    text
        The body content of this tag (if any).

protected String
    type
        The type of button this is (submit, button, or reset).

    public
    static TYPE
final String
    String

protected String
    value
        The text of the button (this will override any body text).

    public
    static VALUE
final String
    String

```

### Fields from `com.bea.wlw.netui.tags.AbstractBaseTag`

ATTR\_GENERAL, ATTR\_GENERAL\_EXPRESSION, ATTR\_JAVASCRIPT, ATTR\_STYLE, CLASS, ID, JAVASCRIPT\_STATUS, NAME, NETUI\_UNIQUE\_CNT, ONCLICK, ONDBLCLICK, ONKEYDOWN, ONKEYPRESS, ONKEYUP, ONMOUSEDOWN, ONMOUSEMOVE, ONMOUSEOUT, ONMOUSEOVER, ONMOUSEUP, STYLE, TABINDEX

### Fields from `javax.servlet.jsp.tagext.BodyTagSupport`

bodyContent

**Fields from `javax.servlet.jsp.tagext.TagSupport`**`id`, `pageContext`**Fields from interface `javax.servlet.jsp.tagext.BodyTag`**`EVAL_BODY_BUFFERED`, `EVAL_BODY_TAG`**Fields from interface `javax.servlet.jsp.tagext.IterationTag`**`EVAL_BODY_AGAIN`**Fields from interface `javax.servlet.jsp.tagext.Tag`**`EVAL_BODY_INCLUDE`, `EVAL_PAGE`, `SKIP_BODY`, `SKIP_PAGE`

## Constructor

### Summary

***Button()***

## Method

### Summary

```

public
    void addParameter(String name, Object value)
        Adds a URL parameter to the generated hyperlink.

public
    int doAfterBody()
        Save the associated button label from the body content (if any).

public
    int doEndTag()
        Render the button.

public
    int doStartTag()
        Process the start of the Button.

public
    String getAccessKey()
        Gets the accessKey attribute value.

public
    String getAction()
        Return the action of the Button.

public
    String getAlt()
        Gets the alt attribute on the generate <input tag.

public

```

String **getDisabled()**  
 Returns the boolean value or expression indicating the disable state of the button.

public  
 String **getOnBlur()**  
 Gets the onBlur javascript event.

public  
 String **getOnChange()**  
 Gets the onChange javascript event.

public  
 String **getOnClick()**  
 Gets the onClick javascript event.

public  
 String **getOnDbClick()**  
 Gets the onDbClick javascript event.

public  
 String **getOnFocus()**  
 Gets the onFocus javascript event.

public  
 String **getOnKeyDown()**  
 Gets the onKeyDown javascript event.

public  
 String **getOnKeyPress()**  
 Gets the onKeyPress javascript event.

public  
 String **getOnKeyUp()**  
 Gets the onKeyUp javascript event.

public  
 String **getOnMouseDown()**  
 Gets the onMouseDown javascript event.

public  
 String **getOnMouseMove()**  
 Gets the onMouseMove javascript event.

public  
 String **getOnMouseOut()**  
 Gets the onMouseOut javascript event.

public  
 String **getOnMouseOver()**  
 Gets the onMouseOver javascript event.

public  
 String **getOnMouseUp()**  
 Gets the onMouseUp javascript event.

public  
 String **getOnSelect()**  
 Gets the onSelect javascript event.

public  
 String **getStyle()**

Gets the style of the rendered html tag.

```
public
String getStyleClass()
```

Gets the style class of the rendered html tag.

```
public
String getTabIndex()
```

Gets the tabIndex of the rendered html tag.

```
public
String getTagId()
```

Return the ID of the button.

```
public
String getTagName()
```

Return the name of the Tag.

```
public
String getType()
```

Return the type of the Button.

```
public
String getValue()
```

Returns the value of the Button's text.

```
protected
void localRelease()
```

Release any acquired resources.

```
public
void setAccessKey(String accessKey)
```

Sets the accessKey attribute value.

```
public
void setAction(String action)
```

Set the name of the action for the Button.

```
public
void setAlt(String alt)
```

Sets the alt attribute value.

```
public
void setAttribute(String name, String value)
```

Set an attribute value.

```
public
void setDisabled(String disabled)
```

Set the disable state either with the literal "true" or "false" or with an expression.

```
public
void setOnBlur(String onblur)
```

Sets the onBlur javascript event.

```
public
void setOnChange(String onchange)
```

Sets the onChange javascript event.

```
public
void setOnClick(String onclick)
```

Sets the onClick javascript event.

```

public setOnDbClick(String ondblclick)
    void          Sets the onDbClick javascript event.
public
    void setOnFocus(String onblur)
        Sets the onFocus javascript event.
public
    void setOnKeyDown(String onkeydown)
        Sets the onKeyDown javascript event.
public
    void setOnKeyPress(String onkeypress)
        Sets the onKeyPress javascript event.
public
    void setOnKeyUp(String onkeyup)
        Sets the onKeyUp javascript event.
public
    void setOnMouseDown(String onmousedown)
        Sets the onMouseDown javascript event.
public
    void setOnMouseMove(String onmousemove)
        Sets the onMouseMove javascript event.
public
    void setOnMouseOut(String onmouseout)
        Sets the onMouseOut javascript event.
public
    void setOnMouseOver(String onmouseover)
        Sets the onMouseOver javascript event.
public
    void setOnMouseUp(String onmouseup)
        Sets the onMouseUp javascript event.
public
    void setOnSelect(String onselect)
        Sets the onSelect javascript event.
public
    void setStyle(String style)
        Sets the style of the rendered html tag.
public
    void setStyleClass(String styleClass)
        Sets the style class of the rendered html tag.
public
    void setTabIndex(String tabindex)
        Sets the tabIndex of the rendered html tag.
public
    void setTagId(String tagId)
        Set the ID of the button.
public
    void setType(String type)
        Set the type of the Button (submit, button, or reset).

```

```
public setValue(String value)
void          Set the value of the Button's text.
```

**Methods from `com.bea.wlw.netui.tags.AbstractBaseTag`**

```
addTagIdMapping, evaluateAttributeToString, filter, filter,
getJavaScriptUtils, getNearestForm, getScriptReporter, getUserLocale,
registerTagError, release, renderAttribute, rewriteName, write
```

**Methods from `javax.servlet.jsp.tagext.BodyTagSupport`**

```
doInitBody, getBodyContent, getPreviousOut, release, setBodyContent
```

**Methods from `javax.servlet.jsp.tagext.TagSupport`**

```
doAfterBody, doEndTag, doStartTag, findAncestorWithClass, getId,
getParent, getValue, getValues, release, removeValue, setId,
setPageContext, setParent, setValue
```

**Methods from class `java.lang.Object`**

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll,
toString, wait, wait, wait
```

**Methods from interface `javax.servlet.jsp.tagext.BodyTag`**

```
doInitBody, setBodyContent
```

**Methods from interface `com.bea.wlw.netui.tags.IAttributeConsumer`**

```
setAttribute
```

**Methods from interface `javax.servlet.jsp.tagext.IterationTag`**

```
doAfterBody
```

**Methods from interface `javax.servlet.jsp.tagext.Tag`**

```
doEndTag, doStartTag, getParent, release, setPageContext, setParent
```

**Methods from interface `com.bea.wlw.netui.tags.html.URLParams`**

```
addParameter
```

## Field Detail

### action

```
protected String action
```

The action of this button. If an action is specified for this button, it will override the action of the enclosing `Form` tag.

---

### **ACTION\_OVERRIDE**

```
public static final String ACTION_OVERRIDE
```

---

### **buttonValue**

```
protected String buttonValue
```

---

### **text**

```
protected String text
```

The body content of this tag (if any).

---

### **type**

```
protected String type
```

The type of button this is (submit, button, or reset).

---

### **TYPE**

```
public static final String TYPE
```

---

### **value**

```
protected String value
```

The text of the button (this will override any body text).

---

### **VALUE**

```
public static final String VALUE
```

## Constructor Detail

### Button

```
public Button()
```

## Method Detail

### addParameter(String, Object) Method

```
public void addParameter(String name,
                          Object value)
    throws JspException
```

Adds a URL parameter to the generated hyperlink.

#### Parameters

*name*

– the name of the parameter to be added.

*value*

– the value of the parameter to be added (a String or String[]).

#### Exceptions

*JspException*

---

### doAfterBody() Method

```
public int doAfterBody()
    throws JspException
```

Save the associated button label from the body content (if any).

#### Overrides

```
BodyTagSupport.doAfterBody()
```

#### Exceptions

*JspException*

if a JSP exception has occurred

---



## doEndTag() Method

```
public int doEndTag()  
    throws JspException
```

Render the button.

### *Overrides*

```
BodyTagSupport.doEndTag( )
```

### Exceptions

*JspException*  
if a JSP exception has occurred

---

## doStartTag() Method

```
public int doStartTag()  
    throws JspException
```

Process the start of the Button.

### *Overrides*

```
BodyTagSupport.doStartTag( )
```

### Exceptions

*JspException*  
if a JSP exception has occurred

---

## getAccessKey() Method

```
public String getAccessKey()
```

Gets the accessKey attribute value.

### Returns

*the accessKey value.*

---

## getAction() Method

```
public String getAction()
```

Return the action of the Button.

## Returns

*a String representing the action name of the Button.*

---

## getAlt() Method

```
public String getAlt()
```

Gets the alt attribute on the generate <input tag.

## Returns

*the alt value.*

---

## getDisabled() Method

```
public String getDisabled()
```

Returns the boolean value or expression indicating the disable state of the button.

## Returns

*the disabled state (true or false) or an expression*

---

## getOnBlur() Method

```
public String getOnBlur()
```

Gets the onBlur javascript event.

## Returns

*the onBlur event.*

---

## getOnChange() Method

```
public String getOnChange()
```

Gets the onChange javascript event.

## Returns

*the onChange event.*

---

## getOnClick() Method

```
public String getOnClick()
```

Gets the onClick javascript event.

### Returns

*the onClick event.*

---

## getOnDbClick() Method

```
public String getOnDbClick()
```

Gets the onDbClick javascript event.

### Returns

*the onDbClick event.*

---

## getOnFocus() Method

```
public String getOnFocus()
```

Gets the onFocus javascript event.

### Returns

*the onFocus event.*

---

## getOnKeyDown() Method

```
public String getOnKeyDown()
```

Gets the onKeyDown javascript event.

### Returns

*the onKeyDown event.*

---

## getOnKeyPress() Method

```
public String getOnKeyPress()
```

Gets the onKeyPress javascript event.

## Returns

*the onKeyPress event.*

---

## getOnKeyUp() Method

```
public String getOnKeyUp()
```

Gets the onKeyUp javascript event.

## Returns

*the onKeyUp event.*

---

## getOnMouseDown() Method

```
public String getOnMouseDown()
```

Gets the onMouseDown javascript event.

## Returns

*the onMouseDown event.*

---

## getOnMouseMove() Method

```
public String getOnMouseMove()
```

Gets the onMouseMove javascript event.

## Returns

*the onMouseMove event.*

---

## getOnMouseOut() Method

```
public String getOnMouseOut()
```

Gets the onMouseOut javascript event.

## Returns

*the onMouseOut event.*

---

## getOnMouseOver() Method

```
public String getOnMouseOver()
```

Gets the onMouseOver javascript event.

### Returns

*the onMouseOver event.*

---

## getOnMouseUp() Method

```
public String getOnMouseUp()
```

Gets the onMouseUp javascript event.

### Returns

*the onMouseUp event.*

---

## getOnSelect() Method

```
public String getOnSelect()
```

Gets the onSelect javascript event.

### Returns

*the onSelect event.*

---

## getStyle() Method

```
public String getStyle()
```

Gets the style of the rendered html tag.

### Returns

*the style.*

---

## getStyleClass() Method

```
public String getStyleClass()
```

Gets the style class of the rendered html tag.

**Returns**

*the style class.*

---

**getTabIndex() Method**

```
public String getTabIndex()
```

Gets the tabIndex of the rendered html tag.

**Returns**

*the tabindex.*

---

**getTagId() Method**

```
public String getTagId()
```

Return the ID of the button.

**Returns**

*the ID.*

---

**getTagName() Method**

```
public String getTagName()
```

Return the name of the Tag.

**Overrides**

```
AbstractBaseTag.getTagName()
```

---

**getType() Method**

```
public String getType()
```

Return the type of the Button.

**Returns**

*the Button type.*

---

## getValue() Method

```
public String getValue()
```

Returns the value of the Button's text.

### Returns

*the Button value.*

---

## localRelease() Method

```
protected void localRelease()
```

Release any acquired resources.

### Overrides

```
AbstractBaseTag.localRelease()
```

---

## setAccessKey(String) Method

```
public void setAccessKey(String accessKey)
```

Sets the accessKey attribute value. This should key value of the keyboard navigation key. It is recommended not to use the following values because there are often used by browsers A, C, E, F, G, H, V, left arrow, and right arrow.

### Parameters

*accessKey*  
– the accessKey value.

---

## setAction(String) Method

```
public void setAction(String action)
```

Set the name of the action for the Button.

### Parameters

*action*  
– the name of the action to set for the Button.

---

## setAlt(String) Method

```
public void setAlt(String alt)
```

Sets the alt attribute value.

### Parameters

*alt*

– the alt value.

---

## setAttribute(String, String) Method

```
public void setAttribute(String name,  
                        String value)  
    throws JspException
```

Set an attribute value. The `name` represents the name of the attribute. The `value` represents the value and may contain a netui expression. This method may result in errors being generated. This requires that the tag buffer its body and write attributes in the end tag. For the button tag it is not legal to set the `id`, `name`, `type`, or `value` attributes with this method.

### Parameters

*name*

The name of the attribute. This value may not be null or the empty string.

*value*

The value of the attribute. This may contain a netui expression.

### Exceptions

*JspException*

A `JspException` may be thrown if there is an error setting the attribute.

---

## setDisabled(String) Method

```
public void setDisabled(String disabled)
```

Set the disable state either with the literal "true" or "false" or with an expression.

### Parameters

*disabled*

– true or false or an expression

---



## setOnBlur(String) Method

```
public void setOnBlur(String onblur)
```

Sets the onBlur javascript event.

### Parameters

*onblur*

– the onBlur event.

---

## setOnChange(String) Method

```
public void setOnChange(String onchange)
```

Sets the onChange javascript event.

### Parameters

*onchange*

– the onChange event.

---

## setOnClick(String) Method

```
public void setOnClick(String onclick)
```

Sets the onClick javascript event.

### Parameters

*onclick*

– the onClick event.

---

## setOnDbClick(String) Method

```
public void setOnDbClick(String ondblclick)
```

Sets the onDbClick javascript event.

### Parameters

*ondblclick*

– the onDbClick event.

---

## setOnFocus(String) Method

```
public void setOnFocus(String onblur)
```

Sets the onFocus javascript event.

### Parameters

*onblur*

– the onFocus event.

---

## setOnKeyDown(String) Method

```
public void setOnKeyDown(String onkeydown)
```

Sets the onKeyDown javascript event.

### Parameters

*onkeydown*

– the onKeyDown event.

---

## setOnKeyPress(String) Method

```
public void setOnKeyPress(String onkeypress)
```

Sets the onKeyPress javascript event.

### Parameters

*onkeypress*

– the onKeyPress event.

---

## setOnKeyUp(String) Method

```
public void setOnKeyUp(String onkeyup)
```

Sets the onKeyUp javascript event.

### Parameters

*onkeyup*

– the onKeyUp event.

---

## setOnMouseDown(String) Method

```
public void setOnMouseDown(String onmousedown)
```

Sets the onMouseDown javascript event.

### Parameters

*onmousedown*

– the onMouseDown event.

---

## setOnMouseMove(String) Method

```
public void setOnMouseMove(String onmousemove)
```

Sets the onMouseMove javascript event.

### Parameters

*onmousemove*

– the onMouseMove event.

---

## setOnMouseOut(String) Method

```
public void setOnMouseOut(String onmouseout)
```

Sets the onMouseOut javascript event.

### Parameters

*onmouseout*

– the onMouseOut event.

---

## setOnMouseOver(String) Method

```
public void setOnMouseOver(String onmouseover)
```

Sets the onMouseOver javascript event.

### Parameters

*onmouseover*

– the onMouseOver event.

---

## setOnMouseUp(String) Method

```
public void setOnMouseUp(String onmouseup)
```

Sets the onMouseUp javascript event.

### Parameters

*onmouseup*

– the onMouseUp event.

---

## setOnSelect(String) Method

```
public void setOnSelect(String onselect)
```

Sets the onSelect javascript event.

### Parameters

*onselect*

– the onSelect event.

---

## setStyle(String) Method

```
public void setStyle(String style)
```

Sets the style of the rendered html tag.

### Parameters

*style*

– the html style.

---

## setStyleClass(String) Method

```
public void setStyleClass(String styleClass)
```

Sets the style class of the rendered html tag.

### Parameters

*styleClass*

– the html style class.

---

## setTabIndex(String) Method

```
public void setTabIndex(String tabIndex)
```

Sets the tabIndex of the rendered html tag.

### Parameters

*tabIndex*

– the tab index.

---

## setTagId(String) Method

```
public void setTagId(String tagId)
```

Set the ID of the button.

### Parameters

*tagId*

– the ID.

---

## setType(String) Method

```
public void setType(String type)
```

Set the type of the Button (submit, button, or reset).

### Parameters

*type*

– the type of the Button.

---

## setValue(String) Method

```
public void setValue(String value)
```

Set the value of the Button's text.

### Parameters

*value*

– the value of the Button's text.

# CheckBox Class

*public class* **CheckBox**

extends DefaultableDataSourceTag  
implements IAttributeConsumer

Generates a checkbox which binds to a form bean property or databound expression. CheckBox should be used on its own and not within a CheckBoxGroup. CheckBox ignores its body content. CheckBoxes can bind to boolean, Boolean, and Strings.

Attribute Descriptions			
Attribute	Required	Runtime Expression Evaluation	Data Bindable
<b>accessKey</b>	No	No	No
	Define the keyboard navigation key for the element.		
<b>alt</b>	No	No	Read Only
	Define the alt attribute of the <Input> element.		
<b>dataSource</b>	Yes	No	Read / Write
	The name of a form bean property or an expression that resolves to a bean property.		
<b>defaultValue</b>	No	No	Read Only
	A literal value or an expression that resolves to a bean property.		
<b>disabled</b>	No	No	Read Only
	A boolean value which if true will cause the CheckBox to be disabled.		
<b>tagId</b>	No	No	No
	The ID of this button used by the enclosing html tag to get the tag's real ID attribute for javascript and focus purposes. The real ID attribute id is generated based upon this name.		

## Related Topics

<netui:checkBox> Tag Sample  
 CheckBoxGroup  
 CheckBoxOption

## Hierarchy

Object  
 TagSupport  
 BodyTagSupport  
 AbstractBaseTag  
 DataSourceTag  
 DefaultableDataSourceTag  
 CheckBox

### *All Implemented Interfaces*

BodyTag, IAttributeConsumer, IterationTag, Serializable, Tag

---

## Field Summary

protected Boolean

### ***checked***

Whether the checkBox is checked or not.

public  
 static ***CHECKED***  
 final String String

public  
 static ***DISABLED***  
 final String String

public  
 static ***TYPE***  
 final String String

public  
 static ***VALUE***  
 final String String

### **Fields from com.bea.wlw.netui.tags.AbstractBaseTag**

ATTR\_GENERAL, ATTR\_GENERAL\_EXPRESSION, ATTR\_JAVASCRIPT, ATTR\_STYLE, CLASS, ID, JAVASCRIPT\_STATUS, NAME, NETUI\_UNIQUE\_CNT, ONCLICK, ONDBLCLICK, ONKEYDOWN, ONKEYPRESS, ONKEYUP, ONMOUSEDOWN, ONMOUSEMOVE, ONMOUSEOUT, ONMOUSEOVER, ONMOUSEUP, STYLE, TABINDEX

### **Fields from javax.servlet.jsp.tagext.BodyTagSupport**

bodyContent

### **Fields from javax.servlet.jsp.tagext.TagSupport**

id, pageContext

### **Fields from interface javax.servlet.jsp.tagext.BodyTag**

EVAL\_BODY\_BUFFERED, EVAL\_BODY\_TAG

**Fields from interface javax.servlet.jsp.tagext.IterationTag**

EVAL\_BODY\_AGAIN

**Fields from interface javax.servlet.jsp.tagext.Tag**

EVAL\_BODY\_INCLUDE, EVAL\_PAGE, SKIP\_BODY, SKIP\_PAGE

## Constructor

## Summary

*CheckBox()*

## Method

## Summary

```

public
    int doAfterBody()
public
    int doEndTag()
        Render the checkbox.
public
    int doStartTag()
public
    Object evaluateDefaultValue()
        Evaluate the default value of the CheckBox.
public
    String getAccessKey()
        Gets the accessKey attribute value.
public
    String getAlt()
        Gets the alt attribute on the generate <input tag.
public
    String getDisabled()
        Returns the boolean value or expression indicating the disable state of the
        CheckBox.
public
    String getOnBlur()
        Gets the onBlur javascript event.
public
    String getOnChange()
        Gets the onChange javascript event.
public
    String getOnClick()
        Gets the onClick javascript event.

```



```

public getOnDbClick()
String      Gets the onDbClick javascript event.

public
String getOnFocus()
           Gets the onFocus javascript event.

public
String getOnKeyDown()
           Gets the onKeyDown javascript event.

public
String getOnKeyPress()
           Gets the onKeyPress javascript event.

public
String getOnKeyUp()
           Gets the onKeyUp javascript event.

public
String getOnMouseDown()
           Gets the onMouseDown javascript event.

public
String getOnMouseMove()
           Gets the onMouseMove javascript event.

public
String getOnMouseOut()
           Gets the onMouseOut javascript event.

public
String getOnMouseOver()
           Gets the onMouseOver javascript event.

public
String getOnMouseUp()
           Gets the onMouseUp javascript event.

public
String getOnSelect()
           Gets the onSelect javascript event.

public
String getStyle()
           Gets the style of the rendered html tag.

public
String getStyleClass()
           Gets the style class of the rendered html tag.

public
String getTabIndex()
           Gets the tabIndex of the rendered html tag.

public
String getTagId()
           Return the ID of the checkbox.

public
String getTagName()
           Return the name of the Tag.

```

```

protected localRelease()
    void          Release any acquired resources.

public
    void setAccessKey(String accessKey)
        Sets the accessKey attribute value.

public
    void setAlt(String alt)
        Sets the alt attribute value.

public
    void setAttribute(String name, String value)
        Set an attribute value.

public
    void setDisabled(String disabled)
        Set the disable state either with the literal "true" or "false" or with an expression.

public
    void setOnBlur(String onblur)
        Sets the onBlur javascript event.

public
    void setOnChange(String onchange)
        Sets the onChange javascript event.

public
    void setOnClick(String onclick)
        Sets the onClick javascript event.

public
    void setOnDbClick(String ondblclick)
        Sets the onDbClick javascript event.

public
    void setOnFocus(String onblur)
        Sets the onFocus javascript event.

public
    void setOnKeyDown(String onkeydown)
        Sets the onKeyDown javascript event.

public
    void setOnKeyPress(String onkeypress)
        Sets the onKeyPress javascript event.

public
    void setOnKeyUp(String onkeyup)
        Sets the onKeyUp javascript event.

public
    void setOnMouseDown(String onmousedown)
        Sets the onMouseDown javascript event.

public
    void setOnMouseMove(String onmousemove)
        Sets the onMouseMove javascript event.

public
    void setOnMouseOut(String onmouseout)
        Sets the onMouseOut javascript event.

```

```

public setOnMouseOver(String onmouseover)
    void          Sets the onMouseOver javascript event.
public
    void setOnMouseUp(String onmouseup)
        Sets the onMouseUp javascript event.
public
    void setOnSelect(String onselect)
        Sets the onSelect javascript event.
public
    void setStyle(String style)
        Sets the style of the rendered html tag.
public
    void setStyleClass(String styleClass)
        Sets the style class of the rendered html tag.
public
    void setTabIndex(String tabindex)
        Sets the tabIndex of the rendered html tag.
public
    void setTagId(String tagId)
        Set the ID of the checkbox.
    
```

### Methods from **com.bea.wlw.netui.tags.DefaultableDataSourceTag**

getDefaultValue, setDefaultValue

### Methods from **com.bea.wlw.netui.tags.DataSourceTag**

getDataSource, localRelease, setDataSource

### Methods from **com.bea.wlw.netui.tags.AbstractBaseTag**

addTagIdMapping, evaluateAttributeToString, filter, filter,  
getJavaScriptUtils, getNearestForm, getScriptReporter, getUserLocale,  
localRelease, registerTagError, release, renderAttribute, rewriteName,  
write

### Methods from **javax.servlet.jsp.tagext.BodyTagSupport**

doInitBody, getBodyContent, getPreviousOut, release, setBodyContent

### Methods from **javax.servlet.jsp.tagext.TagSupport**

doAfterBody, doEndTag, doStartTag, findAncestorWithClass, getId,  
getParent, getValue, getValues, release, removeValue, setId,  
setPageContext, setParent, setValue

### Methods from class **java.lang.Object**

clone, equals, finalize, getClass, hashCode, notify, notifyAll,  
toString, wait, wait, wait

**Methods from interface `javax.servlet.jsp.tagext.BodyTag`**

`doInitBody`, `setBodyContent`

**Methods from interface `com.bea.wlw.netui.tags.IAttributeConsumer`**

`setAttribute`

**Methods from interface `javax.servlet.jsp.tagext.IterationTag`**

`doAfterBody`

**Methods from interface `javax.servlet.jsp.tagext.Tag`**

`doEndTag`, `doStartTag`, `getParent`, `release`, `setPageContext`, `setParent`

## Field Detail

### checked

protected Boolean checked

Whether the checkBox is checked or not.

---

### CHECKED

public static final String CHECKED

---

### DISABLED

public static final String DISABLED

---

### TYPE

public static final String TYPE

---

### VALUE

public static final String VALUE

## Constructor Detail

### CheckBox

```
public CheckBox()
```

## Method Detail

### doAfterBody() Method

```
public int doAfterBody()  
    throws JspException
```

#### *Overrides*

```
BodyTagSupport.doAfterBody()
```

### Exceptions

*JspException*  
if a JSP exception has occurred

---

### doEndTag() Method

```
public int doEndTag()  
    throws JspException
```

Render the checkbox.

#### *Overrides*

```
BodyTagSupport.doEndTag()
```

### Exceptions

*JspException*  
if a JSP exception has occurred

---

### doStartTag() Method

```
public int doStartTag()  
    throws JspException
```

#### *Overrides*

```
BodyTagSupport.doStartTag()
```

## Exceptions

*JspException*

if a JSP exception has occurred

---

## evaluateDefaultValue() Method

```
public Object evaluateDefaultValue()
```

Evaluate the default value of the CheckBox.

### Overrides

```
DefaultableDataSourceTag.evaluateDefaultValue()
```

---

## getAccessKey() Method

```
public String getAccessKey()
```

Gets the accessKey attribute value.

### Returns

*the accessKey value.*

---

## getAlt() Method

```
public String getAlt()
```

Gets the alt attribute on the generate <input tag.

### Returns

*the alt value.*

---

## getDisabled() Method

```
public String getDisabled()
```

Returns the boolean value or expression indicating the disable state of the CheckBox.

### Returns

*the disabled state (true or false) or an expression*

---

## getOnBlur() Method

```
public String getOnBlur()
```

Gets the onBlur javascript event.

### Returns

*the onBlur event.*

---

## getOnChange() Method

```
public String getOnChange()
```

Gets the onChange javascript event.

### Returns

*the onChange event.*

---

## getOnClick() Method

```
public String getOnClick()
```

Gets the onClick javascript event.

### Returns

*the onClick event.*

---

## getOnDbClick() Method

```
public String getOnDbClick()
```

Gets the onDbClick javascript event.

### Returns

*the onDbClick event.*

---

## getOnFocus() Method

```
public String getOnFocus()
```

Gets the onFocus javascript event.

**Returns**

*the onFocus event.*

---

**getOnKeyDown() Method**

```
public String getOnKeyDown()
```

Gets the onKeyDown javascript event.

**Returns**

*the onKeyDown event.*

---

**getOnKeyPress() Method**

```
public String getOnKeyPress()
```

Gets the onKeyPress javascript event.

**Returns**

*the onKeyPress event.*

---

**getOnKeyUp() Method**

```
public String getOnKeyUp()
```

Gets the onKeyUp javascript event.

**Returns**

*the onKeyUp event.*

---

**getOnMouseDown() Method**

```
public String getOnMouseDown()
```

Gets the onMouseDown javascript event.

**Returns**

*the onMouseDown event.*

---



## getOnMouseMove() Method

```
public String getOnMouseMove()
```

Gets the onMouseMove javascript event.

### Returns

*the onMouseMove event.*

---

## getOnMouseOut() Method

```
public String getOnMouseOut()
```

Gets the onMouseOut javascript event.

### Returns

*the onMouseOut event.*

---

## getOnMouseOver() Method

```
public String getOnMouseOver()
```

Gets the onMouseOver javascript event.

### Returns

*the onMouseOver event.*

---

## getOnMouseUp() Method

```
public String getOnMouseUp()
```

Gets the onMouseUp javascript event.

### Returns

*the onMouseUp event.*

---

## getOnSelect() Method

```
public String getOnSelect()
```

Gets the onSelect javascript event.

## Returns

*the onSelect event.*

---

## getStyle() Method

```
public String getStyle()
```

Gets the style of the rendered html tag.

## Returns

*the style.*

---

## getStyleClass() Method

```
public String getStyleClass()
```

Gets the style class of the rendered html tag.

## Returns

*the style class.*

---

## getTabIndex() Method

```
public String getTabIndex()
```

Gets the tabIndex of the rendered html tag.

## Returns

*the tabIndex.*

---

## getTagId() Method

```
public String getTagId()
```

Return the ID of the checkbox.

## Returns

*the ID.*

---

## getTagName() Method

```
public String getTagName()
```

Return the name of the Tag.

### *Overrides*

```
AbstractBaseTag.getTagName()
```

---

## localRelease() Method

```
protected void localRelease()
```

Release any acquired resources.

### *Overrides*

```
DefaultableDataSourceTag.localRelease()
```

---

## setAccessKey(String) Method

```
public void setAccessKey(String accessKey)
```

Sets the accessKey attribute value. This should key value of the keyboard navigation key. It is recommended not to use the following values because there are often used by browsers A, C, E, F, G, H, V, left arrow, and right arrow.

### Parameters

*accessKey*  
– the accessKey value.

---

## setAlt(String) Method

```
public void setAlt(String alt)
```

Sets the alt attribute value.

### Parameters

*alt*  
– the alt value.

---

## setAttribute(String, String) Method

```
public void setAttribute(String name,  
                        String value)  
    throws JspException
```

Set an attribute value. The `name` represents the name of the attribute. The `value` represents the value and may contain a netui expression. This method may result in errors being generated. This requires that the tag buffer its body and write attributes in the end tag. For the checkbox tag it is not legal to set the `id`, `name`, `type`, `value`, or `checked` attributes with this method.

### Parameters

*name*

The name of the attribute. This value may not be null or the empty string.

*value*

The value of the attribute. This may contain a netui expression.

### Exceptions

*JspException*

A `JspException` may be thrown if there is an error setting the attribute.

---

## setEnabled(String) Method

```
public void setEnabled(String disabled)
```

Set the disable state either with the literal "true" or "false" or with an expression.

### Parameters

*disabled*

– true or false or an expression

---

## setOnBlur(String) Method

```
public void setOnBlur(String onblur)
```

Sets the onBlur javascript event.

### Parameters

*onblur*

– the onBlur event.

---

## setOnChange(String) Method

```
public void setOnChange(String onchange)
```

Sets the onChange javascript event.

## Parameters

*onChange*

– the onChange event.

---

## setOnClick(String) Method

```
public void setOnClick(String onclick)
```

Sets the onClick javascript event.

## Parameters

*onclick*

– the onClick event.

---

## setOnDbClick(String) Method

```
public void setOnDbClick(String ondblclick)
```

Sets the onDbClick javascript event.

## Parameters

*ondblclick*

– the onDbClick event.

---

## setOnFocus(String) Method

```
public void setOnFocus(String onblur)
```

Sets the onFocus javascript event.

## Parameters

*onblur*

– the onFocus event.

---

## setOnKeyDown(String) Method

```
public void setOnKeyDown(String onkeydown)
```

Sets the onKeyDown javascript event.

## Parameters

*onkeydown*

– the onKeyDown event.

---

## setOnKeyPress(String) Method

```
public void setOnKeyPress(String onkeypress)
```

Sets the onKeyPress javascript event.

## Parameters

*onkeypress*

– the onKeyPress event.

---

## setOnKeyUp(String) Method

```
public void setOnKeyUp(String onkeyup)
```

Sets the onKeyUp javascript event.

## Parameters

*onkeyup*

– the onKeyUp event.

---

## setOnMouseDown(String) Method

```
public void setOnMouseDown(String onmousedown)
```

Sets the onMouseDown javascript event.

## Parameters

*onmousedown*

– the onMouseDown event.

---

## setOnMouseMove(String) Method

```
public void setOnMouseMove(String onmousemove)
```

Sets the onMouseMove javascript event.

## Parameters

*onmousemove*

– the onMouseMove event.

---

## setOnMouseOut(String) Method

```
public void setOnMouseOut(String onmouseout)
```

Sets the onMouseOut javascript event.

## Parameters

*onmouseout*

– the onMouseOut event.

---

## setOnMouseOver(String) Method

```
public void setOnMouseOver(String onmouseover)
```

Sets the onMouseOver javascript event.

## Parameters

*onmouseover*

– the onMouseOver event.

---

## setOnMouseUp(String) Method

```
public void setOnMouseUp(String onmouseup)
```

Sets the onMouseUp javascript event.

## Parameters

*onmouseup*

– the onMouseUp event.

---

## setOnSelect(String) Method

```
public void setOnSelect(String onselect)
```

Sets the onSelect javascript event.

## Parameters

*onselect*

– the onSelect event.

---

## setStyle(String) Method

```
public void setStyle(String style)
```

Sets the style of the rendered html tag.

## Parameters

*style*

– the html style.

---

## setStyleClass(String) Method

```
public void setStyleClass(String styleClass)
```

Sets the style class of the rendered html tag.

## Parameters

*styleClass*

– the html style class.

---

## setTabIndex(String) Method

```
public void setTabIndex(String tabIndex)
```

Sets the tabIndex of the rendered html tag.

## Parameters

*tabindex*

– the tab index.

---

## setTagId(String) Method

```
public void setTagId(String tagId)
```

Set the ID of the checkbox.



## Parameters

*tagId*

– the ID.

# CheckBoxGroup Class

*public class* **CheckBoxGroup**

extends `OptionsDataSourceTag`

Groups a collection of `CheckBoxOptions`, and handles databinding of their values. `CheckBoxGroup` binds to an Iterator of Strings. If `CheckBoxGroup` uses any Format tags, it must have those tags come before any nested `CheckBoxOption` tags.

Attribute Descriptions			
Attribute	Required	Runtime Expression Evaluation	Data Bindable
<b>dataSource</b>	Yes	No	Read / Write
	The name of a form bean property or an expression that contains the <code>CheckBoxGroup</code> state.		
<b>defaultValue</b>	No	No	Read Only
	A literal value or an expression that is used to set the <code>CheckBoxGroup</code> 's initial state.		
<b>disabled</b>	No	No	Read Only
	A boolean value which if true will cause the <code>CheckBoxGroup</code> to be disabled.		
<b>labelStyle</b>	No	No	No
	The style of the label for each contained <code>CheckBoxOption</code> .		
<b>labelStyleClass</b>	No	No	No
	The style class of the label for each contained <code>CheckBoxOption</code> .		
<b>optionsDataSource</b>	No	No	Read Only
	An expression that is used to generate a set of dynamic <code>CheckBoxOptions</code> . If the underlying type is a <code>com.bea.wlw.netui.tags.GroupOption</code> the name, value, alt and accesskey may be individually set. If the underlying type is a <code>Map</code> the name and value may be set individually.		
<b>tagId</b>	No	No	No

The ID of this button used by the enclosing html tag to get the tag's real ID attribute for Javascript and focus purposes. The real ID attribute id is generated based upon this name.

## Related Topics

<netui:checkBoxGroup> Tag Sample  
 <netui:checkBoxOption> Tag Sample  
 CheckBox  
 CheckBoxOption

---

## Hierarchy

```

Object
  TagSupport
    BodyTagSupport
      AbstractBaseTag
        DataSourceTag
          DefaultableDataSourceTag
            OptionsDataSourceTag
              CheckBoxGroup
  
```

## All Implemented Interfaces

BodyTag, IterationTag, Serializable, Tag

---

## Field Summary

```

public
static CHECKBOXGROUP_KEY
final String      String
  
```

### Fields from com.bea.wlw.netui.tags.AbstractBaseTag

ATTR\_GENERAL, ATTR\_GENERAL\_EXPRESSION, ATTR\_JAVASCRIPT, ATTR\_STYLE, CLASS, ID, JAVASCRIPT\_STATUS, NAME, NETUI\_UNIQUE\_CNT, ONCLICK, ONDBLCLICK, ONKEYDOWN, ONKEYPRESS, ONKEYUP, ONMOUSEDOWN, ONMOUSEMOVE, ONMOUSEOUT, ONMOUSEOVER, ONMOUSEUP, STYLE, TABINDEX

### Fields from javax.servlet.jsp.tagext.BodyTagSupport

bodyContent

### Fields from javax.servlet.jsp.tagext.TagSupport

id, pageContext

**Fields from interface javax.servlet.jsp.tagext.BodyTag**

EVAL\_BODY\_BUFFERED, EVAL\_BODY\_TAG

**Fields from interface javax.servlet.jsp.tagext.IterationTag**

EVAL\_BODY\_AGAIN

**Fields from interface javax.servlet.jsp.tagext.Tag**

EVAL\_BODY\_INCLUDE, EVAL\_PAGE, SKIP\_BODY, SKIP\_PAGE

## Constructor Summary

*CheckBoxGroup()*

## Method Summary

```

public
    int doAfterBody()
        Save the body content of the CheckBoxGroup.

public
    int doEndTag()
        Render the set of CheckBoxOptions.

public
    int doStartTag()
        Determine the set of matches for the CheckBoxGroup
protected
    Object evaluateOptionsDataSource()
        This method will evaluate the optionsDataSource and return an object.

public
    String getDisabled()
        Returns the boolean value or expression indicating the disable state of the
        CheckBoxGroup.

public
    String getLabelStyle()
        Return the label style for each contained CheckBoxOption..

public
    String getLabelStyleClass()
        Return the label style class for each contained CheckBoxOption..

public
    String getStyle()
        Gets the style of the rendered html tag.

```

```

public getStyleClass()
String      Gets the style class of the rendered html tag.

public
String getTagId()
            Return the ID of the checkbox group.

public
String getTagName()
            Return the name of the Tag.

public
boolean isMatched(String value)
            Checks whether the given value matches one of the CheckBoxGroup's selected
            CheckBoxOptions.

protected
void localRelease()
            Release any acquired resources.

public
void setDisabled(String disabled)
            Set the disable state either with the literal "true" or "false" or with an expression.

public
void setLabelStyle(String labelStyle)
            Set the label style for each contained CheckBoxOption..

public
void setLabelStyleClass(String labelStyleClass)
            Set the label style class for each contained CheckBoxOption..

public
void setStyle(String style)
            Sets the style of the rendered html tag.

public
void setStyleClass(String styleClass)
            Sets the style class of the rendered html tag.

public
void setTagId(String tagId)
            Set the ID of the checkbox group.

```

#### Methods from **com.bea.wlw.netui.tags.OptionsDataSourceTag**

getOptionsDataSource, setOptionsDataSource

#### Methods from **com.bea.wlw.netui.tags.DefaultableDataSourceTag**

getDefaultValue, localRelease, setDefaultValue

#### Methods from **com.bea.wlw.netui.tags.DataSourceTag**

getDataSource, localRelease, setDataSource

#### Methods from **com.bea.wlw.netui.tags.AbstractBaseTag**

`addTagIdMapping, evaluateAttributeToString, filter, filter,  
getJavaScriptUtils, getNearestForm, getScriptReporter, getUserLocale,  
localRelease, registerTagError, release, renderAttribute, rewriteName,  
write`

### Methods from `javax.servlet.jsp.tagext.BodyTagSupport`

`doInitBody, getBodyContent, getPreviousOut, release, setBodyContent`

### Methods from `javax.servlet.jsp.tagext.TagSupport`

`doAfterBody, doEndTag, doStartTag, findAncestorWithClass, getId,  
getParent, getValue, getValues, release, removeValue, setId,  
setPageContext, setParent, setValue`

### Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll,  
toString, wait, wait, wait`

### Methods from interface `javax.servlet.jsp.tagext.BodyTag`

`doInitBody, setBodyContent`

### Methods from interface `javax.servlet.jsp.tagext.IterationTag`

`doAfterBody`

### Methods from interface `javax.servlet.jsp.tagext.Tag`

`doEndTag, doStartTag, getParent, release, setPageContext, setParent`

## Field Detail

### CHECKBOXGROUP\_KEY

```
public static final String CHECKBOXGROUP_KEY
```

## Constructor Detail

### CheckBoxGroup

```
public CheckBoxGroup()
```

## Method Detail

### doAfterBody() Method

```
public int doAfterBody()  
    throws JspException
```

Save the body content of the CheckBoxGroup.

#### *Overrides*

```
BodyTagSupport.doAfterBody( )
```

#### Exceptions

*JspException*  
if a JSP exception has occurred

---

### doEndTag() Method

```
public int doEndTag()  
    throws JspException
```

Render the set of CheckBoxOptions.

#### *Overrides*

```
BodyTagSupport.doEndTag( )
```

#### Exceptions

*JspException*  
if a JSP exception has occurred

---

### doStartTag() Method

```
public int doStartTag()  
    throws JspException
```

Determine the set of matches for the CheckBoxGroup

#### *Overrides*

```
BodyTagSupport.doStartTag( )
```

#### Exceptions

*JspException*  
if a JSP exception has occurred

---

## evaluateOptionsDataSource() Method

```
protected Object evaluateOptionsDataSource()  
    throws JspException
```

This method will evaluate the optionsDataSource and return an object. If the object return from evaluating the expression results in a Map that is returned. Otherwise we will return an iterator. This routine will always return either a valid Iterator or Map.

### Overrides

```
OptionsDataSourceTag.evaluateOptionsDataSource()
```

### Exceptions

*JspException*

---

## getDisabled() Method

```
public String getDisabled()
```

Returns the boolean value or expression indicating the disable state of the CheckBoxGroup.

### Returns

*the disabled state (true or false) or an expression*

---

## getLabelStyle() Method

```
public String getLabelStyle()
```

Return the label style for each contained CheckBoxOption..

### Returns

*the label style*

---

## getLabelStyleClass() Method

```
public String getLabelStyleClass()
```

Return the label style class for each contained CheckBoxOption..

### Returns

*the label style*

---



## getStyle() Method

```
public String getStyle()
```

Gets the style of the rendered html tag.

### Returns

*the style.*

---

## getStyleClass() Method

```
public String getStyleClass()
```

Gets the style class of the rendered html tag.

### Returns

*the style class.*

---

## getTagId() Method

```
public String getTagId()
```

Return the ID of the checkbox group.

### Returns

*the ID.*

---

## getTagName() Method

```
public String getTagName()
```

Return the name of the Tag.

### Overrides

```
AbstractBaseTag.getTagNames()
```

---

## isMatched(String) Method

```
public boolean isMatched(String value)
```

Checks whether the given value matches one of the CheckBoxGroup's selected CheckBoxOptions.

**Parameters***value*Value to be compared

---

**localRelease() Method**

```
protected void localRelease()
```

Release any acquired resources.

**Overrides**

```
OptionsDataSourceTag.localRelease()
```

---

**setDisabled(String) Method**

```
public void setDisabled(String disabled)
```

Set the disable state either with the literal "true" or "false" or with an expression.

**Parameters***disabled*– true or false or an expression

---

**setLabelStyle(String) Method**

```
public void setLabelStyle(String labelStyle)
```

Set the label style for each contained CheckBoxOption..

**Parameters***labelStyle*– the label style

---

**setLabelStyleClass(String) Method**

```
public void setLabelStyleClass(String labelStyleClass)
```

Set the label style class for each contained CheckBoxOption..

**Parameters***labelStyleClass*– the label style

---

## setStyle(String) Method

```
public void setStyle(String style)
```

Sets the style of the rendered html tag.

### Parameters

*style*

– the html style.

---

## setStyleClass(String) Method

```
public void setStyleClass(String styleClass)
```

Sets the style class of the rendered html tag.

### Parameters

*styleClass*

– the html style class.

---

## setTagId(String) Method

```
public void setTagId(String tagId)
```

Set the ID of the checkbox group.

### Parameters

*tagId*

– the ID.

# CheckBoxOption Class

*public class* **CheckBoxOption**

extends AbstractBaseTag  
implements IAttributeConsumer

A checkbox whose state is determined by its enclosing CheckBoxGroup.

Attribute Descriptions			
Attribute	Required	Runtime Expression Evaluation	Data Bindable
accessKey	No	No	No
	Define the keyboard navigation key for the element.		
action	No	No	No
	The overriding action caused by pressing this button.		
labelStyle	No	No	No
	The style of the label for each contained CheckBoxOption		
labelStyleClass	No	No	No
	The style class of the label for each contained CheckBoxOption		
tagId	No	No	No
	The id of this control. The real id written to the HTML stream may be changed by the container (Portal), but may be looked up with the 'tagId' value through the JavaScript method <code>getNetuiTagName( tagId, tag )</code> method written by the <code>&lt;netui:html&gt;</code> tag.		
value	Yes	No	Read Only
	An expression to a bean property, or a literal, which is the value of this option		

## Related Topics

`<netui:checkBoxOption>` Tag Sample  
`<netui:checkBoxGroup>` Tag Sample  
 CheckBoxGroup  
 CheckBox

***Hierarchy***

```

Object
  TagSupport
    BodyTagSupport
      AbstractBaseTag
        CheckBoxOption

```

***All Implemented Interfaces***

BodyTag, IAttributeConsumer, IterationTag, Serializable, Tag

---

## Field Summary

```

    public
    static CHECKED
final String      String

    public
    static DISABLED
final String      String

protected String

    labelStyle
        The style of the label for each contained CheckBoxOption.

    public
    static LABELSTYLE
final String      String

    public
    static LABELSTYLECLASS
final String      String

protected String

    text
        The message text to be displayed to the user for this tag (if any)

    public
    static TYPE
final String      String

protected String

    value
        The server value for this checkbox, also used to match against the current property value
        to determine whether this checkbox should be marked as checked.

    public
    static VALUE
final String      String

```

**Fields from com.bea.wlw.netui.tags.AbstractBaseTag**

ATTR\_GENERAL, ATTR\_GENERAL\_EXPRESSION, ATTR\_JAVASCRIPT, ATTR\_STYLE, CLASS, ID, JAVASCRIPT\_STATUS, NAME, NETUI\_UNIQUE\_CNT, ONCLICK, ONDBLCLICK, ONKEYDOWN, ONKEYPRESS, ONKEYUP, ONMOUSEDOWN, ONMOUSEMOVE, ONMOUSEOUT, ONMOUSEOVER, ONMOUSEUP, STYLE, TABINDEX

**Fields from `javax.servlet.jsp.tagext.BodyTagSupport`**`bodyContent`**Fields from `javax.servlet.jsp.tagext.TagSupport`**`id, pageContext`**Fields from interface `javax.servlet.jsp.tagext.BodyTag`**`EVAL_BODY_BUFFERED, EVAL_BODY_TAG`**Fields from interface `javax.servlet.jsp.tagext.IterationTag`**`EVAL_BODY_AGAIN`**Fields from interface `javax.servlet.jsp.tagext.Tag`**`EVAL_BODY_INCLUDE, EVAL_PAGE, SKIP_BODY, SKIP_PAGE`

## Constructor Summary

*`CheckBoxOption()`*

## Method Summary

```

public
    int doAfterBody()
        Process the body text of this tag (if any).

public
    int doEndTag()
        Process the end of this tag.

public
    int doStartTag()
        Process the start of this tag.

public
    String getAccessKey()
        Gets the accessKey attribute value.

public
    String getAlt()
        Gets the alt attribute on the generate <input tag.

public
    String getLabelStyle()
        Return the label style for each contained CheckBoxOption.

```

```
public getLabelStyleClass()
String      Return the label style class for each contained CheckBoxOption.

public
String getOnBlur()
           Gets the onBlur javascript event.

public
String getOnChange()
           Gets the onChange javascript event.

public
String getOnClick()
           Gets the onClick javascript event.

public
String getOnDbClick()
           Gets the onDbClick javascript event.

public
String getOnFocus()
           Gets the onFocus javascript event.

public
String getOnKeyDown()
           Gets the onKeyDown javascript event.

public
String getOnKeyPress()
           Gets the onKeyPress javascript event.

public
String getOnKeyUp()
           Gets the onKeyUp javascript event.

public
String getOnMouseDown()
           Gets the onMouseDown javascript event.

public
String getOnMouseMove()
           Gets the onMouseMove javascript event.

public
String getOnMouseOut()
           Gets the onMouseOut javascript event.

public
String getOnMouseOver()
           Gets the onMouseOver javascript event.

public
String getOnMouseUp()
           Gets the onMouseUp javascript event.

public
String getOnSelect()
           Gets the onSelect javascript event.

public
String getStyle()
           Gets the style of the rendered html tag.
```

```

public getStyleClass()
String      Gets the style class of the rendered html tag.

public
String getTabIndex()
           Gets the tabIndex of the rendered html tag.

public
String getTagId()
           Return the ID of the tag.

public
String getTagName()
           Return the name of the Tag.

public
String getValue()
           Return the value of this CheckBoxOption.

protected
void localRelease()
           Release any acquired resources.

public
void setAccessKey(String accessKey)
           Sets the accessKey attribute value.

public
void setAlt(String alt)
           Sets the alt attribute value.

public
void setAttribute(String name, String value)
           Set an attribute value.

public
void setLabelStyle(String labelStyle)
           Set the label style for each contained CheckBoxOption.

public
void setLabelStyleClass(String labelStyleClass)
           Set the label style class for each contained CheckBoxOption.

public
void setOnBlur(String onblur)
           Sets the onBlur javascript event.

public
void setOnChange(String onchange)
           Sets the onChange javascript event.

public
void setOnClick(String onclick)
           Sets the onClick javascript event.

public
void setOnDbClick(String ondblclick)
           Sets the onDbClick javascript event.

public
void setOnFocus(String onblur)
           Sets the onFocus javascript event.

```



```

public setOnKeyDown(String onkeydown)
    void          Sets the onKeyDown javascript event.
public
    void setOnKeyPress(String onkeypress)
        Sets the onKeyPress javascript event.
public
    void setOnKeyUp(String onkeyup)
        Sets the onKeyUp javascript event.
public
    void setOnMouseDown(String onmousedown)
        Sets the onMouseDown javascript event.
public
    void setOnMouseMove(String onmousemove)
        Sets the onMouseMove javascript event.
public
    void setOnMouseOut(String onmouseout)
        Sets the onMouseOut javascript event.
public
    void setOnMouseOver(String onmouseover)
        Sets the onMouseOver javascript event.
public
    void setOnMouseUp(String onmouseup)
        Sets the onMouseUp javascript event.
public
    void setOnSelect(String onselect)
        Sets the onSelect javascript event.
public
    void setStyle(String style)
        Sets the style of the rendered html tag.
public
    void setStyleClass(String styleClass)
        Sets the style class of the rendered html tag.
public
    void setTabIndex(String tabindex)
        Sets the tabIndex of the rendered html tag.
public
    void setTagId(String tagId)
        Set the ID of the tag.
public
    void setValue(String value)
        Set the value of this CheckBoxOption.
protected
    String text()
        Return the text to be displayed to the user for this checkboxoption (if any).

```

**Methods from `com.bea.wlw.netui.tags.AbstractBaseTag`**

`addTagIdMapping, evaluateAttributeToString, filter, filter,  
getJavaScriptUtils, getNearestForm, getScriptReporter, getUserLocale,  
registerTagError, release, renderAttribute, rewriteName, write`

**Methods from `javax.servlet.jsp.tagext.BodyTagSupport`**

`doInitBody, getBodyContent, getPreviousOut, release, setBodyContent`

**Methods from `javax.servlet.jsp.tagext.TagSupport`**

`doAfterBody, doEndTag, doStartTag, findAncestorWithClass, getId,  
getParent, getValue, getValues, release, removeValue, setId,  
setPageContext, setParent, setValue`

**Methods from class `java.lang.Object`**

`clone, equals, finalize, getClass, hashCode, notify, notifyAll,  
toString, wait, wait, wait`

**Methods from interface `javax.servlet.jsp.tagext.BodyTag`**

`doInitBody, setBodyContent`

**Methods from interface `com.bea.wlw.netui.tags.IAttributeConsumer`**

`setAttribute`

**Methods from interface `javax.servlet.jsp.tagext.IterationTag`**

`doAfterBody`

**Methods from interface `javax.servlet.jsp.tagext.Tag`**

`doEndTag, doStartTag, getParent, release, setPageContext, setParent`

## Field Detail

### CHECKED

`public static final String CHECKED`

---

### DISABLED

`public static final String DISABLED`

---

## labelStyle

`protected String labelStyle`

The style of the label for each contained CheckBoxOption.

---

## LABELSTYLE

`public static final String LABELSTYLE`

---

## LABELSTYLECLASS

`public static final String LABELSTYLECLASS`

---

## text

`protected String text`

The message text to be displayed to the user for this tag (if any)

---

## TYPE

`public static final String TYPE`

---

## value

`protected String value`

The server value for this checkbox, also used to match against the current property value to determine whether this checkbox should be marked as checked.

---

## VALUE

`public static final String VALUE`

## Constructor Detail

### CheckBoxOption

`public CheckBoxOption()`

## Method Detail

### doAfterBody() Method

```
public int doAfterBody()  
    throws JspException
```

Process the body text of this tag (if any).

#### *Overrides*

```
BodyTagSupport.doAfterBody()
```

### Exceptions

*JspException*  
if a JSP exception has occurred

---

### doEndTag() Method

```
public int doEndTag()  
    throws JspException
```

Process the end of this tag.

#### *Overrides*

```
BodyTagSupport.doEndTag()
```

### Exceptions

*JspException*  
if a JSP exception has occurred

---

### doStartTag() Method

```
public int doStartTag()  
    throws JspException
```

Process the start of this tag.

#### *Overrides*

```
BodyTagSupport.doStartTag()
```

### Exceptions

*JspException*  
if a JSP exception has occurred

### **getAccessKey() Method**

```
public String getAccessKey()
```

Gets the accessKey attribute value.

#### **Returns**

*the accessKey value.*

---

### **getAlt() Method**

```
public String getAlt()
```

Gets the alt attribute on the generate <input tag.

#### **Returns**

*the alt value.*

---

### **getLabelStyle() Method**

```
public String getLabelStyle()
```

Return the label style for each contained CheckBoxOption.

#### **Returns**

*the label style*

---

### **getLabelStyleClass() Method**

```
public String getLabelStyleClass()
```

Return the label style class for each contained CheckBoxOption.

#### **Returns**

*the label style*

---

### **getOnBlur() Method**

```
public String getOnBlur()
```

Gets the onBlur javascript event.

**Returns**

*the onBlur event.*

---

**getOnChange() Method**

```
public String getOnChange()
```

Gets the onChange javascript event.

**Returns**

*the onChange event.*

---

**getOnClick() Method**

```
public String getOnClick()
```

Gets the onClick javascript event.

**Returns**

*the onClick event.*

---

**getOnDbClick() Method**

```
public String getOnDbClick()
```

Gets the onDbClick javascript event.

**Returns**

*the onDbClick event.*

---

**getOnFocus() Method**

```
public String getOnFocus()
```

Gets the onFocus javascript event.

**Returns**

*the onFocus event.*

---

## getOnKeyDown() Method

```
public String getOnKeyDown()
```

Gets the onKeyDown javascript event.

### Returns

*the onKeyDown event.*

---

## getOnKeyPress() Method

```
public String getOnKeyPress()
```

Gets the onKeyPress javascript event.

### Returns

*the onKeyPress event.*

---

## getOnKeyUp() Method

```
public String getOnKeyUp()
```

Gets the onKeyUp javascript event.

### Returns

*the onKeyUp event.*

---

## getOnMouseDown() Method

```
public String getOnMouseDown()
```

Gets the onMouseDown javascript event.

### Returns

*the onMouseDown event.*

---

## getOnMouseMove() Method

```
public String getOnMouseMove()
```

Gets the onMouseMove javascript event.

## Returns

*the onMouseMove event.*

---

## getOnMouseOut() Method

```
public String getOnMouseOut()
```

Gets the onMouseOut javascript event.

## Returns

*the onMouseOut event.*

---

## getOnMouseOver() Method

```
public String getOnMouseOver()
```

Gets the onMouseOver javascript event.

## Returns

*the onMouseOver event.*

---

## getOnMouseUp() Method

```
public String getOnMouseUp()
```

Gets the onMouseUp javascript event.

## Returns

*the onMouseUp event.*

---

## getOnSelect() Method

```
public String getOnSelect()
```

Gets the onSelect javascript event.

## Returns

*the onSelect event.*

---



## getStyle() Method

```
public String getStyle()
```

Gets the style of the rendered html tag.

### Returns

*the style.*

---

## getStyleClass() Method

```
public String getStyleClass()
```

Gets the style class of the rendered html tag.

### Returns

*the style class.*

---

## getTabIndex() Method

```
public String getTabIndex()
```

Gets the tabIndex of the rendered html tag.

### Returns

*the tabindex.*

---

## getTagId() Method

```
public String getTagId()
```

Return the ID of the tag. The id may be rewritten by the container (such as a portal) to make sure it is unique. JavaScript may lookup the actual id of the element by looking it up in the netui\_names table written into the HTML.

### Returns

*the tagId.*

---

## getTagName() Method

```
public String getTagName()
```

Return the name of the Tag.

**Overrides**

`AbstractBaseTag.getTagName()`

---

**getValue() Method**

`public String getValue()`

Return the value of this CheckBoxOption.

**Returns**

*the CheckBoxOption value*

---

**localRelease() Method**

`protected void localRelease()`

Release any acquired resources.

**Overrides**

`AbstractBaseTag.localRelease()`

---

**setAccessKey(String) Method**

`public void setAccessKey(String accessKey)`

Sets the accessKey attribute value. This should key value of the keyboard navigation key. It is recommended not to use the following values because there are often used by browsers A, C, E, F, G, H, V, left arrow, and right arrow.

**Parameters**

*accessKey*  
– the accessKey value.

---

**setAlt(String) Method**

`public void setAlt(String alt)`

Sets the alt attribute value.

**Parameters***alt*

– the alt value.

**setAttribute(String, String) Method**

```
public void setAttribute(String name,
                        String value)
    throws JspException
```

Set an attribute value. The *name* represents the name of the attribute. The *value* represents the value and may contain a netui expression. This method may result in errors being generated. This requires that the tag buffer its body and write attributes in the end tag. For the checkboxoption tag it is not legal to set the *id*, *name*, *type*, *value*, or *checked* attributes with this method.

**Parameters***name*

The name of the attribute. This value may not be null or the empty string.

*value*

The value of the attribute. This may contain a netui expression.

**Exceptions***JspException*

A JspException may be thrown if there is an error setting the attribute.

**setLabelStyle(String) Method**

```
public void setLabelStyle(String labelStyle)
```

Set the label style for each contained CheckBoxOption. The label style here will override a labelStyle at the CheckBoxGroup level.

**Parameters***labelStyle*

– the label style

**setLabelStyleClass(String) Method**

```
public void setLabelStyleClass(String labelStyleClass)
```

Set the label style class for each contained CheckBoxOption. The label style class here will override a labelStyleClass at the CheckBoxGroup level.

## Parameters

*labelStyleClass*

– the label style

---

## setOnBlur(String) Method

```
public void setOnBlur(String onblur)
```

Sets the onBlur javascript event.

## Parameters

*onblur*

– the onBlur event.

---

## setOnChange(String) Method

```
public void setOnChange(String onchange)
```

Sets the onChange javascript event.

## Parameters

*onchange*

– the onChange event.

---

## setOnClick(String) Method

```
public void setOnClick(String onclick)
```

Sets the onClick javascript event.

## Parameters

*onclick*

– the onClick event.

---

## setOnDbClick(String) Method

```
public void setOnDbClick(String ondblclick)
```

Sets the onDbClick javascript event.

**Parameters**

*ondblclick*

– the onDbClick event.

---

**setOnFocus(String) Method**

```
public void setOnFocus(String onblur)
```

Sets the onFocus javascript event.

**Parameters**

*onblur*

– the onFocus event.

---

**setOnKeyDown(String) Method**

```
public void setOnKeyDown(String onkeydown)
```

Sets the onKeyDown javascript event.

**Parameters**

*onkeydown*

– the onKeyDown event.

---

**setOnKeyPress(String) Method**

```
public void setOnKeyPress(String onkeypress)
```

Sets the onKeyPress javascript event.

**Parameters**

*onkeypress*

– the onKeyPress event.

---

**setOnKeyUp(String) Method**

```
public void setOnKeyUp(String onkeyup)
```

Sets the onKeyUp javascript event.

## Parameters

*onkeyup*

– the onKeyUp event.

---

## setOnMouseDown(String) Method

```
public void setOnMouseDown(String onmousedown)
```

Sets the onMouseDown javascript event.

## Parameters

*onmousedown*

– the onMouseDown event.

---

## setOnMouseMove(String) Method

```
public void setOnMouseMove(String onmousemove)
```

Sets the onMouseMove javascript event.

## Parameters

*onmousemove*

– the onMouseMove event.

---

## setOnMouseOut(String) Method

```
public void setOnMouseOut(String onmouseout)
```

Sets the onMouseOut javascript event.

## Parameters

*onmouseout*

– the onMouseOut event.

---

## setOnMouseOver(String) Method

```
public void setOnMouseOver(String onmouseover)
```

Sets the onMouseOver javascript event.

## Parameters

*onmouseover*

– the onMouseOver event.

---

## setOnMouseUp(String) Method

```
public void setOnMouseUp(String onmouseup)
```

Sets the onMouseUp javascript event.

## Parameters

*onmouseup*

– the onMouseUp event.

---

## setOnSelect(String) Method

```
public void setOnSelect(String onselect)
```

Sets the onSelect javascript event.

## Parameters

*onselect*

– the onSelect event.

---

## setStyle(String) Method

```
public void setStyle(String style)
```

Sets the style of the rendered html tag.

## Parameters

*style*

– the html style.

---

## setStyleClass(String) Method

```
public void setStyleClass(String styleClass)
```

Sets the style class of the rendered html tag.

## Parameters

*styleClass*  
– the html style class.

---

## setTabIndex(String) Method

```
public void setTabIndex(String tabIndex)
```

Sets the tabIndex of the rendered html tag.

## Parameters

*tabindex*  
– the tab index.

---

## setTagId(String) Method

```
public void setTagId(String tagId)
```

Set the ID of the tag.

## Parameters

*tagId*  
– the tagId.

---

## setValue(String) Method

```
public void setValue(String value)
```

Set the value of this CheckBoxOption.

## Parameters

*value*  
– the CheckBoxOption value

---

## text() Method

```
protected String text()  
    throws JspException
```

Return the text to be displayed to the user for this checkboxoption (if any).



## Exceptions

*JspException*

if an error occurs

# Content Class

*public class* **Content**

extends `AbstractBaseTag`

The Content tag is used to display text, or the result of an expression, to the page.

Attribute Descriptions			
Attribute	Required	Runtime Expression Evaluation	Data Bindable
value	Yes	No	Read Only
	The expression or value to be used to output the content.		

## Related Topics

<netui:content> Tag Sample  
Label

---

## Hierarchy

```

Object
  TagSupport
    BodyTagSupport
      AbstractBaseTag
        Content
  
```

## All Implemented Interfaces

BodyTag, IterationTag, Serializable, Tag

---

## Field Summary

### Fields from `com.bea.wlw.netui.tags.AbstractBaseTag`

ATTR\_GENERAL, ATTR\_GENERAL\_EXPRESSION, ATTR\_JAVASCRIPT, ATTR\_STYLE, CLASS, ID, JAVASCRIPT\_STATUS, NAME, NETUI\_UNIQUE\_CNT, ONCLICK, ONDBLCLICK, ONKEYDOWN, ONKEYPRESS, ONKEYUP, ONMOUSEDOWN, ONMOUSEMOVE, ONMOUSEOUT, ONMOUSEOVER, ONMOUSEUP, STYLE, TABINDEX

### Fields from `javax.servlet.jsp.tagext.BodyTagSupport`

bodyContent

**Fields from `javax.servlet.jsp.tagext.TagSupport`**`id, pageContext`**Fields from interface `javax.servlet.jsp.tagext.BodyTag`**`EVAL_BODY_BUFFERED, EVAL_BODY_TAG`**Fields from interface `javax.servlet.jsp.tagext.IterationTag`**`EVAL_BODY_AGAIN`**Fields from interface `javax.servlet.jsp.tagext.Tag`**`EVAL_BODY_INCLUDE, EVAL_PAGE, SKIP_BODY, SKIP_PAGE`

## Constructor Summary

*`Content()`*

## Method Summary

```

    public
        int doEndTag()
            Render the content.

    public
        String getDefaultValue()
            Return the default value of this Content.

    public
        String getTagName()
            Return the name of the Tag.

    public
        String getValue()
            Return the value of this Content.

protected
        void localRelease()
            Release any acquired resources.

    public
        void setDefaultValue(String defaultValue)
            Set the default value of this Content.

    public
        void setValue(String value)
            Set the value of this Content.

```

**Methods from `com.bea.wlw.netui.tags.AbstractBaseTag`**

`addTagIdMapping`, `evaluateAttributeToString`, `filter`, `filter`,  
`getJavaScriptUtils`, `getNearestForm`, `getScriptReporter`, `getUserLocale`,  
`registerTagError`, `release`, `renderAttribute`, `rewriteName`, `write`

**Methods from `javax.servlet.jsp.tagext.BodyTagSupport`**

`doAfterBody`, `doInitBody`, `doStartTag`, `getBodyContent`, `getPreviousOut`,  
`release`, `setBodyContent`

**Methods from `javax.servlet.jsp.tagext.TagSupport`**

`doAfterBody`, `doEndTag`, `doStartTag`, `findAncestorWithClass`, `getId`,  
`getParent`, `getValue`, `getValues`, `release`, `removeValue`, `setId`,  
`setPageContext`, `setParent`, `setValue`

**Methods from class `java.lang.Object`**

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`,  
`toString`, `wait`, `wait`, `wait`

**Methods from interface `javax.servlet.jsp.tagext.BodyTag`**

`doInitBody`, `setBodyContent`

**Methods from interface `javax.servlet.jsp.tagext.IterationTag`**

`doAfterBody`

**Methods from interface `javax.servlet.jsp.tagext.Tag`**

`doEndTag`, `doStartTag`, `getParent`, `release`, `setPageContext`, `setParent`

## Constructor Detail

### Content

```
public Content()
```

## Method Detail

### `doEndTag()` Method

```
public int doEndTag()  
    throws JspException
```

Render the content.

**Overrides**

`BodyTagSupport.doEndTag()`

**Exceptions**

*JspException*

if a JSP exception has occurred

---

**getDefaultValue() Method**

`public String getDefaultValue()`

Return the default value of this Content.

**Returns**

*the Content's default value*

---

**getTagName() Method**

`public String getTagName()`

Return the name of the Tag.

**Overrides**

`AbstractBaseTag.getTagName()`

---

**getValue() Method**

`public String getValue()`

Return the value of this Content.

**Returns**

*the Content's value*

---

**localRelease() Method**

`protected void localRelease()`

Release any acquired resources.

**Overrides**

`AbstractBaseTag.localRelease()`

---

## **setDefaultValue(String) Method**

```
public void setDefaultValue(String defaultValue)
```

Set the default value of this Content.

### **Parameters**

*defaultValue*

– the default value

---

## **setValue(String) Method**

```
public void setValue(String value)
```

Set the value of this Content.

### **Parameters**

*value*

– the Content value

# Error Class

*public class* **Error**

extends AbstractBaseTag

Renders an error message with a given error key value if that key can be found in the ActionErrors registered in the pageContext at `org.apache.struts.action.Action.ERROR_KEY`. Error ignores its body content.

The following optional message keys will be utilized if corresponding messages exist for them in the application resources:

- **error.prefix** – If present, the corresponding message will be rendered before each individual error message.
- **error.suffix** – If present, the corresponding message will be rendered after each individual error message.

Attribute Descriptions			
Attribute	Required	Runtime Expression Evaluation	Data Bindable
<b>bundle</b>	No	No	No
	The message bundle where the error message can be found		
<b>locale</b>	No	No	No
	The session attribute key for the locale		
<b>value</b>	Yes	No	Read Only
	The value for getting the error key		

## Related Topics

Form Validation Sample  
Errors

---

## Hierarchy

```

Object
  TagSupport
    BodyTagSupport
      AbstractBaseTag
        Error
  
```

## All Implemented Interfaces

BodyTag, IterationTag, Serializable, Tag

## Field Summary

protected String

***bundle***

The message bundle where the error messages can be found.

protected  
static Locale

***defaultLocale***

The default locale on the server.

protected String

***locale***

The session attribute key for the locale.

protected String

***value***

The value for getting the error key.

### Fields from `com.bea.wlw.netui.tags.AbstractBaseTag`

ATTR\_GENERAL, ATTR\_GENERAL\_EXPRESSION, ATTR\_JAVASCRIPT, ATTR\_STYLE, CLASS, ID, JAVASCRIPT\_STATUS, NAME, NETUI\_UNIQUE\_CNT, ONCLICK, ONDBLCLICK, ONKEYDOWN, ONKEYPRESS, ONKEYUP, ONMOUSEDOWN, ONMOUSEMOVE, ONMOUSEOUT, ONMOUSEOVER, ONMOUSEUP, STYLE, TABINDEX

### Fields from `javax.servlet.jsp.tagext.BodyTagSupport`

bodyContent

### Fields from `javax.servlet.jsp.tagext.TagSupport`

id, pageContext

### Fields from interface `javax.servlet.jsp.tagext.BodyTag`

EVAL\_BODY\_BUFFERED, EVAL\_BODY\_TAG

### Fields from interface `javax.servlet.jsp.tagext.IterationTag`

EVAL\_BODY\_AGAIN

### Fields from interface `javax.servlet.jsp.tagext.Tag`

EVAL\_BODY\_INCLUDE, EVAL\_PAGE, SKIP\_BODY, SKIP\_PAGE

## Constructor Summary

***Error()***



## Method Summary

```

public
    int doStartTag()
        Render the specified error message if it can be
        found.

public
    String getBundle()
        Returns the name of the message bundle.

public
    String getLocale()
        Returns the name of the locale attribute.

public
    String getTagName()
        Return the name of the Tag.

public
    String getValue()
        Returns the value of the error key.

protected
    void localRelease()
        Release any acquired resources.

public
    void setBundle(String bundle)
        Set the name of the message bundle.

public
    void setLocale(String locale)
        Set the name of the locale attribute.

public
    void setValue(String value)
        Set the value of the error key.

```

### Methods from `com.bea.wlw.netui.tags.AbstractBaseTag`

addTagIdMapping, evaluateAttributeToString, filter, filter,  
getJavaScriptUtils, getNearestForm, getScriptReporter, getUserLocale,  
registerTagError, release, renderAttribute, rewriteName, write

### Methods from `javax.servlet.jsp.tagext.BodyTagSupport`

doAfterBody, doEndTag, doInitBody, getBodyContent, getPreviousOut,  
release, setBodyContent

### Methods from `javax.servlet.jsp.tagext.TagSupport`

doAfterBody, doEndTag, doStartTag, findAncestorWithClass, getId,  
getParent, getValue, getValues, release, removeValue, setId,  
setPageContext, setParent, setValue

### Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

### Methods from interface `javax.servlet.jsp.tagext.BodyTag`

`doInitBody`, `setBodyContent`

### Methods from interface `javax.servlet.jsp.tagext.IterationTag`

`doAfterBody`

### Methods from interface `javax.servlet.jsp.tagext.Tag`

`doEndTag`, `doStartTag`, `getParent`, `release`, `setPageContext`, `setParent`

## Field Detail

### bundle

`protected String bundle`

The message bundle where the error messages can be found. This defaults to `org.apache.struts.action.Action.MESSAGES_KEY`.

---

### defaultLocale

`protected static Locale defaultLocale`

The default locale on the server.

---

### locale

`protected String locale`

The session attribute key for the locale. This defaults to `org.apache.struts.action.Action.LOCALE_KEY`.

---

### value

`protected String value`

The value for getting the error key. This is either a keyname or an expression which gets the key name.

## Constructor Detail

### Error

```
public Error()
```

## Method Detail

### doStartTag() Method

```
public int doStartTag()  
    throws JspException
```

Render the specified error message if it can be found.

#### *Overrides*

```
BodyTagSupport.doStartTag( )
```

### Exceptions

*JspException*  
if a JSP exception has occurred

---

### getBundle() Method

```
public String getBundle()
```

Returns the name of the message bundle.

### Returns

*the bundle's name*

---

### getLocale() Method

```
public String getLocale()
```

Returns the name of the locale attribute.

### Returns

*the locale attribute name*

---

## getTagName() Method

```
public String getTagName()
```

Return the name of the Tag.

### *Overrides*

```
AbstractBaseTag.getTagName()
```

---

## getValue() Method

```
public String getValue()
```

Returns the value of the error key.

### **Returns**

*the error key value*

---

## localRelease() Method

```
protected void localRelease()
```

Release any acquired resources.

### *Overrides*

```
AbstractBaseTag.localRelease()
```

---

## setBundle(String) Method

```
public void setBundle(String bundle)
```

Set the name of the message bundle.

### **Parameters**

*bundle*

– the bundle name

---

## setLocale(String) Method

```
public void setLocale(String locale)
```

Set the name of the locale attribute.

## Parameters

*locale*

– the locale attribute name

---

## setValue(String) Method

```
public void setValue(String value)
```

Set the value of the error key.

## Parameters

*value*

– the error key value

# Errors Class

*public class* **Errors**

extends AbstractBaseTag

Renders the set of error messages found in the ActionErrors registered in the pageContext at org.apache.struts.action.Action.ERROR\_KEY. Errors ignores its body content.

The following optional message keys will be utilized if corresponding messages exist for them in the application resources:

- **errors.header** – If present, the corresponding message will be rendered prior to the individual list of error messages.
- **errors.footer** – If present, the corresponding message will be rendered following the individual list of error messages.
- **errors.prefix** – If present, the corresponding message will be rendered before each individual error message.
- **errors.suffix** – If present, the corresponding message will be rendered after each individual error message.

Attribute Descriptions			
Attribute	Required	Runtime Expression Evaluation	Data Bindable
<b>bundle</b>	No	No	No
	The message bundle where the error messages can be found		
<b>locale</b>	No	No	No
	The session attribute key for the locale		

## Related Topics

Form Validation Sample  
Error

---

## Hierarchy

```

Object
  TagSupport
    BodyTagSupport
      AbstractBaseTag
        Errors
  
```

## All Implemented Interfaces

BodyTag, IterationTag, Serializable, Tag

## Field Summary

protected String

### ***bundle***

The message bundle where the error messages can be found.

protected  
static Locale

### ***defaultLocale***

The default locale on our server.

protected String

### ***locale***

The session attribute key for the locale.

### Fields from `com.bea.wlw.netui.tags.AbstractBaseTag`

ATTR\_GENERAL, ATTR\_GENERAL\_EXPRESSION, ATTR\_JAVASCRIPT, ATTR\_STYLE, CLASS, ID, JAVASCRIPT\_STATUS, NAME, NETUI\_UNIQUE\_CNT, ONCLICK, ONDBLCLICK, ONKEYDOWN, ONKEYPRESS, ONKEYUP, ONMOUSEDOWN, ONMOUSEMOVE, ONMOUSEOUT, ONMOUSEOVER, ONMOUSEUP, STYLE, TABINDEX

### Fields from `javax.servlet.jsp.tagext.BodyTagSupport`

bodyContent

### Fields from `javax.servlet.jsp.tagext.TagSupport`

id, pageContext

### Fields from interface `javax.servlet.jsp.tagext.BodyTag`

EVAL\_BODY\_BUFFERED, EVAL\_BODY\_TAG

### Fields from interface `javax.servlet.jsp.tagext.IterationTag`

EVAL\_BODY\_AGAIN

### Fields from interface `javax.servlet.jsp.tagext.Tag`

EVAL\_BODY\_INCLUDE, EVAL\_PAGE, SKIP\_BODY, SKIP\_PAGE

## Constructor Summary

*Errors()*

## Method Summary

```

public
    int doStartTag()
        Render the specified error messages if there are
        any.

public
    String getBundle()
        Returns the name of the message bundle.

public
    String getLocale()
        Returns the name of the locale attribute.

public
    String getTagName()
        Return the name of the Tag.

protected
    void localRelease()
        Release any acquired resources.

public
    void setBundle(String bundle)
        Set the name of the message bundle.

public
    void setLocale(String locale)
        Set the name of the locale attribute.

```

### Methods from `com.bea.wlw.netui.tags.AbstractBaseTag`

addTagIdMapping, evaluateAttributeToString, filter, filter,  
getJavaScriptUtils, getNearestForm, getScriptReporter, getUserLocale,  
registerTagError, release, renderAttribute, rewriteName, write

### Methods from `javax.servlet.jsp.tagext.BodyTagSupport`

doAfterBody, doEndTag, doInitBody, getBodyContent, getPreviousOut,  
release, setBodyContent

### Methods from `javax.servlet.jsp.tagext.TagSupport`

doAfterBody, doEndTag, doStartTag, findAncestorWithClass, getId,  
getParent, getValue, getValues, release, removeValue, setId,  
setPageContext, setParent, setValue

### Methods from class `java.lang.Object`

clone, equals, finalize, getClass, hashCode, notify, notifyAll,  
toString, wait, wait, wait

### Methods from interface `javax.servlet.jsp.tagext.BodyTag`



`doInitBody, setBodyContent`

#### Methods from interface `javax.servlet.jsp.tagext.IterationTag`

`doAfterBody`

#### Methods from interface `javax.servlet.jsp.tagext.Tag`

`doEndTag, doStartTag, getParent, release, setPageContext, setParent`

## Field Detail

### bundle

`protected String bundle`

The message bundle where the error messages can be found. This defaults to `org.apache.struts.action.Action.MESSAGES_KEY`.

---

### defaultLocale

`protected static Locale defaultLocale`

The default locale on our server.

---

### locale

`protected String locale`

The session attribute key for the locale. This defaults to `org.apache.struts.action.Action.LOCALE_KEY`.

## Constructor Detail

### Errors

`public Errors()`

## Method Detail

### doStartTag() Method

```
public int doStartTag()  
    throws JspException
```

Render the specified error messages if there are any.

#### *Overrides*

`BodyTagSupport.doStartTag()`

### Exceptions

*JspException*  
if a JSP exception has occurred

---

### getBundle() Method

```
public String getBundle()
```

Returns the name of the message bundle.

#### Returns

*the bundle's name*

---

### getLocale() Method

```
public String getLocale()
```

Returns the name of the locale attribute.

#### Returns

*the locale attribute name*

---

### getTagName() Method

```
public String getTagName()
```

Return the name of the Tag.

#### *Overrides*

`AbstractBaseTag.getTagName()`

---

## localRelease() Method

```
protected void localRelease()
```

Release any acquired resources.

### *Overrides*

```
AbstractBaseTag.localRelease()
```

---

## setBundle(String) Method

```
public void setBundle(String bundle)
```

Set the name of the message bundle.

### Parameters

*bundle*

– the bundle name

---

## setLocale(String) Method

```
public void setLocale(String locale)
```

Set the name of the locale attribute.

### Parameters

*locale*

– the locale attribute name

# Exceptions Class

*public class* **Exceptions**

extends TagSupport

Renders formatted exception data, as found in the Request with the key:

`org.apache.struts.action.Action.EXCEPTION_KEY`. Exceptions ignores its body content.

Attribute Descriptions			
Attribute	Required	Runtime Expression Evaluation	Data Bindable
<b>showMessage</b>	No	No	No
	Determines whether or not the message of the exception should be shown		
<b>showStackTrace</b>	No	No	No
	Returns whether or not the stack trace is being shown		

## Related Topics

<netui:exceptions> Tag Sample

---

## Hierarchy

```

Object
  TagSupport
    Exceptions
  
```

## All Implemented Interfaces

IterationTag, Serializable, Tag

---

## Field Summary

protected boolean

### **showMessage**

Determines whether or not the message of the exception should be shown.

protected boolean

### **showStackTrace**

Determines whether or not the stack trace of the exception should be

shown.

#### Fields from `javax.servlet.jsp.tagext.TagSupport`

`id`, `pageContext`

#### Fields from interface `javax.servlet.jsp.tagext.IterationTag`

`EVAL_BODY_AGAIN`

#### Fields from interface `javax.servlet.jsp.tagext.Tag`

`EVAL_BODY_INCLUDE`, `EVAL_PAGE`, `SKIP_BODY`, `SKIP_PAGE`

## Constructor Summary

*Exceptions()*

## Method Summary

```

public
    int doStartTag()
        Render the exception text based on the display attributes.

public
boolean getShowMessage()
    Returns whether or not the exception message is being
    shown.

public
boolean getShowStackTrace()
    Returns whether or not the stack trace is being shown.

protected
    void localRelease()
        Release any acquired resources.

public
    void setShowMessage(boolean showMessage)
        Set whether or not the exception message is being shown.

public
    void setShowStackTrace(boolean showStackTrace)
        Set whether or not the stack trace is being shown.

```

#### Methods from `javax.servlet.jsp.tagext.TagSupport`

`doAfterBody`, `doEndTag`, `findAncestorWithClass`, `getId`, `getParent`,  
`getValue`, `getValues`, `release`, `removeValue`, `setId`, `setPageContext`,  
`setParent`, `setValue`

### Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

### Methods from interface `javax.servlet.jsp.tagext.IterationTag`

`doAfterBody`

### Methods from interface `javax.servlet.jsp.tagext.Tag`

`doEndTag`, `doStartTag`, `getParent`, `release`, `setPageContext`, `setParent`

## Field Detail

### `showMessage`

`protected boolean showMessage`

Determines whether or not the message of the exception should be shown. Defaults to true.

---

### `showStackTrace`

`protected boolean showStackTrace`

Determines whether or not the stack trace of the exception should be shown. Defaults to false.

## Constructor Detail

### Exceptions

`public Exceptions()`

## Method Detail

### `doStartTag()` Method

`public int doStartTag()`  
    throws `JspException`

Render the exception text based on the display attributes.

### *Overrides*

`TagSupport.doStartTag()`

### **Exceptions**

*JspException*

if a JSP exception has occurred

---

## **getShowMessage() Method**

`public boolean getShowMessage()`

Returns whether or not the exception message is being shown.

### **Returns**

*the showMessage state*

---

## **getShowStackTrace() Method**

`public boolean getShowStackTrace()`

Returns whether or not the stack trace is being shown.

### **Returns**

*the showStackTrace state*

---

## **localRelease() Method**

`protected void localRelease()`

Release any acquired resources.

---

## **setShowMessage(boolean) Method**

`public void setShowMessage(boolean showMessage)`

Set whether or not the exception message is being shown.

### **Parameters**

*showMessage*

– true or false depending on the setting desired

---

## **setShowStackTrace(boolean) Method**

```
public void setShowStackTrace(boolean showStackTrace)
```

Set whether or not the stack trace is being shown.

### **Parameters**

*showStackTrace*

– true or false depending on the setting desired



# FileUpload Class

*public class FileUpload*

extends DataSourceTag

implements IAttributeConsumer

Renders an input tag with type="file", with the given attributes. Usage of this tag requires the parent Form tag to have enctype="multipart/form-data".

Attribute Descriptions			
Attribute	Required	Runtime Expression Evaluation	Data Bindable
accept	No	No	No
	The set of MIME types accepted for file upload.		
dataSource	Yes	No	Write Only
	The name of a form bean property or an expression that contains the FormFile.		
readOnly	No	No	No
	Whether or not the TextBox is read-only.		
size	No	No	No
	The visible number of characters in the FileUpload.		
tagId	No	No	No
	The ID of this button used by the enclosing html tag to get the tag's real ID attribute for javascript and focus purposes. The real ID attribute id is generated based upon this name.		

## Notes

- The dataSource for the FileUpload tag is write-only. This means that the value of the FileUpload cannot get read back into the FileUpload on postback.
- The page flow or form bean property pointed to with the FileUpload's dataSource should be of type `org.apache.struts.upload.FormFile`.

## Related Topics

<netui:fileUpload> Tag Sample

BinaryFlow Sample

Handling Images and Binary Data in Page Flows

Form

***Hierarchy***

```

Object
  TagSupport
    BodyTagSupport
      AbstractBaseTag
        DataSourceTag
          FileUpload

```

***All Implemented Interfaces***

BodyTag, IAttributeConsumer, IterationTag, Serializable, Tag

**Field****Summary**

```

public
static READONLY
final String      String

public
static TYPE
final String      String

```

**Fields from com.bea.wlw.netui.tags.AbstractBaseTag**

ATTR\_GENERAL, ATTR\_GENERAL\_EXPRESSION, ATTR\_JAVASCRIPT, ATTR\_STYLE, CLASS, ID, JAVASCRIPT\_STATUS, NAME, NETUI\_UNIQUE\_CNT, ONCLICK, ONDBLCLICK, ONKEYDOWN, ONKEYPRESS, ONKEYUP, ONMOUSEDOWN, ONMOUSEMOVE, ONMOUSEOUT, ONMOUSEOVER, ONMOUSEUP, STYLE, TABINDEX

**Fields from javax.servlet.jsp.tagext.BodyTagSupport**

bodyContent

**Fields from javax.servlet.jsp.tagext.TagSupport**

id, pageContext

**Fields from interface javax.servlet.jsp.tagext.BodyTag**

EVAL\_BODY\_BUFFERED, EVAL\_BODY\_TAG

**Fields from interface javax.servlet.jsp.tagext.IterationTag**

EVAL\_BODY\_AGAIN

**Fields from interface javax.servlet.jsp.tagext.Tag**

EVAL\_BODY\_INCLUDE, EVAL\_PAGE, SKIP\_BODY, SKIP\_PAGE

## Constructor

### Summary

*FileUpload()*

## Method

### Summary

```

public
    int doAfterBody()
        Save the body content of the FileUpLoad.

public
    int doEndTag()
        Render the FileUpload.

public
    int doStartTag()
        Prepare FileUpload for rendering

public
    String getAccept()
        Gets the mime-types accepted for the
        FileUpload.

public
    String getOnBlur()
        Gets the onBlur javascript event.

public
    String getOnChange()
        Gets the onChange javascript event.

public
    String getOnClick()
        Gets the onClick javascript event.

public
    String getOnDbClick()
        Gets the onDbClick javascript event.

public
    String getOnFocus()
        Gets the onFocus javascript event.

public
    String getOnKeyDown()
        Gets the onKeyDown javascript event.

public
    String getOnKeyPress()
        Gets the onKeyPress javascript event.

public
    String getOnKeyUp()
        Gets the onKeyUp javascript event.

```

```

public getOnMouseDown()
String      Gets the onMouseDown javascript event.

public
String getOnMouseMove()
           Gets the onMouseMove javascript event.

public
String getOnMouseOut()
           Gets the onMouseOut javascript event.

public
String getOnMouseOver()
           Gets the onMouseOver javascript event.

public
String getOnMouseUp()
           Gets the onMouseUp javascript event.

public
String getOnSelect()
           Gets the onSelect javascript event.

public
boolean getReadonly()
           Gets if this FileUpload is read-only.

public
String getSize()
           Gets the size (in characters) of the FileUpload.

public
String getStyle()
           Gets the style of the rendered html tag.

public
String getStyleClass()
           Gets the style class of the rendered html tag.

public
String getTabIndex()
           Gets the tabIndex of the rendered html tag.

public
String getTagId()
           Return the ID of the FileUpload.

public
String getTagName()
           Return the name of the Tag.

protected
void localRelease()
           Release any acquired resources.

public
void setAccept(String accept)
           Set the mime-types accepted for the FileUpload.

public
void setAttribute(String name, String value)
           Set an attribute value.

```

```

public setOnBlur(String onblur)
    void          Sets the onBlur javascript event.
public
    void setOnChange(String onchange)
        Sets the onChange javascript event.
public
    void setOnClick(String onclick)
        Sets the onClick javascript event.
public
    void setOnDbClick(String ondblclick)
        Sets the onDbClick javascript event.
public
    void setOnFocus(String onblur)
        Sets the onFocus javascript event.
public
    void setOnKeyDown(String onkeydown)
        Sets the onKeyDown javascript event.
public
    void setOnKeyPress(String onkeypress)
        Sets the onKeyPress javascript event.
public
    void setOnKeyUp(String onkeyup)
        Sets the onKeyUp javascript event.
public
    void setOnMouseDown(String onmousedown)
        Sets the onMouseDown javascript event.
public
    void setOnMouseMove(String onmousemove)
        Sets the onMouseMove javascript event.
public
    void setOnMouseOut(String onmouseout)
        Sets the onMouseOut javascript event.
public
    void setOnMouseOver(String onmouseover)
        Sets the onMouseOver javascript event.
public
    void setOnMouseUp(String onmouseup)
        Sets the onMouseUp javascript event.
public
    void setOnSelect(String onselect)
        Sets the onSelect javascript event.
public
    void setReadOnly(boolean readonly)
        Set if this FileUpload is read-only.
public
    void setSize(String size)
        Set the size (in characters) of the FileUpload.

```

```

public setStyle(String style)
    void          Sets the style of the rendered html tag.
public
    void setStyleClass(String styleClass)
        Sets the style class of the rendered html tag.
public
    void setTabIndex(String tabIndex)
        Sets the tabIndex of the rendered html tag.
public
    void setTagId(String tagId)
        Set the ID of the FileUpload.
    
```

### Methods from **com.bea.wlw.netui.tags.DataSourceTag**

getDataSource, setDataSource

### Methods from **com.bea.wlw.netui.tags.AbstractBaseTag**

addTagIdMapping, evaluateAttributeToString, filter, filter,  
getJavaScriptUtils, getNearestForm, getScriptReporter, getUserLocale,  
localRelease, registerTagError, release, renderAttribute, rewriteName,  
write

### Methods from **javax.servlet.jsp.tagext.BodyTagSupport**

doInitBody, getBodyContent, getPreviousOut, release, setBodyContent

### Methods from **javax.servlet.jsp.tagext.TagSupport**

doAfterBody, doEndTag, doStartTag, findAncestorWithClass, getId,  
getParent, getValue, getValues, release, removeValue, setId,  
setPageContext, setParent, setValue

### Methods from class **java.lang.Object**

clone, equals, finalize, getClass, hashCode, notify, notifyAll,  
toString, wait, wait, wait

### Methods from interface **javax.servlet.jsp.tagext.BodyTag**

doInitBody, setBodyContent

### Methods from interface **com.bea.wlw.netui.tags.IAttributeConsumer**

setAttribute

### Methods from interface **javax.servlet.jsp.tagext.IterationTag**

doAfterBody

Methods from interface `javax.servlet.jsp.tagext.Tag`

`doEndTag`, `doStartTag`, `getParent`, `release`, `setPageContext`, `setParent`

## Field Detail

### READONLY

```
public static final String READONLY
```

---

### TYPE

```
public static final String TYPE
```

## Constructor Detail

### FileUpload

```
public FileUpload()
```

## Method Detail

### doAfterBody() Method

```
public int doAfterBody()
    throws JspException
```

Save the body content of the FileUpLoad. In this case any body content will be thrown away.

#### *Overrides*

```
BodyTagSupport.doAfterBody()
```

### Exceptions

*JspException*  
if a JSP exception has occurred

---

### doEndTag() Method

```
public int doEndTag()
    throws JspException
```

Render the FileUpload.

**Overrides**

`BodyTagSupport.doEndTag()`

**Exceptions**

*JspException*

if a JSP exception has occurred

---

**doStartTag() Method**

```
public int doStartTag()
    throws JspException
```

Prepare FileUpload for rendering

**Overrides**

`BodyTagSupport.doStartTag()`

**Exceptions**

*JspException*

if a JSP exception has occurred

---

**getAccept() Method**

```
public String getAccept()
```

Gets the mime-types accepted for the FileUpload.

**Returns**

*the accepted mime-types*

---

**getOnBlur() Method**

```
public String getOnBlur()
```

Gets the onBlur javascript event.

**Returns**

*the onBlur event.*

---



## getOnChange() Method

```
public String getOnChange()
```

Gets the onChange javascript event.

### Returns

*the onChange event.*

---

## getOnClick() Method

```
public String getOnClick()
```

Gets the onClick javascript event.

### Returns

*the onClick event.*

---

## getOnDbClick() Method

```
public String getOnDbClick()
```

Gets the onDbClick javascript event.

### Returns

*the onDbClick event.*

---

## getOnFocus() Method

```
public String getOnFocus()
```

Gets the onFocus javascript event.

### Returns

*the onFocus event.*

---

## getOnKeyDown() Method

```
public String getOnKeyDown()
```

Gets the onKeyDown javascript event.

## Returns

*the onKeyDown event.*

---

## getOnKeyPress() Method

```
public String getOnKeyPress()
```

Gets the onKeyPress javascript event.

## Returns

*the onKeyPress event.*

---

## getOnKeyUp() Method

```
public String getOnKeyUp()
```

Gets the onKeyUp javascript event.

## Returns

*the onKeyUp event.*

---

## getOnMouseDown() Method

```
public String getOnMouseDown()
```

Gets the onMouseDown javascript event.

## Returns

*the onMouseDown event.*

---

## getOnMouseMove() Method

```
public String getOnMouseMove()
```

Gets the onMouseMove javascript event.

## Returns

*the onMouseMove event.*

---

## getOnMouseOut() Method

```
public String getOnMouseOut()
```

Gets the onMouseOut javascript event.

### Returns

*the onMouseOut event.*

---

## getOnMouseOver() Method

```
public String getOnMouseOver()
```

Gets the onMouseOver javascript event.

### Returns

*the onMouseOver event.*

---

## getOnMouseUp() Method

```
public String getOnMouseUp()
```

Gets the onMouseUp javascript event.

### Returns

*the onMouseUp event.*

---

## getOnSelect() Method

```
public String getOnSelect()
```

Gets the onSelect javascript event.

### Returns

*the onSelect event.*

---

## getReadonly() Method

```
public boolean getReadonly()
```

Gets if this FileUpload is read-only.

## Returns

*the read-only state*

---

## getSize() Method

```
public String getSize()
```

Gets the size (in characters) of the FileUpload.

## Returns

*the size*

---

## getStyle() Method

```
public String getStyle()
```

Gets the style of the rendered html tag.

## Returns

*the style.*

---

## getStyleClass() Method

```
public String getStyleClass()
```

Gets the style class of the rendered html tag.

## Returns

*the style class.*

---

## getTabIndex() Method

```
public String getTabIndex()
```

Gets the tabIndex of the rendered html tag.

## Returns

*the tabIndex.*

---

## getTagId() Method

```
public String getTagId()
```

Return the ID of the FileUpload.

### Returns

*the ID.*

---

## getTagName() Method

```
public String getTagName()
```

Return the name of the Tag.

### Overrides

```
AbstractBaseTag.getTagName()
```

---

## localRelease() Method

```
protected void localRelease()
```

Release any acquired resources.

### Overrides

```
DataSourceTag.localRelease()
```

---

## setAccept(String) Method

```
public void setAccept(String accept)
```

Set the mime-types accepted for the FileUpload.

### Parameters

*accept*

– the accepted mime-types

---

## setAttribute(String, String) Method

```
public void setAttribute(String name,  
                        String value)  
    throws JspException
```

Set an attribute value. The name represents the name of the attribute. The value represents the value and may contain a netui expression. This method may result in errors being generated. This requires that the tag

buffer its body and write attributes in the end tag. For the fileupload tag it is not legal to set the `id`, `name` or `type` attributes with this method.

### Parameters

*name*

The name of the attribute. This value may not be null or the empty string.

*value*

The value of the attribute. This may contain a netui expression.

### Exceptions

*JspException*

A *JspException* may be thrown if there is an error setting the attribute.

---

### setOnBlur(String) Method

```
public void setOnBlur(String onblur)
```

Sets the onBlur javascript event.

#### Parameters

*onblur*

– the onBlur event.

---

### setOnChange(String) Method

```
public void setOnChange(String onchange)
```

Sets the onChange javascript event.

#### Parameters

*onchange*

– the onChange event.

---

### setOnClick(String) Method

```
public void setOnClick(String onclick)
```

Sets the onClick javascript event.

**Parameters***onclick*

– the onClick event.

---

**setOnDbClick(String) Method**

```
public void setOnDbClick(String ondblclick)
```

Sets the onDbClick javascript event.

**Parameters***ondblclick*

– the onDbClick event.

---

**setOnFocus(String) Method**

```
public void setOnFocus(String onblur)
```

Sets the onFocus javascript event.

**Parameters***onblur*

– the onFocus event.

---

**setOnKeyDown(String) Method**

```
public void setOnKeyDown(String onkeydown)
```

Sets the onKeyDown javascript event.

**Parameters***onkeydown*

– the onKeyDown event.

---

**setOnKeyPress(String) Method**

```
public void setOnKeyPress(String onkeypress)
```

Sets the onKeyPress javascript event.

**Parameters**

*onkeypress*

– the onKeyPress event.

---

**setOnKeyUp(String) Method**

```
public void setOnKeyUp(String onkeyup)
```

Sets the onKeyUp javascript event.

**Parameters**

*onkeyup*

– the onKeyUp event.

---

**setOnMouseDown(String) Method**

```
public void setOnMouseDown(String onmousedown)
```

Sets the onMouseDown javascript event.

**Parameters**

*onmousedown*

– the onMouseDown event.

---

**setOnMouseMove(String) Method**

```
public void setOnMouseMove(String onmousemove)
```

Sets the onMouseMove javascript event.

**Parameters**

*onmousemove*

– the onMouseMove event.

---

**setOnMouseOut(String) Method**

```
public void setOnMouseOut(String onmouseout)
```

Sets the onMouseOut javascript event.



## Parameters

*onmouseout*

– the onMouseOut event.

---

## setOnMouseOver(String) Method

```
public void setOnMouseOver(String onmouseover)
```

Sets the onMouseOver javascript event.

## Parameters

*onmouseover*

– the onMouseOver event.

---

## setOnMouseUp(String) Method

```
public void setOnMouseUp(String onmouseup)
```

Sets the onMouseUp javascript event.

## Parameters

*onmouseup*

– the onMouseUp event.

---

## setOnSelect(String) Method

```
public void setOnSelect(String onselect)
```

Sets the onSelect javascript event.

## Parameters

*onselect*

– the onSelect event.

---

## setReadOnly(boolean) Method

```
public void setReadOnly(boolean readonly)
```

Set if this FileUpload is read-only.

## Parameters

*readonly*

– the read-only state

---

## setSize(String) Method

```
public void setSize(String size)
```

Set the size (in characters) of the FileUpload.

## Parameters

*size*

– the size

---

## setStyle(String) Method

```
public void setStyle(String style)
```

Sets the style of the rendered html tag.

## Parameters

*style*

– the html style.

---

## setStyleClass(String) Method

```
public void setStyleClass(String styleClass)
```

Sets the style class of the rendered html tag.

## Parameters

*styleClass*

– the html style class.

---

## setTabIndex(String) Method

```
public void setTabIndex(String tabIndex)
```

Sets the tabIndex of the rendered html tag.

## Parameters

*tabindex*

– the tab index.

---

## setTagId(String) Method

```
public void setTagId(String tagId)
```

Set the ID of the FileUpload.

## Parameters

*tagId*

– the ID.

# Form Class

*public class Form*

extends AbstractBaseTag  
implements IAttributeConsumer, URLParams

This tag represents an input form, associated with a bean whose properties correspond to the various fields of the form.

Attribute Descriptions			
Attribute	Required	Runtime Expression Evaluation	Data Bindable
<b>action</b>	Yes	No	No
	Required. The action invoked by submitting the Form.		
<b>encType</b>	No	No	No
	The content encoding to be used on a post submit.		
<b>focus</b>	No	No	No
	The tagID of a nested Netui tag which should receive the focus.		
<b>location</b>	No	No	No
	The location hash to append to the url.		
<b>method</b>	No	No	No
	The request method used when submitting this form.		
<b>name</b>	No	No	No
	The attribute key under which the associated ActionForm bean used to populate the input form is stored. This ActionForm is found in the scope defined by the <code>scope</code> attribute.		
<b>scope</b>	No	No	No
	The scope ('request' or 'session') under which the associated ActionForm bean used to populate the form input fields is stored. Using the name, type and scope attributes defines an the ActionForm object used.		
<b>tagId</b>	No	No	No
	The ID of this button used by the enclosing html tag to get the tag's real ID attribute for javascript and focus purposes. The real ID attribute id is generated based upon this name.		

<b>target</b>	No	No	No
	The window target		
<b>type</b>	No	No	No
	The Java class name of the ActionForm bean to be created, if necessary. This bean will be created if the name and scope attributes are set. The bean is then used to populate the form input fields.		

**Related Topics**[<netui:form> Tag Sample](#)[Using Form Beans to Encapsulate Data](#)***Hierarchy***

```

Object
  TagSupport
    BodyTagSupport
      AbstractBaseTag
        Form
  
```

***All Implemented Interfaces***

BodyTag, IAttributeConsumer, IterationTag, Serializable, Tag, URLParams

## Field Summary

```

    public
    static ACTION
final String
    String
protected String
    beanScope
        The scope of the form bean to (create and) use.
protected String
    beanType
        The type of the form bean to (create and) use.
protected String
    enctype
        The content encoding to be used on a post submit.
protected String
    focus
        The name of the field to receive focus, if any.
protected String
    location
        The location hash to append to the url.
  
```

```
protected String method
    The request method used when submitting this form.
    public
    static METHOD
    final String
        String
protected String
    name
    The attribute key under which our associated bean is stored.
protected String
    scope
    The scope (request or session) under which our associated bean is stored.
protected String
    target
    The window target.
protected String
    type
    The Java class name of the bean to be created, if necessary.
```

## Fields from com.bea.wlw.netui.tags.AbstractBaseTag

```
ATTR_GENERAL, ATTR_GENERAL_EXPRESSION, ATTR_JAVASCRIPT, ATTR_STYLE,
CLASS, ID, JAVASCRIPT_STATUS, NAME, NETUI_UNIQUE_CNT, ONCLICK,
ONDBLCLICK, ONKEYDOWN, ONKEYPRESS, ONKEYUP, ONMOUSEDOWN, ONMOUSEMOVE,
ONMOUSEOUT, ONMOUSEOVER, ONMOUSEUP, STYLE, TABINDEX
```

## Fields from javax.servlet.jsp.tagext.BodyTagSupport

```
bodyContent
```

## Fields from javax.servlet.jsp.tagext.TagSupport

```
id, pageContext
```

## Fields from interface javax.servlet.jsp.tagext.BodyTag

```
EVAL_BODY_BUFFERED, EVAL_BODY_TAG
```

## Fields from interface javax.servlet.jsp.tagext.IterationTag

```
EVAL_BODY_AGAIN
```

## Fields from interface javax.servlet.jsp.tagext.Tag

```
EVAL_BODY_INCLUDE, EVAL_PAGE, SKIP_BODY, SKIP_PAGE
```

## Constructor Summary

*Form()*

## Method Summary

```

public
    void addParameter(String name, Object value)
        Adds a URL parameter to the generated hyperlink.

public
    void addTagID(String tagID, String name)
        Adds a tagId and name to the Form's focusMap.

public
    int doAfterBody()
        Save the body content of the Form.

public
    int doEndTag()
        Render the end of this form.

public
    int doStartTag()
        Render the beginning of this form.

public
    void generateRealName()

public
    String getAction()
        Return the action of the Form.

public
    String getBeanName()
        Return the name of the form bean corresponding to this tag.

public
    String getEnctype()
        Return the content encoding to be used on a post submit.

public
    String getFocus()
        Return the name of the field to receive focus.

public
    String getLocation()
        Return the location hash to append to the url.

public
    String getMethod()
        Return request method used when submitting this form.

public
    String getName()
        Return the attribute key under which our associated bean is stored.

public
    String getOnClick()
        Gets the onClick javascript event.

```

```

public getOnDbClick()
String      Gets the onDbClick javascript event.

public
String getOnKeyDown()
           Gets the onKeyDown javascript event.

public
String getOnKeyPress()
           Gets the onKeyPress javascript event.

public
String getOnKeyUp()
           Gets the onKeyUp javascript event.

public
String getOnMouseDown()
           Gets the onMouseDown javascript event.

public
String getOnMouseMove()
           Gets the onMouseMove javascript event.

public
String getOnMouseOut()
           Gets the onMouseOut javascript event.

public
String getOnMouseOver()
           Gets the onMouseOver javascript event.

public
String getOnMouseUp()
           Gets the onMouseUp javascript event.

public
String getOnReset()
           Gets the onReset javascript event.

public
String getOnSubmit()
           Gets the onSubmit javascript event.

public
String getRealName()
           this is used to

public
String getScope()
           Gets the scope (request or session) under which the associated bean is
           stored.

public
String getStyle()
           Gets the style of the rendered html tag.

public
String getStyleClass()
           Gets the style class of the rendered html tag.

public
String getTabIndex()

```



Gets the tabIndex of the rendered html tag.

```
public
String getTagId()
```

Return the ID of the form.

```
public
String getTagName()
```

Return the name of the Tag.

```
public
String getTarget()
```

Gets the window target.

```
public
String getType()
```

Gets the Java class name of the bean to be created, if necessary.

```
protected
void localRelease()
```

Release any acquired resources.

```
public
void setAction(String action)
```

Set the name of the action for the Form.

```
public
void setAttribute(String name, String value)
```

Set an attribute value.

```
public
void setEnctype(String enctype)
```

Set the content encoding to be used on a post submit.

```
public
void setFocus(String focus)
```

Set the name of the field to receive focus.

```
public
void setLocation(String location)
```

Set the location hash to append to the url.

```
public
void setMethod(String method)
```

Set the request method used when submitting this form.

```
public
void setName(String name)
```

Set the attribute key under which our associated bean is stored.

```
public
void setOnClick(String onclick)
```

Sets the onClick javascript event.

```
public
void setOnDbClick(String ondblclick)
```

Sets the onDbClick javascript event.

```
public
void setOnKeyDown(String onkeydown)
```

Sets the onKeyDown javascript event.

```

public setOnKeyPress(String onkeypress)
    void          Sets the onKeyPress javascript event.
public
    void setOnKeyUp(String onkeyup)
        Sets the onKeyUp javascript event.
public
    void setOnMouseDown(String onmousedown)
        Sets the onMouseDown javascript event.
public
    void setOnMouseMove(String onmousemove)
        Sets the onMouseMove javascript event.
public
    void setOnMouseOut(String onmouseout)
        Sets the onMouseOut javascript event.
public
    void setOnMouseOver(String onmouseover)
        Sets the onMouseOver javascript event.
public
    void setOnMouseUp(String onmouseup)
        Sets the onMouseUp javascript event.
public
    void setOnReset(String onReset)
        Sets the onReset javascript event.
public
    void setOnSubmit(String onSubmit)
        Sets the onSubmit javascript event.
public
    void setScope(String scope)
        Sets the scope (request or session) under which the associated bean is
        stored.
public
    void setStyle(String style)
        Sets the style of the rendered html tag.
public
    void setStyleClass(String styleClass)
        Sets the style class of the rendered html tag.
public
    void setTabIndex(String tabindex)
        Sets the tabIndex of the rendered html tag.
public
    void setTagId(String tagID)
        Set the ID of the form.
public
    void setTarget(String target)
        Sets the window target.
public
    void setType(String type)

```

## Page Flow API Reference

Sets the Java class name of the bean to be created, if necessary.

### Methods from `com.bea.wlw.netui.tags.AbstractBaseTag`

`addTagIdMapping`, `evaluateAttributeToString`, `filter`, `filter`,  
`getJavaScriptUtils`, `getNearestForm`, `getScriptReporter`, `getUserLocale`,  
`registerTagError`, `release`, `renderAttribute`, `rewriteName`, `write`

### Methods from `javax.servlet.jsp.tagext.BodyTagSupport`

`doInitBody`, `getBodyContent`, `getPreviousOut`, `release`, `setBodyContent`

### Methods from `javax.servlet.jsp.tagext.TagSupport`

`doAfterBody`, `doEndTag`, `doStartTag`, `findAncestorWithClass`, `getId`,  
`getParent`, `getValue`, `getValues`, `release`, `removeValue`, `setId`,  
`setPageContext`, `setParent`, `setValue`

### Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`,  
`toString`, `wait`, `wait`, `wait`

### Methods from interface `javax.servlet.jsp.tagext.BodyTag`

`doInitBody`, `setBodyContent`

### Methods from interface `com.bea.wlw.netui.tags.IAttributeConsumer`

`setAttribute`

### Methods from interface `javax.servlet.jsp.tagext.IterationTag`

`doAfterBody`

### Methods from interface `javax.servlet.jsp.tagext.Tag`

`doEndTag`, `doStartTag`, `getParent`, `release`, `setPageContext`, `setParent`

### Methods from interface `com.bea.wlw.netui.tags.html.URLParams`

`addParameter`

## Field Detail

### ACTION

```
public static final String ACTION
```

---

## beanScope

`protected String beanScope`

The scope of the form bean to (create and) use. This is either the same as the 'scope' attribute, if that was specified, or is obtained from the associated `ActionMapping` otherwise.

---

## beanType

`protected String beanType`

The type of the form bean to (create and) use. This is either the same as the 'type' attribute, if that was specified, or is obtained from the associated `ActionMapping` otherwise.

---

## enctype

`protected String enctype`

The content encoding to be used on a post submit.

---

## focus

`protected String focus`

The name of the field to receive focus, if any.

---

## location

`protected String location`

The location hash to append to the url.

---

## method

`protected String method`

The request method used when submitting this form.

---

## METHOD

`public static final String METHOD`

---

**name**

`protected String name`

The attribute key under which our associated bean is stored.

---

**scope**

`protected String scope`

The scope (request or session) under which our associated bean is stored.

---

**target**

`protected String target`

The window target.

---

**type**

`protected String type`

The Java class name of the bean to be created, if necessary.

## Constructor Detail

**Form**

`public Form()`

## Method Detail

**addParameter(String, Object) Method**

```
public void addParameter(String name,  
                        Object value)  
    throws JspException
```

Adds a URL parameter to the generated hyperlink.

**Parameters***name*

– the name of the parameter to be added.

*value*

– the value of the parameter to be added (a String or String[]).

**Exceptions***JspException*

---

**addTagID(String, String) Method**

```
public void addTagID(String tagID,
                    String name)
```

Adds a tagId and name to the Form's focusMap.

**Parameters***tagID*

– the tagID of a child tag.

*name*– the name of a child tag.

---

**doAfterBody() Method**

```
public int doAfterBody()
    throws JspException
```

Save the body content of the Form.

**Overrides**`BodyTagSupport.doAfterBody()`**Exceptions***JspException*if a JSP exception has occurred

---

**doEndTag() Method**

```
public int doEndTag()
    throws JspException
```

Render the end of this form.

**Overrides**

`BodyTagSupport.doEndTag()`

**Exceptions**

*JspException*

if a JSP exception has occurred

---

**doStartTag() Method**

```
public int doStartTag()  
    throws JspException
```

Render the beginning of this form.

**Overrides**

`BodyTagSupport.doStartTag()`

**Exceptions**

*JspException*

if a JSP exception has occurred

---

**generateRealName() Method**

```
public void generateRealName()
```

---

**getAction() Method**

```
public String getAction()
```

Return the action of the Form.

**Returns**

*a String representing the action name of the Form.*

---

**getBeanName() Method**

```
public String getBeanName()
```

Return the name of the form bean corresponding to this tag. There is no corresponding setter method; this method exists so that the nested tag classes can obtain the actual bean name derived from other attributes of the tag.

---

## getEnctype() Method

```
public String getEnctype()
```

Return the content encoding to be used on a post submit.

### Returns

*the content encoding type.*

---

## getFocus() Method

```
public String getFocus()
```

Return the name of the field to receive focus.

### Returns

*the focus field name*

---

## getLocation() Method

```
public String getLocation()
```

Return the location hash to append to the url.

### Returns

*the location hash*

---

## getMethod() Method

```
public String getMethod()
```

Return request method used when submitting this form.

### Returns

*the request method*

---

## getName() Method

```
public String getName()
```

Return the attribute key under which our associated bean is stored.



**Returns**

*the attribute key name*

---

**getOnClick() Method**

```
public String getOnClick()
```

Gets the onClick javascript event.

**Returns**

*the onClick event.*

---

**getOnDbClick() Method**

```
public String getOnDbClick()
```

Gets the onDbClick javascript event.

**Returns**

*the onDbClick event.*

---

**getOnKeyDown() Method**

```
public String getOnKeyDown()
```

Gets the onKeyDown javascript event.

**Returns**

*the onKeyDown event.*

---

**getOnKeyPress() Method**

```
public String getOnKeyPress()
```

Gets the onKeyPress javascript event.

**Returns**

*the onKeyPress event.*

---

## getOnKeyUp() Method

```
public String getOnKeyUp()
```

Gets the onKeyUp javascript event.

### Returns

*the onKeyUp event.*

---

## getOnMouseDown() Method

```
public String getOnMouseDown()
```

Gets the onMouseDown javascript event.

### Returns

*the onMouseDown event.*

---

## getOnMouseMove() Method

```
public String getOnMouseMove()
```

Gets the onMouseMove javascript event.

### Returns

*the onMouseMove event.*

---

## getOnMouseOut() Method

```
public String getOnMouseOut()
```

Gets the onMouseOut javascript event.

### Returns

*the onMouseOut event.*

---

## getOnMouseOver() Method

```
public String getOnMouseOver()
```

Gets the onMouseOver javascript event.

## Returns

*the onMouseOver event.*

---

## getOnMouseUp() Method

```
public String getOnMouseUp()
```

Gets the onMouseUp javascript event.

## Returns

*the onMouseUp event.*

---

## getOnReset() Method

```
public String getOnReset()
```

Gets the onReset javascript event.

## Returns

*the onReset event.*

---

## getOnSubmit() Method

```
public String getOnSubmit()
```

Gets the onSubmit javascript event.

## Returns

*the onSubmit event.*

---

## getRealName() Method

```
public String getRealName()
```

this is used to

---

## getScope() Method

```
public String getScope()
```

Gets the scope (request or session) under which the associated bean is stored.

## Returns

*the scope.*

---

## getStyle() Method

```
public String getStyle()
```

Gets the style of the rendered html tag.

## Returns

*the style.*

---

## getStyleClass() Method

```
public String getStyleClass()
```

Gets the style class of the rendered html tag.

## Returns

*the style class.*

---

## getTabIndex() Method

```
public String getTabIndex()
```

Gets the tabIndex of the rendered html tag.

## Returns

*the tabIndex.*

---

## getTagId() Method

```
public String getTagId()
```

Return the ID of the form.

## Returns

*the ID.*

---

## getTagName() Method

```
public String getTagName()
```

Return the name of the Tag.

### *Overrides*

```
AbstractBaseTag.getTagName()
```

---

## getTarget() Method

```
public String getTarget()
```

Gets the window target.

### **Returns**

*the window target.*

---

## getType() Method

```
public String getType()
```

Gets the Java class name of the bean to be created, if necessary.

### **Returns**

*the class name.*

---

## localRelease() Method

```
protected void localRelease()
```

Release any acquired resources.

### *Overrides*

```
AbstractBaseTag.localRelease()
```

---

## setAction(String) Method

```
public void setAction(String action)
```

Set the name of the action for the Form.

**Parameters***action*

– the name of the action to set for the Form.

**setAttribute(String, String) Method**

```
public void setAttribute(String name,
                        String value)
    throws JspException
```

Set an attribute value. The *name* represents the name of the attribute. The *value* represents the value and may contain a netui expression. This method may result in errors being generated. This requires that the tag buffer its body and write attributes in the end tag. For the form tag it is not legal to set the *id*, *name*, *action*, or *method* attributes with this method.

**Parameters***name*

The name of the attribute. This value may not be null or the empty string.

*value*

The value of the attribute. This may contain a netui expression.

**Exceptions***JspException*A *JspException* may be thrown if there is an error setting the attribute.**setEnctype(String) Method**

```
public void setEnctype(String enctype)
```

Set the content encoding to be used on a post submit.

**Parameters***enctype*

– the content encoding type.

**setFocus(String) Method**

```
public void setFocus(String focus)
```

Set the name of the field to receive focus.

## Parameters

*focus*

– the focus field name.

---

## setLocation(String) Method

```
public void setLocation(String location)
```

Set the location hash to append to the url.

## Parameters

*location*

– the location hash

---

## setMethod(String) Method

```
public void setMethod(String method)
```

Set the request method used when submitting this form.

## Parameters

*method*

– the request method

---

## setName(String) Method

```
public void setName(String name)
```

Set the attribute key under which our associated bean is stored.

## Parameters

*name*

– the attribute key name

---

## setOnClick(String) Method

```
public void setOnClick(String onclick)
```

Sets the onClick javascript event.

## Parameters

*onclick*

– the onClick event.

---

## setOnDbClick(String) Method

```
public void setOnDbClick(String ondblick)
```

Sets the onDbClick javascript event.

## Parameters

*ondblick*

– the onDbClick event.

---

## setOnKeyDown(String) Method

```
public void setOnKeyDown(String onkeydown)
```

Sets the onKeyDown javascript event.

## Parameters

*onkeydown*

– the onKeyDown event.

---

## setOnKeyPress(String) Method

```
public void setOnKeyPress(String onkeypress)
```

Sets the onKeyPress javascript event.

## Parameters

*onkeypress*

– the onKeyPress event.

---

## setOnKeyUp(String) Method

```
public void setOnKeyUp(String onkeyup)
```

Sets the onKeyUp javascript event.



## Parameters

*onkeyup*

– the onKeyUp event.

---

## setOnMouseDown(String) Method

```
public void setOnMouseDown(String onmousedown)
```

Sets the onMouseDown javascript event.

## Parameters

*onmousedown*

– the onMouseDown event.

---

## setOnMouseMove(String) Method

```
public void setOnMouseMove(String onmousemove)
```

Sets the onMouseMove javascript event.

## Parameters

*onmousemove*

– the onMouseMove event.

---

## setOnMouseOut(String) Method

```
public void setOnMouseOut(String onmouseout)
```

Sets the onMouseOut javascript event.

## Parameters

*onmouseout*

– the onMouseOut event.

---

## setOnMouseOver(String) Method

```
public void setOnMouseOver(String onmouseover)
```

Sets the onMouseOver javascript event.

## Parameters

*onmouseover*

– the onMouseOver event.

---

## setOnMouseUp(String) Method

```
public void setOnMouseUp(String onMouseup)
```

Sets the onMouseUp javascript event.

## Parameters

*onmouseup*

– the onMouseUp event.

---

## setOnReset(String) Method

```
public void setOnReset(String onReset)
```

Sets the onReset javascript event.

## Parameters

*onReset*

– the onReset event.

---

## setOnSubmit(String) Method

```
public void setOnSubmit(String onSubmit)
```

Sets the onSubmit javascript event.

## Parameters

*onSubmit*

– the onReset event.

---

## setScope(String) Method

```
public void setScope(String scope)
```

Sets the scope (request or session) under which the associated bean is stored.

## Parameters

*scope*

– the scope.

---

## setStyle(String) Method

```
public void setStyle(String style)
```

Sets the style of the rendered html tag.

## Parameters

*style*

– the html style.

---

## setStyleClass(String) Method

```
public void setStyleClass(String styleClass)
```

Sets the style class of the rendered html tag.

## Parameters

*styleClass*

– the html style class.

---

## setTabIndex(String) Method

```
public void setTabIndex(String tabIndex)
```

Sets the tabIndex of the rendered html tag.

## Parameters

*tabindex*

– the tab index.

---

## setTagId(String) Method

```
public void setTagId(String tagID)
```

Set the ID of the form.

**Parameters***tagID*

– the ID.

---

**setTarget(String) Method**

```
public void setTarget(String target)
```

Sets the window target.

**Parameters***target*

– the window target.

---

**setType(String) Method**

```
public void setType(String type)
```

Sets the Java class name of the bean to be created, if necessary.

**Parameters***type*

– the class name

# FormatDate Class

*public class* **FormatDate**

extends `FormatTag`

A formatter used to format dates. This formatter uses patterns that conform to `java.text.SimpleDateFormat` pattern syntax. Valid types for formatting are:

- `String`
- `java.sql.Date`
- `java.util.Date`
- `java.util.Calendar`

If the type is a `String`, we attempt to create a date out of the `String`. The `String` must be defined by a common format list below. If the string is equal the the empty string, it will be returned as the empty string. See `java.text.SimpleDateFormat` for more information.

The valid formats are:

- `MM/dd/yy`
- `yyyy-MM-dd`
- `MMddyy`
- and the local default

Attribute Descriptions			
Attribute	Required	Runtime Expression Evaluation	Data Bindable
<b>country</b>	No	No	No
	The country code used to determine the locale.		
<b>language</b>	No	No	No
	The language code used to determine the locale.		
<b>pattern</b>	No	No	Read Only
	The <code>SimpleDateFormat</code> pattern used to format this date.		
<b>stringInputPattern</b>	No	No	Read Only
	The pattern used to convert a <code>String</code> value into a date.		

## Related Topics

<netui:format...> Tag Samples  
[FormatNumber](#)

FormatString  
SimpleDateFormat

---

### ***Hierarchy***

```

Object
  TagSupport
    BodyTagSupport
      AbstractBaseTag
        FormatTag
          FormatDate

```

### ***All Implemented Interfaces***

BodyTag, IterationTag, Serializable, Tag

---

## **Field Summary**

### **Fields from `com.bea.wlw.netui.tags.html.FormatTag`**

pattern,

### **Fields from `com.bea.wlw.netui.tags.AbstractBaseTag`**

ATTR\_GENERAL, ATTR\_GENERAL\_EXPRESSION, ATTR\_JAVASCRIPT, ATTR\_STYLE, CLASS, ID, JAVASCRIPT\_STATUS, NAME, NETUI\_UNIQUE\_CNT, ONCLICK, ONDBLCLICK, ONKEYDOWN, ONKEYPRESS, ONKEYUP, ONMOUSEDOWN, ONMOUSEMOVE, ONMOUSEOUT, ONMOUSEOVER, ONMOUSEUP, STYLE, TABINDEX

### **Fields from `javax.servlet.jsp.tagext.BodyTagSupport`**

bodyContent

### **Fields from `javax.servlet.jsp.tagext.TagSupport`**

id, pageContext

### **Fields from interface `javax.servlet.jsp.tagext.BodyTag`**

EVAL\_BODY\_BUFFERED, EVAL\_BODY\_TAG

### **Fields from interface `javax.servlet.jsp.tagext.IterationTag`**

EVAL\_BODY\_AGAIN

### **Fields from interface `javax.servlet.jsp.tagext.Tag`**

EVAL\_BODY\_INCLUDE, EVAL\_PAGE, SKIP\_BODY, SKIP\_PAGE

## Constructor

### Summary

*FormatDate()*

## Method

### Summary

```

public
    int doStartTag()
        Create the internal Formatter instance and perform the
        formatting.

public
    String getStringInputPattern()

public
    String getTagName()
        Return the name of the Tag.

protected
    void localRelease()
        Release any acquired resources.

public
    void setStringInputPattern(String pattern)
        Set the pattern to use to convert a String value into a date.

```

#### Methods from `com.bea.wlw.netui.tags.html.FormatTag`

`getCountry`, `getLanguage`, `getLocale`, `getPattern`, `setCountry`, `setLanguage`, `setPattern`

#### Methods from `com.bea.wlw.netui.tags.AbstractBaseTag`

`addTagIdMapping`, `evaluateAttributeToString`, `filter`, `filter`, `getJavaScriptUtils`, `getNearestForm`, `getScriptReporter`, `getUserLocale`, `localRelease`, `registerTagError`, `release`, `renderAttribute`, `rewriteName`, `write`

#### Methods from `javax.servlet.jsp.tagext.BodyTagSupport`

`doAfterBody`, `doEndTag`, `doInitBody`, `doStartTag`, `getBodyContent`, `getPreviousOut`, `release`, `setBodyContent`

#### Methods from `javax.servlet.jsp.tagext.TagSupport`

`doAfterBody`, `doEndTag`, `doStartTag`, `findAncestorWithClass`, `getId`, `getParent`, `getValue`, `getValues`, `release`, `removeValue`, `setId`, `setPageContext`, `setParent`, `setValue`

**Methods from class `java.lang.Object`**

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`,  
`toString`, `wait`, `wait`, `wait`

**Methods from interface `javax.servlet.jsp.tagext.BodyTag`**

`doInitBody`, `setBodyContent`

**Methods from interface `javax.servlet.jsp.tagext.IterationTag`**

`doAfterBody`

**Methods from interface `javax.servlet.jsp.tagext.Tag`**

`doEndTag`, `doStartTag`, `getParent`, `release`, `setPageContext`, `setParent`

## Constructor Detail

### FormatDate

```
public FormatDate()
```

## Method Detail

### doStartTag() Method

```
public int doStartTag()
    throws JspException
```

Create the internal Formatter instance and perform the formatting.

***Overrides***

```
FormatTag.doStartTag()
```

**Exceptions**

*JspException*  
 if a JSP exception has occurred

---

### getStringInputPattern() Method

```
public String getStringInputPattern()
```

---



## getTagName() Method

```
public String getTagName()
```

Return the name of the Tag.

### *Overrides*

```
AbstractBaseTag.getTagName()
```

---

## localRelease() Method

```
protected void localRelease()
```

Release any acquired resources.

### *Overrides*

```
FormatTag.localRelease()
```

---

## setStringInputPattern(String) Method

```
public void setStringInputPattern(String pattern)
```

Set the pattern to use to convert a String value into a date. This will be used before the common formats.

### **Parameters**

*pattern*

the pattern representing the string input

# FormatNumber Class

*public class* **FormatNumber**

extends `FormatTag`

A formatter used to format numbers. This formatter uses patterns that conform to `java.text.NumberFormat` pattern syntax. `FormatNumber` calls `toString()` on the object to be formatted to get the value the pattern is applied to. The valid `FormatNumber` types are:

- number
- currency
- percent

Attribute Descriptions			
Attribute	Required	Runtime Expression Evaluation	Data Bindable
<b>country</b>	No	No	No
	The country code used to determine the locale.		
<b>language</b>	No	No	No
	The language code used to determine the locale.		
<b>pattern</b>	No	No	Read Only
	The <code>NumberFormat</code> pattern used to format this number.		
<b>type</b>	No	No	No
	The type of the format to apply (number, currency, or percent).		

## Related Topics

<netui:format...> Tag Samples  
 FormatDate  
 FormatString  
 NumberFormat

---

## Hierarchy

```

Object
  TagSupport
    BodyTagSupport
      AbstractBaseTag
        FormatTag
          FormatNumber
  
```

*All Implemented Interfaces*BodyTag, IterationTag, Serializable, Tag

---

## Field Summary

`protected String`***type***

The type of number format to be used.

**Fields from `com.bea.wlw.netui.tags.html.FormatTag`**`pattern,`**Fields from `com.bea.wlw.netui.tags.AbstractBaseTag`**`ATTR_GENERAL, ATTR_GENERAL_EXPRESSION, ATTR_JAVASCRIPT, ATTR_STYLE, CLASS, ID, JAVASCRIPT_STATUS, NAME, NETUI_UNIQUE_CNT, ONCLICK, ONDBLCLICK, ONKEYDOWN, ONKEYPRESS, ONKEYUP, ONMOUSEDOWN, ONMOUSEMOVE, ONMOUSEOUT, ONMOUSEOVER, ONMOUSEUP, STYLE, TABINDEX`**Fields from `javax.servlet.jsp.tagext.BodyTagSupport`**`bodyContent`**Fields from `javax.servlet.jsp.tagext.TagSupport`**`id, pageContext`**Fields from interface `javax.servlet.jsp.tagext.BodyTag`**`EVAL_BODY_BUFFERED, EVAL_BODY_TAG`**Fields from interface `javax.servlet.jsp.tagext.IterationTag`**`EVAL_BODY_AGAIN`**Fields from interface `javax.servlet.jsp.tagext.Tag`**`EVAL_BODY_INCLUDE, EVAL_PAGE, SKIP_BODY, SKIP_PAGE`

## Constructor Summary

***FormatNumber()***

## Method Summary

```

public
    int doStartTag()
        Create the internal Formatter instance and perform the formatting.

public
    String getTagName()
        Return the name of the Tag.

public
    String getType()
        Gets the type of number format to be used.

protected
    void localRelease()
        Release any acquired resources.

public
    void setType(String type)
        Sets the type of number format to be used (number, currency, or
        percent).

```

### Methods from `com.bea.wlw.netui.tags.html.FormatTag`

`getCountry`, `getLanguage`, `getLocale`, `getPattern`, `setCountry`, `setLanguage`, `setPattern`

### Methods from `com.bea.wlw.netui.tags.AbstractBaseTag`

`addTagIdMapping`, `evaluateAttributeToString`, `filter`, `filter`, `getJavaScriptUtils`, `getNearestForm`, `getScriptReporter`, `getUserLocale`, `localRelease`, `registerTagError`, `release`, `renderAttribute`, `rewriteName`, `write`

### Methods from `javax.servlet.jsp.tagext.BodyTagSupport`

`doAfterBody`, `doEndTag`, `doInitBody`, `doStartTag`, `getBodyContent`, `getPreviousOut`, `release`, `setBodyContent`

### Methods from `javax.servlet.jsp.tagext.TagSupport`

`doAfterBody`, `doEndTag`, `doStartTag`, `findAncestorWithClass`, `getId`, `getParent`, `getValue`, `getValues`, `release`, `removeValue`, `setId`, `setPageContext`, `setParent`, `setValue`

### Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

### Methods from interface `javax.servlet.jsp.tagext.BodyTag`

`doInitBody, setBodyContent`

#### Methods from interface `javax.servlet.jsp.tagext.IterationTag`

`doAfterBody`

#### Methods from interface `javax.servlet.jsp.tagext.Tag`

`doEndTag, doStartTag, getParent, release, setPageContext, setParent`

## Field Detail

### type

`protected String type`

The type of number format to be used.

## Constructor Detail

### FormatNumber

`public FormatNumber()`

## Method Detail

### doStartTag() Method

`public int doStartTag()`  
     throws `JspException`

Create the internal Formatter instance and perform the formatting.

#### *Overrides*

`FormatTag.doStartTag()`

### Exceptions

*JspException*  
     if a JSP exception has occurred

---

## getTagName() Method

```
public String getTagName()
```

Return the name of the Tag.

### *Overrides*

```
AbstractBaseTag.getTagName()
```

---

## getType() Method

```
public String getType()
```

Gets the type of number format to be used.

### **Returns**

*the type.*

---

## localRelease() Method

```
protected void localRelease()
```

Release any acquired resources.

### *Overrides*

```
FormatTag.localRelease()
```

---

## setType(String) Method

```
public void setType(String type)
```

Sets the type of number format to be used (number, currency, or percent).

### **Parameters**

*type*

– the number format type.

# FormatString Class

*public class* **FormatString**

extends **FormatTag**

A formatter used to format strings. FormatString uses the following pattern syntax:

The '#' character gets replaced by the next character in the string getting formatted, while other characters get put in as literals. For example:

String "5555555555" with pattern "(###)###-####" would result in: (555)555-5555.

The '\*' character will display all characters in the string at that point in the pattern. For example:

String "123456" with pattern "#-\*!" would result in: 1-23456!

If a result with a '#' or '\*' character showing is desired, the '#' or '\*' needs to be escaped with the '\$' character. For example:

String "ABCD" with pattern "\$#-####" would result in: #-ABCD.

To show a '\$' in the result, the '\$' character needs to be escaped. For example:

String "1234" with pattern "\$\$#,###" would result in: \$1,234

If the truncate attribute is set to "true", characters in the string that exceed the pattern will be dropped. Otherwise, they will be appended to the end of the formatted string.

Attribute Descriptions			
Attribute	Required	Runtime Expression Evaluation	Data Bindable
country	No	No	No
	The country code used to determine the locale		
language	No	No	No
	The language code used to determine the locale		
pattern	No	No	Read Only
	The pattern used to format this string		
truncate	No	No	No
	A boolean specifying whether characters that exceed the pattern's length should be dropped.		

**Related Topics**

<netui:format...> Tag Samples

FormatDate

FormatNumber

---

***Hierarchy***

```

Object
  TagSupport
    BodyTagSupport
      AbstractBaseTag
        FormatTag
          FormatString

```

***All Implemented Interfaces***

BodyTag, IterationTag, Serializable, Tag

---

## Field Summary

protected boolean

***truncate***

Whether or not pattern-exceeding characters should be dropped.

**Fields from `com.bea.wlw.netui.tags.html.FormatTag`**

pattern,

**Fields from `com.bea.wlw.netui.tags.AbstractBaseTag`**

ATTR\_GENERAL, ATTR\_GENERAL\_EXPRESSION, ATTR\_JAVASCRIPT, ATTR\_STYLE, CLASS, ID, JAVASCRIPT\_STATUS, NAME, NETUI\_UNIQUE\_CNT, ONCLICK, ONDBLCLICK, ONKEYDOWN, ONKEYPRESS, ONKEYUP, ONMOUSEDOWN, ONMOUSEMOVE, ONMOUSEOUT, ONMOUSEOVER, ONMOUSEUP, STYLE, TABINDEX

**Fields from `javax.servlet.jsp.tagext.BodyTagSupport`**

bodyContent

**Fields from `javax.servlet.jsp.tagext.TagSupport`**

id, pageContext

**Fields from interface `javax.servlet.jsp.tagext.BodyTag`**

EVAL\_BODY\_BUFFERED, EVAL\_BODY\_TAG



**Fields from interface javax.servlet.jsp.tagext.IterationTag**

EVAL\_BODY\_AGAIN

**Fields from interface javax.servlet.jsp.tagext.Tag**

EVAL\_BODY\_INCLUDE, EVAL\_PAGE, SKIP\_BODY, SKIP\_PAGE

## Constructor Summary

*FormatString()*

## Method Summary

```

    public
        int doStartTag()
            Create the internal Formatter instance and perform the formatting.

    public
        String getTagName()
            Return the name of the Tag.

    public
        boolean getTruncate()
            Gets whether or not pattern-exceeding characters should be
            dropped.

protected
        void localRelease()
            Release any acquired resources.

    public
        void setTruncate(boolean truncate)
            Sets whether or not pattern-exceeding characters should be
            dropped.

```

**Methods from com.bea.wlw.netui.tags.html.FormatTag**

getCountry, getLanguage, getLocale, getPattern, setCountry, setLanguage, setPattern

**Methods from com.bea.wlw.netui.tags.AbstractBaseTag**

addTagIdMapping, evaluateAttributeToString, filter, filter, getJavaScriptUtils, getNearestForm, getScriptReporter, getUserLocale, localRelease, registerTagError, release, renderAttribute, rewriteName, write

**Methods from javax.servlet.jsp.tagext.BodyTagSupport**

`doAfterBody`, `doEndTag`, `doInitBody`, `doStartTag`, `getBodyContent`,  
`getPreviousOut`, `release`, `setBodyContent`

### Methods from `javax.servlet.jsp.tagext.TagSupport`

`doAfterBody`, `doEndTag`, `doStartTag`, `findAncestorWithClass`, `getId`,  
`getParent`, `getValue`, `getValues`, `release`, `removeValue`, `setId`,  
`setPageContext`, `setParent`, `setValue`

### Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`,  
`toString`, `wait`, `wait`, `wait`

### Methods from interface `javax.servlet.jsp.tagext.BodyTag`

`doInitBody`, `setBodyContent`

### Methods from interface `javax.servlet.jsp.tagext.IterationTag`

`doAfterBody`

### Methods from interface `javax.servlet.jsp.tagext.Tag`

`doEndTag`, `doStartTag`, `getParent`, `release`, `setPageContext`, `setParent`

## Field Detail

### `truncate`

`protected boolean truncate`

Whether or not pattern-exceeding characters should be dropped.

## Constructor Detail

### `FormatString`

`public FormatString()`

## Method Detail

## doStartTag() Method

```
public int doStartTag()  
    throws JspException
```

Create the internal Formatter instance and perform the formatting.

### Overrides

```
FormatTag.doStartTag()
```

## Exceptions

*JspException*

if a JSP exception has occurred

---

## getTagName() Method

```
public String getTagName()
```

Return the name of the Tag.

### Overrides

```
AbstractBaseTag.getTagName()
```

---

## getTruncate() Method

```
public boolean getTruncate()
```

Gets whether or not pattern-exceeding characters should be dropped.

## Returns

*the truncate value ("true" or "false").*

---

## localRelease() Method

```
protected void localRelease()
```

Release any acquired resources.

### Overrides

```
FormatTag.localRelease()
```

---

## setTruncate(boolean) Method

```
public void setTruncate(boolean truncate)
```

Sets whether or not pattern-exceeding characters should be dropped.

### Parameters

*truncate*

– "true" or "false"

# Formattable Interface

*public interface Formattable*

Interface that indicates whether or not a tag's output can be formatted with by a `FormatTag`.

---

## *All Known Implementing Classes*

`Label`, `Select`, `TextArea`, `TextBox`, `FormattableColumn`

---

## Method

### Summary

```
public
    void addFormatter(FormatTag.Formatter formatter)
        Adds a FormatTag.Formatter instance to the tag's set of
        formatters.
```

## Method Detail

### **addFormatter(FormatTag.Formatter) Method**

```
public void addFormatter(FormatTag.Formatter formatter)
```

Adds a `FormatTag.Formatter` instance to the tag's set of formatters.

### Parameters

*formatter*

- the formatter instance to add to the `Formattable` tag's set of formatters.

# FormatTag Class

*public abstract class FormatTag*

extends AbstractBaseTag

Abstract base class for formatting tags. Provides the basic formatting properties, as well as the base for the internal FormatTag.Formatter class.

---

## **Hierarchy**

Object  
  TagSupport  
    BodyTagSupport  
      AbstractBaseTag  
        FormatTag

## **All Implemented Interfaces**

BodyTag, IterationTag, Serializable, Tag

## **Direct Known Subclasses**

FormatDate, FormatNumber, FormatString

---

## Field Summary

protected String

### ***pattern***

The pattern used by a FormatTag to do its formatting.

### **Fields from com.bea.wlw.netui.tags.AbstractBaseTag**

ATTR\_GENERAL, ATTR\_GENERAL\_EXPRESSION, ATTR\_JAVASCRIPT, ATTR\_STYLE, CLASS, ID, JAVASCRIPT\_STATUS, NAME, NETUI\_UNIQUE\_CNT, ONCLICK, ONDBLCLICK, ONKEYDOWN, ONKEYPRESS, ONKEYUP, ONMOUSEDOWN, ONMOUSEMOVE, ONMOUSEOUT, ONMOUSEOVER, ONMOUSEUP, STYLE, TABINDEX

### **Fields from javax.servlet.jsp.tagext.BodyTagSupport**

bodyContent

### **Fields from javax.servlet.jsp.tagext.TagSupport**

id, pageContext

**Fields from interface javax.servlet.jsp.tagext.BodyTag**

EVAL\_BODY\_BUFFERED, EVAL\_BODY\_TAG

**Fields from interface javax.servlet.jsp.tagext.IterationTag**

EVAL\_BODY\_AGAIN

**Fields from interface javax.servlet.jsp.tagext.Tag**

EVAL\_BODY\_INCLUDE, EVAL\_PAGE, SKIP\_BODY, SKIP\_PAGE

## Constructor Summary

*FormatTag()*

## Method Summary

```

    public
        int doStartTag()
            Create the internal Formatter instance and perform the
            formatting.

    public
        String getCountry()
            Gets the country code for the locale.

    public
        String getLanguage()
            Gets the language code for the locale.

    public
        Locale getLocale()
            Returns the locale based on the country and language.

    public
        String getPattern()
            Gets the pattern to be used by this FormatTag.

protected
        void localRelease()
            Release any acquired resources.

    public
        void setCountry(String country)
            Sets the country code for the locale.

    public
        void setLanguage(String language)
            Sets the language code for the locale.

```

```
public setPattern(String pattern)
void           Sets the pattern to be used by this FormatTag.
```

#### Methods from `com.bea.wlw.netui.tags.AbstractBaseTag`

```
addTagIdMapping, evaluateAttributeToString, filter, filter,
getJavaScriptUtils, getNearestForm, getScriptReporter, getTagName,
getUserLocale, registerTagError, release, renderAttribute, rewriteName,
write
```

#### Methods from `javax.servlet.jsp.tagext.BodyTagSupport`

```
doAfterBody, doEndTag, doInitBody, getBodyContent, getPreviousOut,
release, setBodyContent
```

#### Methods from `javax.servlet.jsp.tagext.TagSupport`

```
doAfterBody, doEndTag, doStartTag, findAncestorWithClass, getId,
getParent, getValue, getValues, release, removeValue, setId,
setPageContext, setParent, setValue
```

#### Methods from class `java.lang.Object`

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll,
toString, wait, wait, wait
```

#### Methods from interface `javax.servlet.jsp.tagext.BodyTag`

```
doInitBody, setBodyContent
```

#### Methods from interface `javax.servlet.jsp.tagext.IterationTag`

```
doAfterBody
```

#### Methods from interface `javax.servlet.jsp.tagext.Tag`

```
doEndTag, doStartTag, getParent, release, setPageContext, setParent
```

## Field Detail

### **pattern**

```
protected String pattern
```

The pattern used by a FormatTag to do its formatting.



## Constructor Detail

### FormatTag

```
public FormatTag()
```

## Method Detail

### doStartTag() Method

```
public int doStartTag()  
    throws JspException
```

Create the internal Formatter instance and perform the formatting.

#### *Overrides*

```
BodyTagSupport.doStartTag()
```

### Exceptions

*JspException*  
if a JSP exception has occurred

---

### getCountry() Method

```
public String getCountry()
```

Gets the country code for the locale.

### Returns

*the country code.*

---

### getLanguage() Method

```
public String getLanguage()
```

Gets the language code for the locale.

### Returns

*the language code*

---

## getLocale() Method

```
public Locale getLocale()
```

Returns the locale based on the country and language.

### Returns

*the locale*

---

## getPattern() Method

```
public String getPattern()
```

Gets the pattern to be used by this FormatTag.

### Returns

*the pattern.*

---

## localRelease() Method

```
protected void localRelease()
```

Release any acquired resources.

### Overrides

```
AbstractBaseTag.localRelease()
```

---

## setCountry(String) Method

```
public void setCountry(String country)
```

Sets the country code for the locale.

### Parameters

*country*  
– the country code

---

## setLanguage(String) Method

```
public void setLanguage(String language)
```

Sets the language code for the locale.

## Parameters

*language*

– the language code

---

## setPattern(String) Method

```
public void setPattern(String pattern)
```

Sets the pattern to be used by this FormatTag.

## Parameters

*pattern*

– the pattern to be used

# GetNetuiTagName Class

*public class* **GetNetuiTagName**

extends `AbstractBaseTag`

Returns the value of the `tagId` attribute. This tag allows you to track the value of the `tagId` attribute, the value of which will be re-written if the web application is part of a Portal application. This tag is especially useful when you need to pass the `tagId` value to an HTML tag. In the following example, the `tagId` value is passed from the tag `<netui:radioButtonOption>` to the HTML tag `<label>`.

```
<netui:radioButtonOption tagId="radio1" value="value1"/>
<label for="<netui:getNetuiTagName tagId="radio1"/>">Display Text 1</label>
```

If the `tagId` attribute in the tag is re-written, the `<label>` can access the re-written value through the tag `<netui:getNetuiTagName>`.

## Related Topics

Using JavaScript in Page Flow and Portal Applications

---

## Hierarchy

```
Object
  TagSupport
    BodyTagSupport
      AbstractBaseTag
        GetNetuiTagName
```

## All Implemented Interfaces

`BodyTag`, `IterationTag`, `Serializable`, `Tag`

---

## Field Summary

### Fields from `com.bea.wlw.netui.tags.AbstractBaseTag`

`ATTR_GENERAL`, `ATTR_GENERAL_EXPRESSION`, `ATTR_JAVASCRIPT`, `ATTR_STYLE`, `CLASS`, `ID`, `JAVASCRIPT_STATUS`, `NAME`, `NETUI_UNIQUE_CNT`, `ONCLICK`, `ONDBLCLICK`, `ONKEYDOWN`, `ONKEYPRESS`, `ONKEYUP`, `ONMOUSEDOWN`, `ONMOUSEMOVE`, `ONMOUSEOUT`, `ONMOUSEOVER`, `ONMOUSEUP`, `STYLE`, `TABINDEX`

### Fields from `javax.servlet.jsp.tagext.BodyTagSupport`

`bodyContent`

**Fields from `javax.servlet.jsp.tagext.TagSupport`**`id, pageContext`**Fields from interface `javax.servlet.jsp.tagext.BodyTag`**`EVAL_BODY_BUFFERED, EVAL_BODY_TAG`**Fields from interface `javax.servlet.jsp.tagext.IterationTag`**`EVAL_BODY_AGAIN`**Fields from interface `javax.servlet.jsp.tagext.Tag`**`EVAL_BODY_INCLUDE, EVAL_PAGE, SKIP_BODY, SKIP_PAGE`

## Constructor Summary

*`GetNetuiTagName()`*

## Method Summary

```

    public
        int doStartTag()

    public
        String getTagId()

    public
        String getTagName()
            Return the name of the Tag.

protected
    void localRelease()
        Release any acquired resources.

    public
        void setTagId(String tagId)
            Return the value of this tagId
            attribute.

```

**Methods from `com.bea.wlw.netui.tags.AbstractBaseTag`**

```

addTagIdMapping, evaluateAttributeToString, filter, filter,
getJavaScriptUtils, getNearestForm, getScriptReporter, getUserLocale,
registerTagError, release, renderAttribute, rewriteName, write

```

**Methods from `javax.servlet.jsp.tagext.BodyTagSupport`**

`doAfterBody`, `doEndTag`, `doInitBody`, `getBodyContent`, `getPreviousOut`,  
`release`, `setBodyContent`

#### Methods from `javax.servlet.jsp.tagext.TagSupport`

`doAfterBody`, `doEndTag`, `doStartTag`, `findAncestorWithClass`, `getId`,  
`getParent`, `getValue`, `getValues`, `release`, `removeValue`, `setId`,  
`setPageContext`, `setParent`, `setValue`

#### Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`,  
`toString`, `wait`, `wait`, `wait`

#### Methods from interface `javax.servlet.jsp.tagext.BodyTag`

`doInitBody`, `setBodyContent`

#### Methods from interface `javax.servlet.jsp.tagext.IterationTag`

`doAfterBody`

#### Methods from interface `javax.servlet.jsp.tagext.Tag`

`doEndTag`, `doStartTag`, `getParent`, `release`, `setPageContext`, `setParent`

## Constructor Detail

### GetNetuiTagName

```
public GetNetuiTagName()
```

## Method Detail

### doStartTag() Method

```
public int doStartTag()  
    throws JspException
```

#### *Overrides*

```
BodyTagSupport.doStartTag()
```

### Exceptions

*JspException*

---

## getTagId() Method

```
public String getTagId()
```

---

## getTagName() Method

```
public String getTagName()
```

Return the name of the Tag.

### *Overrides*

```
AbstractBaseTag.getTagName()
```

---

## localRelease() Method

```
protected void localRelease()
```

Release any acquired resources.

### *Overrides*

```
AbstractBaseTag.localRelease()
```

---

## setTagId(String) Method

```
public void setTagId(String tagId)
```

Return the value of this tagId attribute.

# Hidden Class

*public class* **Hidden**

extends DataSourceTag  
implements IAttributeConsumer

Generates a hidden tag with a given value.

Attribute Descriptions			
Attribute	Required	Runtime Expression Evaluation	Data Bindable
<b>dataInput</b>	No	No	Read Only
	The dataInput attribute overrides the dataSource for the input of the value. It allows a hidden field to read it's value from one place and return the value to a new destination. The dataInput attribute must be a data binding expression enclosed in curly braces, for example, {pageFlow.hiddenValue}		
<b>dataSource</b>	Yes	No	Read / Write
	The name of a form bean property or an expression that resolves to a bean property.		
<b>tagId</b>	No	No	No
	The ID of this button used by the enclosing html tag to get the tag's real ID attribute for JavaScript and focus purposes. The real ID attribute id is generated based upon this name.		

## Hierarchy

```

Object
  TagSupport
    BodyTagSupport
      AbstractBaseTag
        DataSourceTag
          Hidden
  
```

## All Implemented Interfaces

BodyTag, IAttributeConsumer, IterationTag, Serializable, Tag



## Field Summary

```

    public
    static TYPE
final String      String

    public
    static VALUE
final String      String

```

### Fields from `com.bea.wlw.netui.tags.AbstractBaseTag`

ATTR\_GENERAL, ATTR\_GENERAL\_EXPRESSION, ATTR\_JAVASCRIPT, ATTR\_STYLE, CLASS, ID, JAVASCRIPT\_STATUS, NAME, NETUI\_UNIQUE\_CNT, ONCLICK, ONDBLCLICK, ONKEYDOWN, ONKEYPRESS, ONKEYUP, ONMOUSEDOWN, ONMOUSEMOVE, ONMOUSEOUT, ONMOUSEOVER, ONMOUSEUP, STYLE, TABINDEX

### Fields from `javax.servlet.jsp.tagext.BodyTagSupport`

bodyContent

### Fields from `javax.servlet.jsp.tagext.TagSupport`

id, pageContext

### Fields from interface `javax.servlet.jsp.tagext.BodyTag`

EVAL\_BODY\_BUFFERED, EVAL\_BODY\_TAG

### Fields from interface `javax.servlet.jsp.tagext.IterationTag`

EVAL\_BODY\_AGAIN

### Fields from interface `javax.servlet.jsp.tagext.Tag`

EVAL\_BODY\_INCLUDE, EVAL\_PAGE, SKIP\_BODY, SKIP\_PAGE

## Constructor Summary

*Hidden()*

## Method Summary

```

    public
    int doAfterBody()

```

```

public doEndTag()
    int          Generate the hidden input tag.
public
    int doStartTag()
public
String getDataInput()
    The value of the dataInput attribute.
public
String getTabIndex()
    Gets the tabIndex of the rendered html tag.
public
String getTagId()
    Return the ID of the hidden.
public
String getTagName()
    Return the name of the Tag.
protected
    void localRelease()
        Release any acquired resources.
public
    void setAttribute(String name, String value)
        Set an attribute value.
public
    void setDataInput(String dataInput)
        Set the data input.
public
    void setTabIndex(String tabIndex)
        Sets the tabIndex of the rendered html tag.
public
    void setTagId(String tagId)
        Set the ID of the hidden.

```

#### Methods from **com.bea.wlw.netui.tags.DataSourceTag**

getDataSource, setDataSource

#### Methods from **com.bea.wlw.netui.tags.AbstractBaseTag**

addTagIdMapping, evaluateAttributeToString, filter, filter,  
getJavaScriptUtils, getNearestForm, getScriptReporter, getUserLocale,  
localRelease, registerTagError, release, renderAttribute, rewriteName,  
write

#### Methods from **javax.servlet.jsp.tagext.BodyTagSupport**

doInitBody, getBodyContent, getPreviousOut, release, setBodyContent

#### Methods from **javax.servlet.jsp.tagext.TagSupport**

`doAfterBody`, `doEndTag`, `doStartTag`, `findAncestorWithClass`, `getId`,  
`getParent`, `getValue`, `getValues`, `release`, `removeValue`, `setId`,  
`setPageContext`, `setParent`, `setValue`

### Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`,  
`toString`, `wait`, `wait`, `wait`

### Methods from interface `javax.servlet.jsp.tagext.BodyTag`

`doInitBody`, `setBodyContent`

### Methods from interface `com.bea.wlw.netui.tags.IAttributeConsumer`

`setAttribute`

### Methods from interface `javax.servlet.jsp.tagext.IterationTag`

`doAfterBody`

### Methods from interface `javax.servlet.jsp.tagext.Tag`

`doEndTag`, `doStartTag`, `getParent`, `release`, `setPageContext`, `setParent`

## Field Detail

### TYPE

```
public static final String TYPE
```

---

### VALUE

```
public static final String VALUE
```

## Constructor Detail

### Hidden

```
public Hidden()
```

## Method Detail

### doAfterBody() Method

```
public int doAfterBody()  
    throws JspException
```

#### *Overrides*

```
BodyTagSupport.doAfterBody()
```

#### Exceptions

*JspException*  
if a JSP exception has occurred

---

### doEndTag() Method

```
public int doEndTag()  
    throws JspException
```

Generate the hidden input tag.

#### *Overrides*

```
BodyTagSupport.doEndTag()
```

#### Exceptions

*JspException*  
if a JSP exception has occurred

---

### doStartTag() Method

```
public int doStartTag()  
    throws JspException
```

#### *Overrides*

```
BodyTagSupport.doStartTag()
```

#### Exceptions

*JspException*  
if a JSP exception has occurred

---

### getDataInput() Method

```
public String getDataInput()
```

The value of the dataInput attribute.

### Returns

*the current value of the data input attribute.*

---

## getTabIndex() Method

```
public String getTabIndex()
```

Gets the tabIndex of the rendered html tag.

### Returns

*the tabindex.*

---

## getTagId() Method

```
public String getTagId()
```

Return the ID of the hidden.

### Returns

*the ID.*

---

## getTagName() Method

```
public String getTagName()
```

Return the name of the Tag.

### Overrides

```
AbstractBaseTag.getTagName()
```

---

## localRelease() Method

```
protected void localRelease()
```

Release any acquired resources.

### Overrides

```
DataSourceTag.localRelease()
```

---

## setAttribute(String, String) Method

```
public void setAttribute(String name,
                        String value)
    throws JspException
```

Set an attribute value. The name represents the name of the attribute. The value represents the value and may contain a netui expression. This method may result in errors being generated. This requires that the tag buffer its body and write attributes in the end tag. For the hidden tag it is not legal to set the id, name, type, or value attributes with this method.

### Parameters

*name*

The name of the attribute. This value may not be null or the empty string.

*value*

The value of the attribute. This may contain a netui expression.

### Exceptions

*JspException*

A JspException may be thrown if there is an error setting the attribute.

---

## setDataInput(String) Method

```
public void setDataInput(String dataInput)
```

Set the data input. This value will override the `dataSource` and provide the input value on the select box.

### Parameters

*dataInput*

the value of the input to the page. This value may contain an expression.

---

## setTabIndex(String) Method

```
public void setTabIndex(String tabIndex)
```

Sets the `tabIndex` of the rendered html tag.

### Parameters

*tabindex*

- the tab index.

## **setTagId(String) Method**

```
public void setTagId(String tagId)
```

Set the ID of the hidden.

### **Parameters**

*tagId*  
- the ID.

# Html Class

*public class* **Html**

extends `ScriptContainer`  
implements `ErrorReporter`

Generates the html element and performs error handling within its body.

Attribute Descriptions			
Attribute	Required	Runtime Expression Evaluation	Data Bindable
<b>dir</b>	No	No	No
	The direction of the HTML, left-to-right (LTR) or right-to-left (RTL).		
<b>useLocale</b>	No	No	No
	A boolean specifying whether the html output should include <code>xml:lang=defaultLocale.getLanguage()</code> .		

## Hierarchy

```

Object
  TagSupport
    BodyTagSupport
      AbstractBaseTag
        ScriptContainer
          Html
  
```

## All Implemented Interfaces

BodyTag, ErrorReporter, IterationTag, ScriptReporter, Serializable, Tag

## Field Summary

```

public
static HTML_TAG_ID
final String      String
  
```

## Fields from com.bea.wlw.netui.tags.AbstractBaseTag

ATTR\_GENERAL, ATTR\_GENERAL\_EXPRESSION, ATTR\_JAVASCRIPT, ATTR\_STYLE, CLASS, ID, JAVASCRIPT\_STATUS, NAME, NETUI\_UNIQUE\_CNT, ONCLICK, ONDBLCLICK, ONKEYDOWN, ONKEYPRESS, ONKEYUP, ONMOUSEDOWN, ONMOUSEMOVE,



ONMOUSEOUT, ONMOUSEOVER, ONMOUSEUP, STYLE, TABINDEX

**Fields from `javax.servlet.jsp.tagext.BodyTagSupport`**

`bodyContent`

**Fields from `javax.servlet.jsp.tagext.TagSupport`**

`id`, `pageContext`

**Fields from interface `javax.servlet.jsp.tagext.BodyTag`**

`EVAL_BODY_BUFFERED`, `EVAL_BODY_TAG`

**Fields from interface `com.bea.wlw.netui.tags.ErrorReporter`**

`CONTAINER_ERRORS`

**Fields from interface `javax.servlet.jsp.tagext.IterationTag`**

`EVAL_BODY_AGAIN`

**Fields from interface `javax.servlet.jsp.tagext.Tag`**

`EVAL_BODY_INCLUDE`, `EVAL_PAGE`, `SKIP_BODY`, `SKIP_PAGE`

## Constructor Summary

***Html()***

## Method Summary

```
public
    void addError(AbstractPageError ape)
        Add an error to the errors being reported by this tag.

public
    int doAfterBody()
        Save any body content of this tag, which will generally be the option(s) representing
        the values displayed to the user.

public
    int doEndTag()
        Write out the body content and report any errors that occurred.

public
    int doStartTag()
```

```

public getDir()
String      Gets the dir value of the html.

public
    int getNextId()
                This method will return the next unique integer within the scope of this tag.

public
String getTagName()
                Returns the name of the Tag.

public
boolean getUseLocale()
                Gets whether the default locale's language should be used.

public
boolean isReporting()
                This boolean indicates if an ErrorReporter is reporting errors or not.

protected
    void localRelease()
                Release any acquired resources.

public
ArrayList returnErrors()
                Return an ArrayList of the errors

public
    void setDir(String dir)
                Sets the dir value of the html.

public
    void setUseLocale(boolean locale)
                Sets whether the default locale's language should be used.

```

#### Methods from **com.bea.wlw.netui.tags.html.ScriptContainer**

```

addScriptCode, addScriptFunction, addTagId, getScopeId, nextTagIdRef,
processIdMap, replaceTagNameRefs, resolveTagIdRef, setScopeId,
writeScriptBlock

```

#### Methods from **com.bea.wlw.netui.tags.AbstractBaseTag**

```

addTagIdMapping, evaluateAttributeToString, filter, filter,
getJavaScriptUtils, getNearestForm, getScriptReporter, getTagName,
getUserLocale, localRelease, registerTagError, release, renderAttribute,
rewriteName, write

```

#### Methods from **javax.servlet.jsp.tagext.BodyTagSupport**

```

doAfterBody, doEndTag, doInitBody, doStartTag, getBodyContent,
getPreviousOut, release, setBodyContent

```

#### Methods from **javax.servlet.jsp.tagext.TagSupport**

```

doAfterBody, doEndTag, doStartTag, findAncestorWithClass, getId,
getParent, getValue, getValues, release, removeValue, setId,

```

`setPageContext, setParent, setValue`

#### Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

#### Methods from interface `javax.servlet.jsp.tagext.BodyTag`

`doInitBody, setBodyContent`

#### Methods from interface `com.bea.wlw.netui.tags.ErrorReporter`

`addError, isReporting, returnErrors`

#### Methods from interface `javax.servlet.jsp.tagext.IterationTag`

`doAfterBody`

#### Methods from interface `com.bea.wlw.netui.tags.ScriptReporter`

`addScriptCode, addScriptFunction, addTagId`

#### Methods from interface `javax.servlet.jsp.tagext.Tag`

`doEndTag, doStartTag, getParent, release, setPageContext, setParent`

## Field Detail

### HTML\_TAG\_ID

```
public static final String HTML_TAG_ID
```

## Constructor Detail

### Html

```
public Html()
```

## Method Detail

## addError(AbstractPageError) Method

```
public void addError(AbstractPageError ape)
```

Add an error to the errors being reported by this tag.

### Parameters

*ape*

– The AbstractPageError to add

---

## doAfterBody() Method

```
public int doAfterBody()  
    throws JspException
```

Save any body content of this tag, which will generally be the option(s) representing the values displayed to the user.

### Overrides

```
ScriptContainer.doAfterBody()
```

### Exceptions

*JspException*

if a JSP exception has occurred

---

## doEndTag() Method

```
public int doEndTag()  
    throws JspException
```

Write out the body content and report any errors that occurred.

### Overrides

```
ScriptContainer.doEndTag()
```

### Exceptions

*JspException*

if a JSP exception has occurred

---

## doStartTag() Method

```
public int doStartTag()  
    throws JspException
```

### Overrides

```
ScriptContainer.doStartTag()
```

## Exceptions

*JspException*

---

## getDir() Method

```
public String getDir()
```

Gets the dir value of the html.

## Returns

*"LTR" or "RTL"*

---

## getNextId() Method

```
public int getNextId()
```

This method will return the next unique integer within the scope of this tag.

## Returns

*a unique integer*

---

## getTagName() Method

```
public String getTagName()
```

Returns the name of the Tag.

## Overrides

```
ScriptContainer.getTagName()
```

---

## getUseLocale() Method

```
public boolean getUseLocale()
```

Gets whether the default locale's language should be used.

## Returns

*true or false*

---

## isReporting() Method

```
public boolean isReporting()
```

This boolean indicates if an `ErrorReporter` is reporting errors or not. The caller should check this before calling `addError` because the `ErrorReporter` may be off for some reason.

### Returns

*a boolean indicating if the tag is reporting errors or not.*

---

## localRelease() Method

```
protected void localRelease()
```

Release any acquired resources.

### Overrides

```
ScriptContainer.localRelease()
```

---

## returnErrors() Method

```
public ArrayList returnErrors()
```

Return an `ArrayList` of the errors

### Returns

*an `ArrayList` of all errors.*

---

## setDir(String) Method

```
public void setDir(String dir)
```

Sets the `dir` value of the `html`.

### Parameters

*dir*

– "LTR" or "RTL"

---

## setUseLocale(boolean) Method

```
public void setUseLocale(boolean locale)
```

Sets whether the default locale's language should be used.

## Parameters

*locale*

– true or false

# Image Class

*public class Image*

extends AbstractBaseTag  
implements IAttributeConsumer, URLParams

Generates an image with the specified attributes. Image ignores its body content.

Attribute Descriptions			
Attribute	Required	Runtime Expression Evaluation	Data Bindable
align	No	No	No
	Property to specify where to align the image.		
alt	No	No	Read Only
	Property to specify alt text of an image.		
border	No	No	No
	The border size around the image.		
height	No	No	Read Only
	The image height.		
hspace	No	No	Read Only
	The horizontal spacing around the image.		
id	No	No	No
	<b>The id attribute has been deprecated;</b> instead, use tagId. The id attribute of the generated Image. You should use the tagId which will encode the id correctly for containers such as the Portal which needs to rewrite id's to make them unique.		
isMap	No	No	No
	The server-side image map declaration.		
location	No	No	No
	The location hash to append to the url.		
lowsrc	No	No	No
	The low resolution image source URI.		
page	No	No	No



## Page Flow API Reference

	<b>The <code>page</code> attribute has been deprecated.</b> The module–relative path of the image to be displayed by this rendered tag.		
<b>src</b>	No	No	Read Only
	The image source URI.		
<b>tagId</b>	No	No	No
	The id of this image. The real id written to the HTML stream may be changed by the container (Portal), but may be looked up with the <code>tagId</code> value through the JavaScript method <code>getNetuiTagName( tagId, tag )</code> method written by the <code>&lt;netui:html&gt;</code> tag.		
<b>useMap</b>	No	No	No
	The client–side image map declaration.		
<b>vspace</b>	No	No	Read Only
	The vertical spacing around the image.		
<b>width</b>	No	No	Read Only
	The image width.		

### Hierarchy

```

Object
  TagSupport
    BodyTagSupport
      AbstractBaseTag
        Image
  
```

### All Implemented Interfaces

BodyTag, IAttributeConsumer, IterationTag, Serializable, Tag, URLParams

## Field Summary

```

public
static SRC
final String
protected String
    tagId
String
  
```

### Fields from `com.bea.wlw.netui.tags.AbstractBaseTag`

ATTR\_GENERAL, ATTR\_GENERAL\_EXPRESSION, ATTR\_JAVASCRIPT, ATTR\_STYLE, CLASS, ID, JAVASCRIPT\_STATUS, NAME, NETUI\_UNIQUE\_CNT, ONCLICK,

ONDBLCLICK, ONKEYDOWN, ONKEYPRESS, ONKEYUP, ONMOUSEDOWN, ONMOUSEMOVE, ONMOUSEOUT, ONMOUSEOVER, ONMOUSEUP, STYLE, TABINDEX

### Fields from `javax.servlet.jsp.tagext.BodyTagSupport`

`bodyContent`

### Fields from `javax.servlet.jsp.tagext.TagSupport`

`id`, `pageContext`

### Fields from interface `javax.servlet.jsp.tagext.BodyTag`

`EVAL_BODY_BUFFERED`, `EVAL_BODY_TAG`

### Fields from interface `javax.servlet.jsp.tagext.IterationTag`

`EVAL_BODY_AGAIN`

### Fields from interface `javax.servlet.jsp.tagext.Tag`

`EVAL_BODY_INCLUDE`, `EVAL_PAGE`, `SKIP_BODY`, `SKIP_PAGE`

## Constructor Summary

***Image()***

## Method Summary

```
public
    void addParameter(String name, Object value)
        Adds a URL parameter to the generated hyperlink.

public
    int doEndTag()
        Render the end of the IMG tag.

public
    int doStartTag()
        Render the beginning of the IMG tag.

public
    String getAlign()
        Gets the property to specify where to align the image.

public
    String getAlt()
        Gets the property to specify alt text of the image.
```

```

public getBorder()
String      Gets the border size around the image.

public
String getHeight()
           Gets the image height.

public
String getHspace()
           Gets the horizontal spacing around the image.

public
String getId()
           Gets the image id.

public
String getIsmap()
           Gets the server-side image map declaration.

public
String getLocation()
           Gets the location hash to append to the url.

public
String getLowsrc()
           Gets the low resolution image source URI.

public
String getOnClick()
           Gets the onClick javascript event.

public
String getOnDbClick()
           Gets the onDbClick javascript event.

public
String getOnKeyDown()
           Gets the onKeyDown javascript event.

public
String getOnKeyPress()
           Gets the onKeyPress javascript event.

public
String getOnKeyUp()
           Gets the onKeyUp javascript event.

public
String getOnMouseDown()
           Gets the onMouseDown javascript event.

public
String getOnMouseMove()
           Gets the onMouseMove javascript event.

public
String getOnMouseOut()
           Gets the onMouseOut javascript event.

public
String getOnMouseOver()
           Gets the onMouseOver javascript event.

```

```

public getOnMouseUp()
String      Gets the onMouseUp javascript event.

public
String getPage()
            Gets the module–relative path, starting with a slash character, of the image to be
            displayed by this rendered tag.

public
String getSrc()
            Gets the image source URI.

public
String getStyle()
            Gets the style of the rendered html tag.

public
String getStyleClass()
            Gets the style class of the rendered html tag.

public
String getTabIndex()
            Gets the tabIndex of the rendered html tag.

public
String getTagId()
            Return the ID of the image.

public
String getTagName()
            Return the name of the Tag.

public
String getUsemap()
            Gets the client–side image map declaration.

public
String getVspace()
            Gets the vertical spacing around the image.

public
String getWidth()
            Gets the image width.

protected
void localRelease()
            Release any acquired resources.

public
void setAlign(String align)
            Sets the property to specify where to align the image.

public
void setAlt(String alt)
            Sets the property to specify the alt text of the image.

public
void setAttribute(String name, String value)
            Set an attribute value.

public
void setBorder(String border)

```

Sets the border size around the image.

```

public
    void setHeight(String height)
        Sets the image height.

public
    void setHspace(String hspace)
        Sets the the horizontal spacing around the image.

public
    void setId(String id)
        Sets the image id.

public
    void setIsmap(String ismap)
        Sets the server-side image map declaration.

public
    void setLocation(String location)
        Sets the location hash to append to the url.

public
    void setLowsrc(String lowsrc)
        Sets the the low resolution image source URI.

public
    void setOnClick(String onclick)
        Sets the onClick javascript event.

public
    void setOnDbClick(String ondblclick)
        Sets the onDbClick javascript event.

public
    void setOnKeyDown(String onkeydown)
        Sets the onKeyDown javascript event.

public
    void setOnKeyPress(String onkeypress)
        Sets the onKeyPress javascript event.

public
    void setOnKeyUp(String onkeyup)
        Sets the onKeyUp javascript event.

public
    void setOnMouseDown(String onmousedown)
        Sets the onMouseDown javascript event.

public
    void setOnMouseMove(String onmousemove)
        Sets the onMouseMove javascript event.

public
    void setOnMouseOut(String onmouseout)
        Sets the onMouseOut javascript event.

public
    void setOnMouseOver(String onmouseover)
        Sets the onMouseOver javascript event.
    
```

```

public setOnMouseUp(String onMouseup)
    void          Sets the onMouseUp javascript event.
public
    void setPage(String page)
        Sets the module–relative path, starting with a slash character, of the image to be
        displayed by this rendered tag.
public
    void setSrc(String src)
        Sets the image source URI.
public
    void setStyle(String style)
        Sets the style of the rendered html tag.
public
    void setStyleClass(String styleClass)
        Sets the style class of the rendered html tag.
public
    void setTabIndex(String tabIndex)
        Sets the tabIndex of the rendered html tag.
public
    void setTagId(String tagId)
        Set the ID of the anchor.
public
    void setUsemap(String usemap)
        Sets the client–side image map declaration.
public
    void setVspace(String vspace)
        Sets the vertical spacing around the image.
public
    void setWidth(String width)
        Sets the image width.

```

### Methods from `com.bea.wlw.netui.tags.AbstractBaseTag`

addTagIdMapping, evaluateAttributeToString, filter, filter,  
 getJavaScriptUtils, getNearestForm, getScriptReporter, getUserLocale,  
 registerTagError, release, renderAttribute, rewriteName, write

### Methods from `javax.servlet.jsp.tagext.BodyTagSupport`

doAfterBody, doInitBody, getBodyContent, getPreviousOut, release,  
 setBodyContent

### Methods from `javax.servlet.jsp.tagext.TagSupport`

doAfterBody, doEndTag, doStartTag, findAncestorWithClass, getParent,  
 getValue, getValues, release, removeValue, setPageContext, setParent,  
 setValue

### Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

**Methods from interface `javax.servlet.jsp.tagext.BodyTag`**

`doInitBody, setBodyContent`

**Methods from interface `com.bea.wlw.netui.tags.IAttributeConsumer`**

`setAttribute`

**Methods from interface `javax.servlet.jsp.tagext.IterationTag`**

`doAfterBody`

**Methods from interface `javax.servlet.jsp.tagext.Tag`**

`doEndTag, doStartTag, getParent, release, setPageContext, setParent`

**Methods from interface `com.bea.wlw.netui.tags.html.URLParams`**

`addParameter`

## Field Detail

### SRC

```
public static final String SRC
```

---

### tagId

```
protected String tagId
```

## Constructor Detail

### Image

```
public Image()
```

## Method Detail

## addParameter(String, Object) Method

```
public void addParameter(String name,  
                        Object value)  
    throws JspException
```

Adds a URL parameter to the generated hyperlink.

### Parameters

*name*

– the name of the parameter to be added.

*value*

– the value of the parameter to be added (a String or String[]).

### Exceptions

*JspException*

---

## doEndTag() Method

```
public int doEndTag()  
    throws JspException
```

Render the end of the IMG tag.

### Overrides

`BodyTagSupport.doEndTag()`

### Exceptions

*JspException*

if a JSP exception has occurred

---

## doStartTag() Method

```
public int doStartTag()  
    throws JspException
```

Render the beginning of the IMG tag.

### Overrides

`BodyTagSupport.doStartTag()`

### Exceptions

*JspException*

if a JSP exception has occurred



## **getAlign() Method**

```
public String getAlign()
```

Gets the property to specify where to align the image.

### **Returns**

*the image alignment.*

---

## **getAlt() Method**

```
public String getAlt()
```

Gets the property to specify alt text of the image.

### **Returns**

*the alt string.*

---

## **getBorder() Method**

```
public String getBorder()
```

Gets the border size around the image.

### **Returns**

*the border size.*

---

## **getHeight() Method**

```
public String getHeight()
```

Gets the image height.

### **Returns**

*the height.*

---

## **getHspace() Method**

```
public String getHspace()
```

Gets the horizontal spacing around the image.

### Returns

*the horizontal spacing.*

---

## getId() Method

```
public String getId()
```

Gets the image id.

### Overrides

```
TagSupport.getId()
```

### Returns

*the image id.*

---

## getIsmap() Method

```
public String getIsmap()
```

Gets the server-side image map declaration.

### Returns

*the image map declaration.*

---

## getLocation() Method

```
public String getLocation()
```

Gets the location hash to append to the url.

### Returns

*the location hash.*

---

## getLowsrc() Method

```
public String getLowsrc()
```

Gets the low resolution image source URI.

## Returns

*the low-res image source URI.*

---

## getOnClick() Method

```
public String getOnClick()
```

Gets the onClick javascript event.

## Returns

*the onClick event.*

---

## getOnDbClick() Method

```
public String getOnDbClick()
```

Gets the onDbClick javascript event.

## Returns

*the onDbClick event.*

---

## getOnKeyDown() Method

```
public String getOnKeyDown()
```

Gets the onKeyDown javascript event.

## Returns

*the onKeyDown event.*

---

## getOnKeyPress() Method

```
public String getOnKeyPress()
```

Gets the onKeyPress javascript event.

## Returns

*the onKeyPress event.*

---

## getOnKeyUp() Method

```
public String getOnKeyUp()
```

Gets the onKeyUp javascript event.

### Returns

*the onKeyUp event.*

---

## getOnMouseDown() Method

```
public String getOnMouseDown()
```

Gets the onMouseDown javascript event.

### Returns

*the onMouseDown event.*

---

## getOnMouseMove() Method

```
public String getOnMouseMove()
```

Gets the onMouseMove javascript event.

### Returns

*the onMouseMove event.*

---

## getOnMouseOut() Method

```
public String getOnMouseOut()
```

Gets the onMouseOut javascript event.

### Returns

*the onMouseOut event.*

---

## getOnMouseOver() Method

```
public String getOnMouseOver()
```

Gets the onMouseOver javascript event.

**Returns**

*the onMouseOver event.*

---

**getOnMouseUp() Method**

```
public String getOnMouseUp()
```

Gets the onMouseUp javascript event.

**Returns**

*the onMouseUp event.*

---

**getPage() Method**

```
public String getPage()
```

Gets the module–relative path, starting with a slash character, of the image to be displayed by this rendered tag.

**Returns**

*the module–relative path.*

---

**getSrc() Method**

```
public String getSrc()
```

Gets the image source URI.

**Returns**

*the source URI.*

---

**getStyle() Method**

```
public String getStyle()
```

Gets the style of the rendered html tag.

**Returns**

*the style.*

---

## getStyleClass() Method

```
public String getStyleClass()
```

Gets the style class of the rendered html tag.

### Returns

*the style class.*

---

## getTabIndex() Method

```
public String getTabIndex()
```

Gets the tabIndex of the rendered html tag.

### Returns

*the tabindex.*

---

## getTagId() Method

```
public String getTagId()
```

Return the ID of the image. The id may be rewritten by the container (such as a portal) to make sure it is unique. JavaScript may lookup the actual id of the element by looking it up in the netui\_names table written into the HTML.

### Returns

*the tagId.*

---

## getTagName() Method

```
public String getTagName()
```

Return the name of the Tag.

### Overrides

```
AbstractBaseTag.getTagName()
```

---

## getUsemap() Method

```
public String getUsemap()
```

Gets the client-side image map declaration.

**Returns**

*the map declaration.*

---

**getVspace() Method**

```
public String getVspace()
```

Gets the vertical spacing around the image.

**Returns**

*the vertical spacing.*

---

**getWidth() Method**

```
public String getWidth()
```

Gets the image width.

**Returns**

*the image width.*

---

**localRelease() Method**

```
protected void localRelease()
```

Release any acquired resources.

**Overrides**

```
AbstractBaseTag.localRelease()
```

---

**setAlign(String) Method**

```
public void setAlign(String align)
```

Sets the property to specify where to align the image.

**Parameters**

*align*

– the image alignment.

---

## setAlt(String) Method

```
public void setAlt(String alt)
```

Sets the property to specify the alt text of the image.

### Parameters

*alt*

– the image alignment.

---

## setAttribute(String, String) Method

```
public void setAttribute(String name,  
                        String value)  
    throws JspException
```

Set an attribute value. The `name` represents the name of the attribute. The `value` represents the value and may contain a netui expression. This method may result in errors being generated. This requires that the tag buffer its body and write attributes in the end tag. For the image tag it is not legal to set the `id`, `name`, or `src` attributes with this method.

### Parameters

*name*

The name of the attribute. This value may not be null or the empty string.

*value*

The value of the attribute. This may contain a netui expression.

### Exceptions

*JspException*

A `JspException` may be thrown if there is an error setting the attribute.

---

## setBorder(String) Method

```
public void setBorder(String border)
```

Sets the border size around the image.

### Parameters

*border*

– the border size.

---



## setHeight(String) Method

```
public void setHeight(String height)
```

Sets the image height.

### Parameters

*height*

– the height.

---

## setHspace(String) Method

```
public void setHspace(String hspace)
```

Sets the the horizontal spacing around the image.

### Parameters

*hspace*

– the horizontal spacing.

---

## setId(String) Method

```
public void setId(String id)
```

Sets the image id.

### Overrides

`TagSupport.setId(String)`

### Parameters

*id*

– the image id

---

## setIsmap(String) Method

```
public void setIsmap(String ismap)
```

Sets the server-side image map declaration.

### Parameters

*ismap*

– the image map declaration.

---

## setLocation(String) Method

```
public void setLocation(String location)
```

Sets the location hash to append to the url.

### Parameters

*location*

– the location hash.

---

## setLowsrc(String) Method

```
public void setLowsrc(String lowsrc)
```

Sets the the low resolution image source URI.

### Parameters

*lowsrc*

– the low-res image source URI.

---

## setOnClick(String) Method

```
public void setOnClick(String onclick)
```

Sets the onClick javascript event.

### Parameters

*onclick*

– the onClick event.

---

## setOnDbClick(String) Method

```
public void setOnDbClick(String ondblclick)
```

Sets the onDbClick javascript event.

### Parameters

*ondblclick*

– the onDbClick event.

---

## setOnKeyDown(String) Method

```
public void setOnKeyDown(String onkeydown)
```

Sets the onKeyDown javascript event.

### Parameters

*onkeydown*

– the onKeyDown event.

---

## setOnKeyPress(String) Method

```
public void setOnKeyPress(String onkeypress)
```

Sets the onKeyPress javascript event.

### Parameters

*onkeypress*

– the onKeyPress event.

---

## setOnKeyUp(String) Method

```
public void setOnKeyUp(String onkeyup)
```

Sets the onKeyUp javascript event.

### Parameters

*onkeyup*

– the onKeyUp event.

---

## setOnMouseDown(String) Method

```
public void setOnMouseDown(String onmousedown)
```

Sets the onMouseDown javascript event.

### Parameters

*onmousedown*

– the onMouseDown event.

---

## setOnMouseMove(String) Method

```
public void setOnMouseMove(String onmousemove)
```

Sets the onMouseMove javascript event.

### Parameters

*onmousemove*

– the onMouseMove event.

---

## setOnMouseOut(String) Method

```
public void setOnMouseOut(String onmouseout)
```

Sets the onMouseOut javascript event.

### Parameters

*onmouseout*

– the onMouseOut event.

---

## setOnMouseOver(String) Method

```
public void setOnMouseOver(String onmouseover)
```

Sets the onMouseOver javascript event.

### Parameters

*onmouseover*

– the onMouseOver event.

---

## setOnMouseUp(String) Method

```
public void setOnMouseUp(String onmouseup)
```

Sets the onMouseUp javascript event.

### Parameters

*onmouseup*

– the onMouseUp event.

---

## setPage(String) Method

```
public void setPage(String page)
```

Sets the module–relative path, starting with a slash character, of the image to be displayed by this rendered tag.

### Parameters

*page*

– the module–relative path.

---

## setSrc(String) Method

```
public void setSrc(String src)
```

Sets the image source URI.

### Parameters

*src*

– the source URI.

---

## setStyle(String) Method

```
public void setStyle(String style)
```

Sets the style of the rendered html tag.

### Parameters

*style*

– the html style.

---

## setStyleClass(String) Method

```
public void setStyleClass(String styleClass)
```

Sets the style class of the rendered html tag.

### Parameters

*styleClass*

– the html style class.

---

## setTabIndex(String) Method

```
public void setTabIndex(String tabIndex)
```

Sets the tabIndex of the rendered html tag.

### Parameters

*tabIndex*

– the tab index.

---

## setTagId(String) Method

```
public void setTagId(String tagId)
```

Set the ID of the anchor.

### Parameters

*tagId*

– the tagId.

---

## setUsemap(String) Method

```
public void setUsemap(String usemap)
```

Sets the client-side image map declaration.

### Parameters

*usemap*

– the map declaration.

---

## setVspace(String) Method

```
public void setVspace(String vspace)
```

Sets the vertical spacing around the image.

### Parameters

*vspace*

– the vertical spacing.

---

## **setWidth(String) Method**

```
public void setWidth(String width)
```

Sets the image width.

### **Parameters**

*width*

– the image width.

---

# ImageAnchor Class

*public class ImageAnchor*

extends Anchor

Generates a hyperlink with a clickable image. ImageAnchor provides support for image rollovers. An imageAnchor must have one of five attributes to correctly create the hyperlink:

- action – an action invoked by clicking the hyperlink.
- forward – a forward to redirect to.
- href – a URL to go to
- linkName – an internal place in the page to move to
- page – The module–relative page URL to which this hyperlink will be rendered.

Attribute Descriptions			
Attribute	Required	Runtime Expression Evaluation	Data Bindable
accessKey	No	No	No
	Define the keyboard navigation key for the element.		
action	No	No	Read Only
	An action invoked by clicking the link.		
align	No	No	No
	Property to specify where to align the image.		
alt	No	No	Read Only
	Property to specify alt text of an image.		
border	No	No	Read Only
	The border size around the image		
formSubmit	No	No	No
	True if clicking this link should submit its enclosing form.		
forward	No	No	No
	The logical forward name from which to retrieve the hyperlink URI.		
height	No	No	Read Only
	The image height		



## Page Flow API Reference

<b>href</b>	No	No	Read Only
	A URL to go to.		
<b>hspace</b>	No	No	Read Only
	The horizontal spacing around the image		
<b>id</b>	No	No	No
	<b>The id attribute has been deprecated;</b> instead, use <code>tagId</code> . The <code>id</code> attribute of the generated <code>ImageAnchor</code> . You should use the <code>tagId</code> which will encode the <code>id</code> correctly for containers such as the Portal which needs to rewrite IDs to make them unique.		
<b>isMap</b>	No	No	No
	The server-side image map declaration.		
<b>linkName</b>	No	No	No
	An internal place in the page to move to.		
<b>location</b>	No	No	No
	Location within the URI to visit.		
<b>lowsrc</b>	No	No	No
	The low resolution image source URI.		
<b>page</b>	No	No	No
	<b>The page attribute has been deprecated.</b> The module-relative page URL to which this hyperlink will be rendered.		
<b>rolloverImage</b>	No	No	No
	The URI of the rollover image.		
<b>src</b>	No	No	Read Only
	The image source URI		
<b>target</b>	No	No	No
	The window target.		
<b>tagId</b>	No	No	No
	The <code>id</code> of this <code>ImageAnchor</code> . The real <code>id</code> written to the HTML stream may be changed by the container (Portal), but may be looked up with the <code>tagId</code> value through the JavaScript method <code>getNetuiTagName( tagId, tag )</code> method written by the <code>&lt;netui:html&gt;</code> tag.		
<b>useMap</b>	No	No	No

	The client–side image map declaration.		
<b>vspace</b>	No	No	Read Only
	The vertical spacing around the image.		
<b>width</b>	No	No	Read Only
	The image width.		

**Related Topics**

<netui:imageAnchor> Tag Sample

---

***Hierarchy***

```

Object
  TagSupport
    BodyTagSupport
      AbstractBaseTag
        Anchor
          ImageAnchor

```

***All Implemented Interfaces***

BodyTag, IAttributeConsumer, IterationTag, Serializable, Tag, URLParams

---

## Field Summary

protected String

***src***

String

**Fields from `com.bea.wlw.netui.tags.html.Anchor`**

HREF, id

**Fields from `com.bea.wlw.netui.tags.AbstractBaseTag`**

ATTR\_GENERAL, ATTR\_GENERAL\_EXPRESSION, ATTR\_JAVASCRIPT, ATTR\_STYLE, CLASS, ID, JAVASCRIPT\_STATUS, NAME, NETUI\_UNIQUE\_CNT, ONCLICK, ONDBLCLICK, ONKEYDOWN, ONKEYPRESS, ONKEYUP, ONMOUSEDOWN, ONMOUSEMOVE, ONMOUSEOUT, ONMOUSEOVER, ONMOUSEUP, STYLE, TABINDEX

**Fields from `javax.servlet.jsp.tagext.BodyTagSupport`**

bodyContent

**Fields from `javax.servlet.jsp.tagext.TagSupport`**

`id, pageContext`

#### Fields from interface `javax.servlet.jsp.tagext.BodyTag`

`EVAL_BODY_BUFFERED, EVAL_BODY_TAG`

#### Fields from interface `javax.servlet.jsp.tagext.IterationTag`

`EVAL_BODY_AGAIN`

#### Fields from interface `javax.servlet.jsp.tagext.Tag`

`EVAL_BODY_INCLUDE, EVAL_PAGE, SKIP_BODY, SKIP_PAGE`

## Constructor Summary

***ImageAnchor()***

## Method Summary

```
protected
    void createAnchorAttributes(UnsyncStringBuffer results)
        Override so we only write the style and javascript attributes on the anchor.

    public
        int doEndTag()
            Render the image and hyperlink.

    public
        int doStartTag()
            Insert rollover javascript.

    public
        String getAccessKey()
            Gets the accessKey attribute value.

    public
        String getAlign()
            Gets the property to specify where to align the image.

    public
        String getAlt()
            Gets the property to specify alt text of the image.

    public
        String getBorder()
            Gets the border size around the image.

    public
        String getHeight()
            Gets the image height.
```

```

public getHspace()
String      Gets the horizontal spacing around the image.

public
String getIsmap()
            Gets the server–side image map declaration.

public
String getLowsrc()
            Gets the low resolution image source URI.

public
String getPage()
            Gets the module–relative path, starting with a slash character, of the image to be
            displayed by this rendered tag.

public
String getRolloverImage()
            Gets the roll–over image of the ImageAnchor.

public
String getSrc()
            Gets the image source URI.

public
String getTagName()
            Return the name of the Tag.

public
String getUsemap()
            Gets the client–side image map declaration.

public
String getVspace()
            Gets the vertical spacing around the image.

public
String getWidth()
            Gets the image width.

protected
void localRelease()
            Release any acquired resources.

public
void setAccessKey(String accessKey)
            Sets the accessKey attribute value.

public
void setAlign(String align)
            Sets the property to specify where to align the image.

public
void setAlt(String alt)
            Sets the property to specify the alt text of the image.

public
void setBorder(String border)
            Sets the border size around the image.

public
void setHeight(String height)

```

Sets the image height.

```

public
    void setHspace(String hspace)
        Sets the the horizontal spacing around the image.

public
    void setIsmap(String ismap)
        Sets the server-side image map declaration.

public
    void setLowsrc(String lowsrc)
        Sets the the low resolution image source URI.

public
    void setPage(String page)
        Sets the module-relative path, starting with a slash character, of the image to be
        displayed by this rendered tag.

public
    void setRolloverImage(String rolloverImage)
        Sets the roll-over image of the ImageAnchor.

public
    void setSrc(String src)
        Sets the image source URI.

public
    void setUsemap(String usemap)
        Sets the client-side image map declaration.

public
    void setVspace(String vspace)
        Sets the vertical spacing around the image.

public
    void setWidth(String width)
        Sets the image width.
    
```

### Methods from **com.bea.wlw.netui.tags.html.Anchor**

```

addParameter, createAnchorBeginTag, doAfterBody, getAction,
getFormSubmit, getForward, getHref, getId, getLinkName, getLocation,
getOnBlur, getOnClick, getOnDbClick, getOnFocus, getOnKeyDown,
getOnKeyPress, getOnKeyUp, getOnMouseDown, getOnMouseMove,
getOnMouseOut, getOnMouseOver, getOnMouseUp, getRealFormName, getStyle,
getStyleClass, getTabIndex, getTagId, getTarget, setAction,
setAttribute, setFormSubmit, setForward, setHref, setId, setLinkName,
setLocation, setOnBlur, setOnClick, setOnDbClick, setOnFocus,
setOnKeyDown, setOnKeyPress, setOnKeyUp, setOnMouseDown, setOnMouseMove,
setOnMouseOut, setOnMouseOver, setOnMouseUp, setStyle, setStyleClass,
setTabIndex, setTagId, setTarget,
    
```

### Methods from **com.bea.wlw.netui.tags.AbstractBaseTag**

```

addTagIdMapping, evaluateAttributeToString, filter, filter,
getJavaScriptUtils, getNearestForm, getScriptReporter, getTagName,
getUserLocale, localRelease, registerTagError, release, renderAttribute,
    
```

rewriteName, write

**Methods from javax.servlet.jsp.tagext.BodyTagSupport**

doAfterBody, doEndTag, doInitBody, doStartTag, getBodyContent, getPreviousOut, release, setBodyContent

**Methods from javax.servlet.jsp.tagext.TagSupport**

doAfterBody, doEndTag, doStartTag, findAncestorWithClass, getId, getParent, getValue, getValues, release, removeValue, setId, setPageContext, setParent, setValue

**Methods from class java.lang.Object**

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

**Methods from interface javax.servlet.jsp.tagext.BodyTag**

doInitBody, setBodyContent

**Methods from interface com.bea.wlw.netui.tags.IAttributeConsumer**

setAttribute

**Methods from interface javax.servlet.jsp.tagext.IterationTag**

doAfterBody

**Methods from interface javax.servlet.jsp.tagext.Tag**

doEndTag, doStartTag, getParent, release, setPageContext, setParent

**Methods from interface com.bea.wlw.netui.tags.html.URLParams**

addParameter

## Field Detail

### src

protected String src

## Constructor Detail

### ImageAnchor

```
public ImageAnchor()
```

## Method Detail

### createAnchorAttributes(UnsyncStringBuffer) Method

```
protected void createAnchorAttributes(UnsyncStringBuffer results)
```

Override so we only write the style and javascript attributes on the anchor.

#### *Overrides*

```
Anchor.createAnchorAttributes(UnsyncStringBuffer)
```

---

### doEndTag() Method

```
public int doEndTag()  
    throws JspException
```

Render the image and hyperlink.

#### *Overrides*

```
Anchor.doEndTag()
```

### Exceptions

*JspException*  
if a JSP exception has occurred

---

### doStartTag() Method

```
public int doStartTag()  
    throws JspException
```

Insert rollover javascript.

Support for indexed property since Struts 1.1

#### *Overrides*

```
Anchor.doStartTag()
```

## Exceptions

*JspException*

if a JSP exception has occurred

---

## getAccessKey() Method

```
public String getAccessKey()
```

Gets the accessKey attribute value.

### Overrides

```
Anchor.getAccessKey()
```

## Returns

*the accessKey value.*

---

## getAlign() Method

```
public String getAlign()
```

Gets the property to specify where to align the image.

## Returns

*the image alignment.*

---

## getAlt() Method

```
public String getAlt()
```

Gets the property to specify alt text of the image.

## Returns

*the alt string.*

---

## getBorder() Method

```
public String getBorder()
```

Gets the border size around the image.



## Returns

*the border size.*

---

## getHeight() Method

```
public String getHeight()
```

Gets the image height.

## Returns

*the height.*

---

## getHspace() Method

```
public String getHspace()
```

Gets the horizontal spacing around the image.

## Returns

*the horizontal spacing.*

---

## getIsmap() Method

```
public String getIsmap()
```

Gets the server-side image map declaration.

## Returns

*the image map declaration.*

---

## getLowsrc() Method

```
public String getLowsrc()
```

Gets the low resolution image source URI.

## Returns

*the low-res image source URI.*

---

## getPage() Method

```
public String getPage()
```

Gets the module–relative path, starting with a slash character, of the image to be displayed by this rendered tag.

### *Overrides*

```
Anchor.getPage()
```

### Returns

*the module–relative path.*

---

## getRolloverImage() Method

```
public String getRolloverImage()
```

Gets the roll–over image of the ImageAnchor.

### Returns

*the rollover image.*

---

## getSrc() Method

```
public String getSrc()
```

Gets the image source URI.

### Returns

*the image source URI.*

---

## getTagName() Method

```
public String getTagName()
```

Return the name of the Tag.

### *Overrides*

```
Anchor.getTagName()
```

---

## getUsemap() Method

```
public String getUsemap()
```

Gets the client-side image map declaration.

### Returns

*the map declaration.*

---

## getVspace() Method

```
public String getVspace()
```

Gets the vertical spacing around the image.

### Returns

*the vertical spacing.*

---

## getWidth() Method

```
public String getWidth()
```

Gets the image width.

### Returns

*the image width.*

---

## localRelease() Method

```
protected void localRelease()
```

Release any acquired resources.

### Overrides

`Anchor.localRelease()`

---

## setAccessKey(String) Method

```
public void setAccessKey(String accessKey)
```

Sets the accessKey attribute value. This should key value of the keyboard navigation key. It is recommended not to use the following values because there are often used by browsers A, C, E, F, G, H, V, left arrow, and right arrow.

### ***Overrides***

`Anchor.setAccessKey(String)`

### **Parameters**

*accessKey*

– the accessKey value.

---

## **setAlign(String) Method**

```
public void setAlign(String align)
```

Sets the property to specify where to align the image.

### **Parameters**

*align*

– the image alignment.

---

## **setAlt(String) Method**

```
public void setAlt(String alt)
```

Sets the property to specify the alt text of the image.

### **Parameters**

*alt*

– the image alignment.

---

## **setBorder(String) Method**

```
public void setBorder(String border)
```

Sets the border size around the image.

### **Parameters**

*border*

– the border size.

---

## **setHeight(String) Method**

```
public void setHeight(String height)
```

Sets the image height.

**Parameters**

*height*  
– the height.

---

**setHspace(String) Method**

```
public void setHspace(String hspace)
```

Sets the the horizontal spacing around the image.

**Parameters**

*hspace*  
– the horizontal spacing.

---

**setIsmap(String) Method**

```
public void setIsmap(String ismap)
```

Sets the server–side image map declaration.

**Parameters**

*ismap*  
– the image map declaration.

---

**setLowsrc(String) Method**

```
public void setLowsrc(String lowsrc)
```

Sets the the low resolution image source URI.

**Parameters**

*lowsrc*  
– the low–res image source URI.

---

**setPage(String) Method**

```
public void setPage(String page)
```

Sets the module–relative path, starting with a slash character, of the image to be displayed by this rendered tag.

**Overrides**

`Anchor.setPage(String)`

### Parameters

*page*

– the module–relative path.

---

### setRolloverImage(String) Method

```
public void setRolloverImage(String rolloverImage)
```

Sets the roll–over image of the ImageAnchor.

### Parameters

*rolloverImage*

– the rollover image.

---

### setSrc(String) Method

```
public void setSrc(String src)
```

Sets the image source URI.

### Parameters

*src*

– the image source URI.

---

### setUsemap(String) Method

```
public void setUsemap(String usemap)
```

Sets the client–side image map declaration.

### Parameters

*usemap*

– the map declaration.

---

### setVspace(String) Method

```
public void setVspace(String vspace)
```

Sets the vertical spacing around the image.

## Parameters

*vspace*

– the vertical spacing.

---

## setWidth(String) Method

```
public void setWidth(String width)
```

Sets the image width.

## Parameters

*width*

– the image width.

---

# ImageButton Class

*public class* **ImageButton**

extends AbstractBaseTag  
implements IAttributeConsumer

Generates an `<input type="image">` tag with the specified attributes. ImageButton ignores its body content.

Attribute Descriptions			
Attribute	Required	Runtime Expression Evaluation	Data Bindable
<b>accessKey</b>	No	No	No
	Define the keyboard navigation key for the element.		
<b>align</b>	No	No	No
	Property to specify where to align the image.		
<b>alt</b>	No	No	Read Only
	Property to specify alt text of an image.		
<b>disabled</b>	No	No	Read Only
	A boolean value which if true will cause the ImageButton to be disabled.		
<b>height</b>	No	No	Read Only
	The image height		
<b>hspace</b>	No	No	Read Only
	The horizontal spacing around the image		
<b>id</b>	No	No	No
	<b>The id attribute has been deprecated;</b> instead, use <code>tagId</code> . The <code>id</code> attribute of the generated ImageButton. You should use the <code>tagId</code> which will encode the id correctly for containers such as the Portal which needs to rewrite IDs to make them unique.		
<b>page</b>	No	No	No
	<b>The page attribute has been deprecated.</b> The module-relative page URL to which this hyperlink will be rendered.		



<b>rolloverImage</b>	No	No	No
	The URI of the rollover image		
<b>src</b>	No	No	Read Only
	The URI of the image		
<b>tagId</b>	No	No	No
	The id of this image. The real id written to the HTML stream may be changed by the container (Portal), but may be looked up with the 'tagId' value through the JavaScript method <code>getNetuiTagName( tagId, tag )</code> method written by the <code>&lt;netui:html&gt;</code> tag.		
<b>value</b>	No	No	No
	The value of the image button. <b>This attribute has been deprecated.</b>		
<b>vspace</b>	No	No	Read Only
	The vertical spacing around the image.		
<b>width</b>	No	No	Read Only
	The image width.		

## Related Topics

`<netui:imageButton>` Tag Sample

---

## Hierarchy

```

Object
  TagSupport
    BodyTagSupport
      AbstractBaseTag
        ImageButton

```

## All Implemented Interfaces

BodyTag, IAttributeConsumer, IterationTag, Serializable, Tag

---

## Field Summary

protected String

***align***

The alignment for this image.

protected String

***page***

The module–relative URI of the image.

protected String

***rolloverImage***

The roll–over image of the ImageButton.

protected String

***src***

The URL of this image.

public

static

***SRC***

final String

String

protected String

***tagId***

String

protected String

***value***

The value of the image.

public

static

***VALUE***

final String

String

#### Fields from `com.bea.wlw.netui.tags.AbstractBaseTag`

ATTR\_GENERAL, ATTR\_GENERAL\_EXPRESSION, ATTR\_JAVASCRIPT, ATTR\_STYLE, CLASS, ID, JAVASCRIPT\_STATUS, NAME, NETUI\_UNIQUE\_CNT, ONCLICK, ONDBLCLICK, ONKEYDOWN, ONKEYPRESS, ONKEYUP, ONMOUSEDOWN, ONMOUSEMOVE, ONMOUSEOUT, ONMOUSEOVER, ONMOUSEUP, STYLE, TABINDEX

#### Fields from `javax.servlet.jsp.tagext.BodyTagSupport`

bodyContent

#### Fields from `javax.servlet.jsp.tagext.TagSupport`

id, pageContext

#### Fields from interface `javax.servlet.jsp.tagext.BodyTag`

EVAL\_BODY\_BUFFERED, EVAL\_BODY\_TAG

#### Fields from interface `javax.servlet.jsp.tagext.IterationTag`

EVAL\_BODY\_AGAIN

#### Fields from interface `javax.servlet.jsp.tagext.Tag`

EVAL\_BODY\_INCLUDE, EVAL\_PAGE, SKIP\_BODY, SKIP\_PAGE

## Constructor

### Summary

*ImageButton()*

## Method

### Summary

```

public
    int doAfterBody()
        Save the body content of the ImageButton.

public
    int doEndTag()
        Process the end of this tag.

public
    int doStartTag()
        Process the start of this tag.

public
    String getAccessKey()
        Gets the accessKey attribute value.

public
    String getAlign()
        Gets the property to specify where to align the image.

public
    String getAlt()
        Gets the property to specify alt text of the image.

public
    String getDisabled()
        Get a boolean value indicating if this ImageButton is disabled or not.

public
    String getHeight()
        Gets the image height.

public
    String getHspace()
        Gets the horizontal spacing around the image.

public
    String getId()
        Gets the imageButton id.

public
    String getOnBlur()
        Gets the onBlur javascript event.

public
    String getOnChange()
        Gets the onChange javascript event.

```

```

public getOnClick()
String      Gets the onClick javascript event.

public
String getOnDbClick()
           Gets the onDbClick javascript event.

public
String getOnFocus()
           Gets the onFocus javascript event.

public
String getOnKeyDown()
           Gets the onKeyDown javascript event.

public
String getOnKeyPress()
           Gets the onKeyPress javascript event.

public
String getOnKeyUp()
           Gets the onKeyUp javascript event.

public
String getOnMouseDown()
           Gets the onMouseDown javascript event.

public
String getOnMouseMove()
           Gets the onMouseMove javascript event.

public
String getOnMouseOut()
           Gets the onMouseOut javascript event.

public
String getOnMouseOver()
           Gets the onMouseOver javascript event.

public
String getOnMouseUp()
           Gets the onMouseUp javascript event.

public
String getOnSelect()
           Gets the onSelect javascript event.

public
String getPage()
           Gets the module–relative path, starting with a slash character, of the image to be
           displayed by this rendered tag.

public
String getRolloverImage()
           Gets the roll–over image of the ImageAnchor.

public
String getSrc()
           Gets the image source URI.

public
String getStyle()

```

Gets the style of the rendered html tag.

```
public
String getStyleClass()
```

Gets the style class of the rendered html tag.

```
public
String getTabIndex()
```

Gets the tabIndex of the rendered html tag.

```
public
String getTagId()
```

Return the ID of the imageButton.

```
public
String getTagName()
```

Return the name of the Tag.

```
public
String getValue()
```

Returns the value of the ImageButton.

```
public
String getVspace()
```

Gets the vertical spacing around the image.

```
public
String getWidth()
```

Gets the image width.

```
protected
void localRelease()
```

Release any acquired resources.

```
public
void setAccessKey(String accessKey)
```

Sets the accessKey attribute value.

```
public
void setAlign(String align)
```

Sets the property to specify where to align the image.

```
public
void setAlt(String alt)
```

Sets the property to specify the alt text of the image.

```
public
void setAttribute(String name, String value)
```

Set an attribute value.

```
public
void setDisabled(String disabled)
```

Sets the disable stat of this ImageButton.

```
public
void setHeight(String height)
```

Sets the image height.

```
public
void setHspace(String hspace)
```

Sets the the horizontal spacing around the image.

```

public setId(String id)
    void          Sets the imageButton id.
public
    void setOnBlur(String onblur)
        Sets the onBlur javascript event.
public
    void setOnChange(String onchange)
        Sets the onChange javascript event.
public
    void setOnClick(String onclick)
        Sets the onClick javascript event.
public
    void setOnDbClick(String ondblclick)
        Sets the onDbClick javascript event.
public
    void setOnFocus(String onblur)
        Sets the onFocus javascript event.
public
    void setOnKeyDown(String onkeydown)
        Sets the onKeyDown javascript event.
public
    void setOnKeyPress(String onkeypress)
        Sets the onKeyPress javascript event.
public
    void setOnKeyUp(String onkeyup)
        Sets the onKeyUp javascript event.
public
    void setOnMouseDown(String onmousedown)
        Sets the onMouseDown javascript event.
public
    void setOnMouseMove(String onmousemove)
        Sets the onMouseMove javascript event.
public
    void setOnMouseOut(String onmouseout)
        Sets the onMouseOut javascript event.
public
    void setOnMouseOver(String onmouseover)
        Sets the onMouseOver javascript event.
public
    void setOnMouseUp(String onmouseup)
        Sets the onMouseUp javascript event.
public
    void setOnSelect(String onselect)
        Sets the onSelect javascript event.
public
    void setPage(String page)
        Sets the module–relative path, starting with a slash character, of the image to be

```

displayed by this rendered tag.

```

public
    void setRolloverImage(String rolloverImage)
        Sets the roll-over image of the ImageAnchor.

public
    void setSrc(String src)
        Sets the image source URI.

public
    void setStyle(String style)
        Sets the style of the rendered html tag.

public
    void setStyleClass(String styleClass)
        Sets the style class of the rendered html tag.

public
    void setTabIndex(String tabindex)
        Sets the tabIndex of the rendered html tag.

public
    void setTagId(String tagId)
        Set the ID of the anchor.

public
    void setValue(String value)
        Set the value of the ImageButton.

public
    void setVspace(String vspace)
        Sets the vertical spacing around the image.

public
    void setWidth(String width)
        Sets the image width.

```

#### Methods from `com.bea.wlw.netui.tags.AbstractBaseTag`

`addTagIdMapping`, `evaluateAttributeToString`, `filter`, `filter`,  
`getJavaScriptUtils`, `getNearestForm`, `getScriptReporter`, `getUserLocale`,  
`registerTagError`, `release`, `renderAttribute`, `rewriteName`, `write`

#### Methods from `javax.servlet.jsp.tagext.BodyTagSupport`

`doInitBody`, `getBodyContent`, `getPreviousOut`, `release`, `setBodyContent`

#### Methods from `javax.servlet.jsp.tagext.TagSupport`

`doAfterBody`, `doEndTag`, `doStartTag`, `findAncestorWithClass`, `getParent`,  
`getValue`, `getValues`, `release`, `removeValue`, `setPageContext`, `setParent`,  
`setValue`

#### Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`,  
`toString`, `wait`, `wait`, `wait`

**Methods from interface `javax.servlet.jsp.tagext.BodyTag`**

`doInitBody`, `setBodyContent`

**Methods from interface `com.bea.wlw.netui.tags.IAttributeConsumer`**

`setAttribute`

**Methods from interface `javax.servlet.jsp.tagext.IterationTag`**

`doAfterBody`

**Methods from interface `javax.servlet.jsp.tagext.Tag`**

`doEndTag`, `doStartTag`, `getParent`, `release`, `setPageContext`, `setParent`

## Field Detail

### **align**

`protected String align`

The alignment for this image.

---

### **page**

`protected String page`

The module-relative URI of the image.

---

### **rolloverImage**

`protected String rolloverImage`

The roll-over image of the ImageButton.

---

### **src**

`protected String src`

The URL of this image.

---



## SRC

```
public static final String SRC
```

---

## tagId

```
protected String tagId
```

---

## value

```
protected String value
```

The value of the image.

---

## VALUE

```
public static final String VALUE
```

## Constructor Detail

### ImageButton

```
public ImageButton()
```

## Method Detail

### doAfterBody() Method

```
public int doAfterBody()  
    throws JspException
```

Save the body content of the ImageButton.

#### *Overrides*

```
BodyTagSupport.doAfterBody()
```

### Exceptions

*JspException*  
if a JSP exception has occurred

---

## doEndTag() Method

```
public int doEndTag()  
    throws JspException
```

Process the end of this tag.

### *Overrides*

```
BodyTagSupport.doEndTag( )
```

### Exceptions

*JspException*  
if a JSP exception has occurred

---

## doStartTag() Method

```
public int doStartTag()  
    throws JspException
```

Process the start of this tag.

### *Overrides*

```
BodyTagSupport.doStartTag( )
```

### Exceptions

*JspException*  
if a JSP exception has occurred

---

## getAccessKey() Method

```
public String getAccessKey()
```

Gets the accessKey attribute value.

### Returns

*the accessKey value.*

---

## getAlign() Method

```
public String getAlign()
```

Gets the property to specify where to align the image.

## Returns

*the image alignment.*

---

## getAlt() Method

```
public String getAlt()
```

Gets the property to specify alt text of the image.

## Returns

*the alt string.*

---

## getDisabled() Method

```
public String getDisabled()
```

Get a boolean value indicating if this ImageButton is disabled or not.

## Returns

*the disabled state (true or false)*

---

## getHeight() Method

```
public String getHeight()
```

Gets the image height.

## Returns

*the height.*

---

## getHspace() Method

```
public String getHspace()
```

Gets the horizontal spacing around the image.

## Returns

*the horizontal spacing.*

---

## getId() Method

```
public String getId()
```

Gets the imageButton id.

### *Overrides*

```
TagSupport.getId()
```

### Returns

*the id.*

---

## getOnBlur() Method

```
public String getOnBlur()
```

Gets the onBlur javascript event.

### Returns

*the onBlur event.*

---

## getOnChange() Method

```
public String getOnChange()
```

Gets the onChange javascript event.

### Returns

*the onChange event.*

---

## getOnClick() Method

```
public String getOnClick()
```

Gets the onClick javascript event.

### Returns

*the onClick event.*

---

## getOnDbClick() Method

```
public String getOnDbClick()
```

Gets the onDbClick javascript event.

### Returns

*the onDbClick event.*

---

## getOnFocus() Method

```
public String getOnFocus()
```

Gets the onFocus javascript event.

### Returns

*the onFocus event.*

---

## getOnKeyDown() Method

```
public String getOnKeyDown()
```

Gets the onKeyDown javascript event.

### Returns

*the onKeyDown event.*

---

## getOnKeyPress() Method

```
public String getOnKeyPress()
```

Gets the onKeyPress javascript event.

### Returns

*the onKeyPress event.*

---

## getOnKeyUp() Method

```
public String getOnKeyUp()
```

Gets the onKeyUp javascript event.

**Returns**

*the onKeyUp event.*

---

**getOnMouseDown() Method**

```
public String getOnMouseDown()
```

Gets the onMouseDown javascript event.

**Returns**

*the onMouseDown event.*

---

**getOnMouseMove() Method**

```
public String getOnMouseMove()
```

Gets the onMouseMove javascript event.

**Returns**

*the onMouseMove event.*

---

**getOnMouseOut() Method**

```
public String getOnMouseOut()
```

Gets the onMouseOut javascript event.

**Returns**

*the onMouseOut event.*

---

**getOnMouseOver() Method**

```
public String getOnMouseOver()
```

Gets the onMouseOver javascript event.

**Returns**

*the onMouseOver event.*

---

## getOnMouseUp() Method

```
public String getOnMouseUp()
```

Gets the onMouseUp javascript event.

### Returns

*the onMouseUp event.*

---

## getOnSelect() Method

```
public String getOnSelect()
```

Gets the onSelect javascript event.

### Returns

*the onSelect event.*

---

## getPage() Method

```
public String getPage()
```

Gets the module–relative path, starting with a slash character, of the image to be displayed by this rendered tag.

### Returns

*the module–relative path.*

---

## getRolloverImage() Method

```
public String getRolloverImage()
```

Gets the roll–over image of the ImageAnchor.

### Returns

*the rollover image.*

---

## getSrc() Method

```
public String getSrc()
```

Gets the image source URI.

## Returns

*the source URI.*

---

## getStyle() Method

```
public String getStyle()
```

Gets the style of the rendered html tag.

## Returns

*the style.*

---

## getStyleClass() Method

```
public String getStyleClass()
```

Gets the style class of the rendered html tag.

## Returns

*the style class.*

---

## getTabIndex() Method

```
public String getTabIndex()
```

Gets the tabIndex of the rendered html tag.

## Returns

*the tabIndex.*

---

## getTagId() Method

```
public String getTagId()
```

Return the ID of the imageButton. The id may be rewritten by the container (such as a portal) to make sure it is unique. JavaScript may lookup the actual id of the element by looking it up in the netui\_names table written into the HTML.

## Returns

*the tagId.*

---



## getTagName() Method

```
public String getTagName()
```

Return the name of the Tag.

### *Overrides*

```
AbstractBaseTag.getTagName()
```

---

## getValue() Method

```
public String getValue()
```

Returns the value of the ImageButton.

### **Returns**

*the ImageButton value.*

---

## getVspace() Method

```
public String getVspace()
```

Gets the vertical spacing around the image.

### **Returns**

*the vertical spacing.*

---

## getWidth() Method

```
public String getWidth()
```

Gets the image width.

### **Returns**

*the image width.*

---

## localRelease() Method

```
protected void localRelease()
```

Release any acquired resources.

### *Overrides*

`AbstractBaseTag.localRelease()`

---

## setAccessKey(String) Method

```
public void setAccessKey(String accessKey)
```

Sets the `accessKey` attribute value. This should key value of the keyboard navigation key. It is recommended not to use the following values because there are often used by browsers `A`, `C`, `E`, `F`, `G`, `H`, `V`, `left arrow`, and `right arrow`.

### Parameters

*accessKey*  
– the `accessKey` value.

---

## setAlign(String) Method

```
public void setAlign(String align)
```

Sets the property to specify where to align the image.

### Parameters

*align*  
– the image alignment.

---

## setAlt(String) Method

```
public void setAlt(String alt)
```

Sets the property to specify the alt text of the image.

### Parameters

*alt*  
– the image alignment.

---

## setAttribute(String, String) Method

```
public void setAttribute(String name,  
                        String value)  
    throws JspException
```

Set an attribute value. The `name` represents the name of the attribute. The `value` represents the value and may contain a netui expression. This method may result in errors being generated. This requires that the tag buffer its body and write attributes in the end tag. For the `imagebutton` tag it is not legal to set the `id`, `name`, `src`, or `value` attributes with this method.

## Parameters

*name*

The name of the attribute. This value may not be null or the empty string.

*value*

The value of the attribute. This may contain a netui expression.

## Exceptions

*JspException*

A JspException may be thrown if there is an error setting the attribute.

---

## setEnabled(String) Method

```
public void setEnabled(String disabled)
```

Sets the enable stat of this ImageButton.

### Parameters

*disabled*

– true or false

---

## setHeight(String) Method

```
public void setHeight(String height)
```

Sets the image height.

### Parameters

*height*

– the height.

---

## setHspace(String) Method

```
public void setHspace(String hspace)
```

Sets the the horizontal spacing around the image.

### Parameters

*hspace*

– the horizontal spacing.

---

## setId(String) Method

```
public void setId(String id)
```

Sets the imageButton id.

### Overrides

```
TagSupport.setId(String)
```

### Parameters

*id*

– the imageBtton id. **The id attribute is deprecated**, use the tagId attribute instead. The id attribute of the rendered HTML tag. You should use the tagId attribute which will encode the id correctly for containers such as the Portal which needs to rewrite id's to make them unique.

---

## setOnBlur(String) Method

```
public void setOnBlur(String onblur)
```

Sets the onBlur javascript event.

### Parameters

*onblur*

– the onBlur event.

---

## setOnChange(String) Method

```
public void setOnChange(String onchange)
```

Sets the onChange javascript event.

### Parameters

*onchange*

– the onChange event.

---

## setOnClick(String) Method

```
public void setOnClick(String onclick)
```

Sets the onClick javascript event.

## Parameters

*onclick*

– the onClick event.

---

## setOnDbClick(String) Method

```
public void setOnDbClick(String ondblick)
```

Sets the onDbClick javascript event.

## Parameters

*ondblick*

– the onDbClick event.

---

## setOnFocus(String) Method

```
public void setOnFocus(String onblur)
```

Sets the onFocus javascript event.

## Parameters

*onblur*

– the onFocus event.

---

## setOnKeyDown(String) Method

```
public void setOnKeyDown(String onkeydown)
```

Sets the onKeyDown javascript event.

## Parameters

*onkeydown*

– the onKeyDown event.

---

## setOnKeyPress(String) Method

```
public void setOnKeyPress(String onkeypress)
```

Sets the onKeyPress javascript event.

## Parameters

*onkeypress*

– the onKeyPress event.

---

## setOnKeyUp(String) Method

```
public void setOnKeyUp(String onkeyup)
```

Sets the onKeyUp javascript event.

## Parameters

*onkeyup*

– the onKeyUp event.

---

## setOnMouseDown(String) Method

```
public void setOnMouseDown(String onmousedown)
```

Sets the onMouseDown javascript event.

## Parameters

*onmousedown*

– the onMouseDown event.

---

## setOnMouseMove(String) Method

```
public void setOnMouseMove(String onmousemove)
```

Sets the onMouseMove javascript event.

## Parameters

*onmousemove*

– the onMouseMove event.

---

## setOnMouseOut(String) Method

```
public void setOnMouseOut(String onmouseout)
```

Sets the onMouseOut javascript event.

## Parameters

*onmouseout*

– the onMouseOut event.

---

## setOnMouseOver(String) Method

```
public void setOnMouseOver(String onmouseover)
```

Sets the onMouseOver javascript event.

## Parameters

*onmouseover*

– the onMouseOver event.

---

## setOnMouseUp(String) Method

```
public void setOnMouseUp(String onmouseup)
```

Sets the onMouseUp javascript event.

## Parameters

*onmouseup*

– the onMouseUp event.

---

## setOnSelect(String) Method

```
public void setOnSelect(String onselect)
```

Sets the onSelect javascript event.

## Parameters

*onselect*

– the onSelect event.

---

## setPage(String) Method

```
public void setPage(String page)
```

Sets the module–relative path, starting with a slash character, of the image to be displayed by this rendered tag.

**Parameters***page*

– the module–relative path.

**setRolloverImage(String) Method**

```
public void setRolloverImage(String rolloverImage)
```

Sets the roll–over image of the ImageAnchor.

**Parameters***rolloverImage*

– the rollover image.

**setSrc(String) Method**

```
public void setSrc(String src)
```

Sets the image source URI.

**Parameters***src*

– the source URI.

**setStyle(String) Method**

```
public void setStyle(String style)
```

Sets the style of the rendered html tag.

**Parameters***style*

– the html style.

**setStyleClass(String) Method**

```
public void setStyleClass(String styleClass)
```

Sets the style class of the rendered html tag.



## Parameters

*styleClass*  
– the html style class.

---

## setTabIndex(String) Method

```
public void setTabIndex(String tabIndex)
```

Sets the tabIndex of the rendered html tag.

## Parameters

*tabindex*  
– the tab index.

---

## setTagId(String) Method

```
public void setTagId(String tagId)
```

Set the ID of the anchor.

## Parameters

*tagId*  
– the tagId.

---

## setValue(String) Method

```
public void setValue(String value)
```

Set the value of the ImageButton.

## Parameters

*value*  
– the value of the ImageButton.

---

## setVspace(String) Method

```
public void setVspace(String vspace)
```

Sets the vertical spacing around the image.

## Parameters

*vspace*

– the vertical spacing.

---

## setWidth(String) Method

```
public void setWidth(String width)
```

Sets the image width.

## Parameters

*width*

– the image width.

---

# Label Class

*public class* **Label**

extends `AbstractBaseTag`

implements `Formattable`, `IAttributeConsumer`

Generates styled text span based on an expression or literal value.

Attribute Descriptions			
Attribute	Required	Runtime Expression Evaluation	Data Bindable
<b>dataformatas</b>	No	No	No
	The data format of the Label		
<b>defaultValue</b>	No	No	Read Only
	A literal default value that will be used when an expression results in a NULL value. If the default value is an expression, the any defined formatters will be applied to the default value. If the defaultValue is not an expression, the value will be output without applying formatters. If the default value is being output and contains a XML Entity, such as "&nbsp;", then the default value is output without applying any escaping to it. The result default value acts like a literal string.		
<b>escapeWhiteSpaceForHtml</b>	No	No	No
	Flag that will set an escape for white space. The default value is false.		
<b>id</b>	No	No	No
	<b>The id attribute has been deprecated; instead, use tagId.</b> The id of the span surrounding the Label text. You should use tagId which will encode the id correctly for containers such as the Portal which rewrite IDs to make them unique.		
<b>tagId</b>	No	No	No
	The id of this label. The real id written to the HTML stream may be changed by the container		

	(Portal), but may be looked up with the <code>tagId</code> value through the JavaScript method <code>getNetuiTagName( tagId, tag )</code> method written by the <code>&lt;netui:html&gt;</code> tag.		
<b>value</b>	Yes	No	Read Only
	A literal value or an expression that resolves to a bean property.		

If the resulting value to be output is the " " String, it will output as the value "&nbsp;".

## Related Topics

`<netui:label>` Tag Sample  
Content

---

### *Hierarchy*

```

Object
  TagSupport
    BodyTagSupport
      AbstractBaseTag
        Label
  
```

### *All Implemented Interfaces*

BodyTag, Formattable, IAttributeConsumer, IterationTag, Serializable, Tag

---

## Field Summary

protected String

### ***dataformatas***

The data format of the  
Label.

protected String

### ***value***

The text of the Label.

### **Fields from `com.bea.wlw.netui.tags.AbstractBaseTag`**

ATTR\_GENERAL, ATTR\_GENERAL\_EXPRESSION, ATTR\_JAVASCRIPT, ATTR\_STYLE, CLASS, ID, JAVASCRIPT\_STATUS, NAME, NETUI\_UNIQUE\_CNT, ONCLICK, ONDBLCLICK, ONKEYDOWN, ONKEYPRESS, ONKEYUP, ONMOUSEDOWN, ONMOUSEMOVE, ONMOUSEOUT, ONMOUSEOVER, ONMOUSEUP, STYLE, TABINDEX

### **Fields from `javax.servlet.jsp.tagext.BodyTagSupport`**

bodyContent

**Fields from `javax.servlet.jsp.tagext.TagSupport`**`id`, `pageContext`**Fields from interface `javax.servlet.jsp.tagext.BodyTag`**`EVAL_BODY_BUFFERED`, `EVAL_BODY_TAG`**Fields from interface `javax.servlet.jsp.tagext.IterationTag`**`EVAL_BODY_AGAIN`**Fields from interface `javax.servlet.jsp.tagext.Tag`**`EVAL_BODY_INCLUDE`, `EVAL_PAGE`, `SKIP_BODY`, `SKIP_PAGE`

## Constructor Summary

*`Label()`*

## Method Summary

```

public
    void addFormatter(FormatTag.Formatter formatter)
        Adds a FormatTag.Formatter to the Label's set of formatters

public
    int doEndTag()
        Render the label.

public
    int doStartTag()
        Prepare the label formatters.

public
    String getDataformatas()
        Gets the data formatters of the Label.

public
    String getDefaultValue()
        Return the default value of this Label.

public
    String getId()
        Return the ID of the label.

public
    String getOnClick()
        Gets the onClick javascript event.

```

```

public getOnDbClick()
String      Gets the onDbClick javascript event.

public
String getOnKeyDown()
           Gets the onKeyDown javascript event.

public
String getOnKeyPress()
           Gets the onKeyPress javascript event.

public
String getOnKeyUp()
           Gets the onKeyUp javascript event.

public
String getOnMouseDown()
           Gets the onMouseDown javascript event.

public
String getOnMouseMove()
           Gets the onMouseMove javascript event.

public
String getOnMouseOut()
           Gets the onMouseOut javascript event.

public
String getOnMouseOver()
           Gets the onMouseOver javascript event.

public
String getOnMouseUp()
           Gets the onMouseUp javascript event.

public
String getStyle()
           Gets the style of the rendered html tag.

public
String getStyleClass()
           Gets the style class of the rendered html tag.

public
String getTabIndex()
           Gets the tabIndex of the rendered html tag.

public
String getTagId()
           Return the ID of the <span> that contains the label text.

public
String getTagName()
           Return the name of the Tag.

public
String getValue()
           Gets the text of the Label.

public
boolean isEscapeWhiteSpaceForHtml()
           Gets the boolean value indicating if we are escaping the whitespace characters in

```

HTML.

```

protected
    void localRelease()
        Release any acquired resources.

public
    void setAttribute(String name, String value)
        Set an attribute value.

public
    void setDataformatas(String dataformatas)
        Sets the data format of the Label.

public
    void setDefaultValue(String defaultValue)
        Set the default value of this Label.

public
    void setEscapeWhiteSpaceForHtml(boolean escapeWhiteSpace)
        Sets a boolean flag indicating if we will escape white space for HTML.

public
    void setId(String id)
        Set the ID of the label.

public
    void setOnClick(String onclick)
        Sets the onClick javascript event.

public
    void setOnDbClick(String ondblclick)
        Sets the onDbClick javascript event.

public
    void setOnKeyDown(String onkeydown)
        Sets the onKeyDown javascript event.

public
    void setOnKeyPress(String onkeypress)
        Sets the onKeyPress javascript event.

public
    void setOnKeyUp(String onkeyup)
        Sets the onKeyUp javascript event.

public
    void setMouseDown(String onmousedown)
        Sets the onMouseDown javascript event.

public
    void setMouseMove(String onmousemove)
        Sets the onMouseMove javascript event.

public
    void setMouseOut(String onmouseout)
        Sets the onMouseOut javascript event.

public
    void setMouseOver(String onmouseover)
        Sets the onMouseOver javascript event.

```

```

public setOnMouseUp(String onMouseup)
    void          Sets the onMouseUp javascript event.
public
    void setStyle(String style)
                Sets the style of the rendered html tag.
public
    void setStyleClass(String styleClass)
                Sets the style class of the rendered html tag.
public
    void setTabIndex(String tabIndex)
                Sets the tabIndex of the rendered html tag.
public
    void setTagId(String tagId)
                Set the ID of the label.
public
    void setValue(String value)
                Sets the text of the Label.

```

### Methods from **com.bea.wlw.netui.tags.AbstractBaseTag**

```

addTagIdMapping, evaluateAttributeToString, filter, getJavaScriptUtils,
getNearestForm, getScriptReporter, getUserLocale, registerTagError,
release, renderAttribute, rewriteName, write

```

### Methods from **javax.servlet.jsp.tagext.BodyTagSupport**

```

doAfterBody, doInitBody, getBodyContent, getPreviousOut, release,
setBodyContent

```

### Methods from **javax.servlet.jsp.tagext.TagSupport**

```

doAfterBody, doEndTag, doStartTag, findAncestorWithClass, getParent,
getValue, getValues, release, removeValue, setPageContext, setParent,
setValue

```

### Methods from class **java.lang.Object**

```

clone, equals, finalize, getClass, hashCode, notify, notifyAll,
toString, wait, wait, wait

```

### Methods from interface **javax.servlet.jsp.tagext.BodyTag**

```

doInitBody, setBodyContent

```

### Methods from interface **com.bea.wlw.netui.tags.html.Formattable**

```

addFormatter

```

### Methods from interface **com.bea.wlw.netui.tags.IAttributeConsumer**



`setAttribute`

**Methods from interface `javax.servlet.jsp.tagext.IterationTag`**

`doAfterBody`

**Methods from interface `javax.servlet.jsp.tagext.Tag`**

`doEndTag`, `doStartTag`, `getParent`, `release`, `setPageContext`, `setParent`

## Field Detail

### **dataformatas**

`protected String dataformatas`

The data format of the Label.

---

### **value**

`protected String value`

The text of the Label.

## Constructor Detail

### **Label**

`public Label()`

## Method Detail

### **addFormatter(`FormatTag.Formatter`) Method**

`public void addFormatter(FormatTag.Formatter formatter)`

Adds a `FormatTag.Formatter` to the Label's set of formatters

#### **Parameters**

*formatter*

– a `FormatTag.Formatter` added by a child `FormatTag`.

---

## doEndTag() Method

```
public int doEndTag()  
    throws JspException
```

Render the label.

### *Overrides*

```
BodyTagSupport.doEndTag( )
```

### Exceptions

*JspException*  
if a JSP exception has occurred

---

## doStartTag() Method

```
public int doStartTag()  
    throws JspException
```

Prepare the label formatters.

### *Overrides*

```
BodyTagSupport.doStartTag( )
```

### Exceptions

*JspException*  
if a JSP exception has occurred

---

## getDataformatas() Method

```
public String getDataformatas()
```

Gets the data formatters of the Label.

### Returns

*the data format.*

---

## getDefaultValue() Method

```
public String getDefaultValue()
```

Return the default value of this Label.

**Returns**

*the Label's default value*

---

**getId() Method**

```
public String getId()
```

Return the ID of the label.

**Overrides**

```
TagSupport.getId()
```

**Returns**

*the ID.*

---

**getOnClick() Method**

```
public String getOnClick()
```

Gets the onClick javascript event.

**Returns**

*the onClick event.*

---

**getOnDbClick() Method**

```
public String getOnDbClick()
```

Gets the onDbClick javascript event.

**Returns**

*the onDbClick event.*

---

**getOnKeyDown() Method**

```
public String getOnKeyDown()
```

Gets the onKeyDown javascript event.

## Returns

*the onKeyDown event.*

---

## getOnKeyPress() Method

```
public String getOnKeyPress()
```

Gets the onKeyPress javascript event.

## Returns

*the onKeyPress event.*

---

## getOnKeyUp() Method

```
public String getOnKeyUp()
```

Gets the onKeyUp javascript event.

## Returns

*the onKeyUp event.*

---

## getOnMouseDown() Method

```
public String getOnMouseDown()
```

Gets the onMouseDown javascript event.

## Returns

*the onMouseDown event.*

---

## getOnMouseMove() Method

```
public String getOnMouseMove()
```

Gets the onMouseMove javascript event.

## Returns

*the onMouseMove event.*

---

## getOnMouseOut() Method

```
public String getOnMouseOut()
```

Gets the onMouseOut javascript event.

### Returns

*the onMouseOut event.*

---

## getOnMouseOver() Method

```
public String getOnMouseOver()
```

Gets the onMouseOver javascript event.

### Returns

*the onMouseOver event.*

---

## getOnMouseUp() Method

```
public String getOnMouseUp()
```

Gets the onMouseUp javascript event.

### Returns

*the onMouseUp event.*

---

## getStyle() Method

```
public String getStyle()
```

Gets the style of the rendered html tag.

### Returns

*the style.*

---

## getStyleClass() Method

```
public String getStyleClass()
```

Gets the style class of the rendered html tag.

**Returns**

*the style class.*

---

**getTabIndex() Method**

```
public String getTabIndex()
```

Gets the tabIndex of the rendered html tag.

**Returns**

*the tabindex.*

---

**getTagId() Method**

```
public String getTagId()
```

Return the ID of the <span> that contains the label text. The id may be rewritten by the container (such as a portal) to make sure it is unique. JavaScript may lookup the actual id of the element by looking it up in the netui\_names table written into the HTML.

**Returns**

*the ID.*

---

**getTagName() Method**

```
public String getTagName()
```

Return the name of the Tag.

**Overrides**

```
AbstractBaseTag.getTagName()
```

---

**getValue() Method**

```
public String getValue()
```

Gets the text of the Label.

**Returns**

*the text value*

---

## isEscapeWhiteSpaceForHtml() Method

```
public boolean isEscapeWhiteSpaceForHtml()
```

Gets the boolean value indicating if we are escaping the whitespace characters in HTML.

### Returns

*the state of white space handling*

---

## localRelease() Method

```
protected void localRelease()
```

Release any acquired resources.

### Overrides

```
AbstractBaseTag.localRelease()
```

---

## setAttribute(String, String) Method

```
public void setAttribute(String name,  
                        String value)  
    throws JspException
```

Set an attribute value. The `name` represents the name of the attribute. The `value` represents the value and may contain a netui expression. This method may result in errors being generated. This requires that the tag buffer its body and write attributes in the end tag. For the label tag it is not legal to set the `id` or `name` attributes with this method.

### Parameters

*name*

The name of the attribute. This value may not be null or the empty string.

*value*

The value of the attribute. This may contain a netui expression.

### Exceptions

*JspException*

A `JspException` may be thrown if there is an error setting the attribute.

---

## setDataformatas(String) Method

```
public void setDataformatas(String dataformatas)
```

Sets the data format of the Label.

## Parameters

*dataformatas*

– the data format.

---

## setDefaultValue(String) Method

```
public void setDefaultValue(String defaultValue)
```

Set the default value of this Label. This can be an expression. If the default value is an expression all formatters will be applied, otherwise the default value will be output without being formatted.

## Parameters

*defaultValue*

– the default value

---

## setEscapeWhiteSpaceForHtml(boolean) Method

```
public void setEscapeWhiteSpaceForHtml(boolean escapeWhiteSpace)
```

Sets a `boolean` flag indicating if we will escape white space for HTML. If this is `true` the white space characters ' ' will be converted into " " and '\n' converted into "\n". The result is that in HTML white space will be represented correctly. If this is `false` then white space will be output as it is found in the value.

## Parameters

*escapeWhiteSpace*

– boolean indicating if we are escaping for white space.

---

## setId(String) Method

```
public void setId(String id)
```

Set the ID of the label.

## Overrides

```
TagSupport.setId(String)
```

## Parameters

*id*

– the ID.

---



## setOnClick(String) Method

```
public void setOnClick(String onclick)
```

Sets the onClick javascript event.

### Parameters

*onclick*

– the onClick event.

---

## setOnDbClick(String) Method

```
public void setOnDbClick(String ondblclick)
```

Sets the onDbClick javascript event.

### Parameters

*ondblclick*

– the onDbClick event.

---

## setOnKeyDown(String) Method

```
public void setOnKeyDown(String onkeydown)
```

Sets the onKeyDown javascript event.

### Parameters

*onkeydown*

– the onKeyDown event.

---

## setOnKeyPress(String) Method

```
public void setOnKeyPress(String onkeypress)
```

Sets the onKeyPress javascript event.

### Parameters

*onkeypress*

– the onKeyPress event.

---

## setOnKeyUp(String) Method

```
public void setOnKeyUp(String onkeyup)
```

Sets the onKeyUp javascript event.

### Parameters

*onkeyup*

– the onKeyUp event.

---

## setOnMouseDown(String) Method

```
public void setOnMouseDown(String onmousedown)
```

Sets the onMouseDown javascript event.

### Parameters

*onmousedown*

– the onMouseDown event.

---

## setOnMouseMove(String) Method

```
public void setOnMouseMove(String onmousemove)
```

Sets the onMouseMove javascript event.

### Parameters

*onmousemove*

– the onMouseMove event.

---

## setOnMouseOut(String) Method

```
public void setOnMouseOut(String onmouseout)
```

Sets the onMouseOut javascript event.

### Parameters

*onmouseout*

– the onMouseOut event.

---

## setOnMouseOver(String) Method

```
public void setOnMouseOver(String onmouseover)
```

Sets the onMouseOver javascript event.

### Parameters

*onmouseover*

– the onMouseOver event.

---

## setOnMouseUp(String) Method

```
public void setOnMouseUp(String onmouseup)
```

Sets the onMouseUp javascript event.

### Parameters

*onmouseup*

– the onMouseUp event.

---

## setStyle(String) Method

```
public void setStyle(String style)
```

Sets the style of the rendered html tag.

### Parameters

*style*

– the html style.

---

## setStyleClass(String) Method

```
public void setStyleClass(String styleClass)
```

Sets the style class of the rendered html tag.

### Parameters

*styleClass*

– the html style class.

---

## setTabIndex(String) Method

```
public void setTabIndex(String tabIndex)
```

Sets the tabIndex of the rendered html tag.

### Parameters

*tabIndex*

– the tab index.

---

## setTagId(String) Method

```
public void setTagId(String tagId)
```

Set the ID of the label.

### Parameters

*tagId*

– the ID.

---

## setValue(String) Method

```
public void setValue(String value)
```

Sets the text of the Label.

### Parameters

*value*

– the Label value or expression.

# Node Class

*public class Node*

extends AbstractBaseTag

Instantiates a `TreeNode` object that will get added to the parent tag (either a `Tree` or another `Node`).

Attribute Descriptions			
Attribute	Required	Runtime Expression Evaluation	Data Bindable
action	No	No	No
	An action invoked by clicking the node.		
expanded	No	No	No
	The node's expanded state.		
icon	No	No	No
	The node's icon URI.		
label	No	No	Read Only
	The label of the Node.		
target	No	No	No
	A window target.		

## Related Topics

<netui:tree> Tag Sample

<netui:tree> Tag: Dynamic Tree Sample

Tree

TreeNode

## Hierarchy

```

Object
  TagSupport
    BodyTagSupport
      AbstractBaseTag
        Node
  
```

## All Implemented Interfaces

BodyTag, IterationTag, Serializable, Tag

## Field Summary

protected String  
     ***action***  
         String

protected String  
     ***expanded***  
         String

protected String  
     ***href***  
         String

protected String  
     ***icon***  
         String

protected String  
     ***label***  
         String

protected TreeNode  
     ***node***  
         TreeNode

protected String  
     ***target***  
         String

### Fields from `com.bea.wlw.netui.tags.AbstractBaseTag`

ATTR\_GENERAL, ATTR\_GENERAL\_EXPRESSION, ATTR\_JAVASCRIPT, ATTR\_STYLE, CLASS, ID, JAVASCRIPT\_STATUS, NAME, NETUI\_UNIQUE\_CNT, ONCLICK, ONDBLCLICK, ONKEYDOWN, ONKEYPRESS, ONKEYUP, ONMOUSEDOWN, ONMOUSEMOVE, ONMOUSEOUT, ONMOUSEOVER, ONMOUSEUP, STYLE, TABINDEX

### Fields from `javax.servlet.jsp.tagext.BodyTagSupport`

bodyContent

### Fields from `javax.servlet.jsp.tagext.TagSupport`

id, pageContext

### Fields from interface `javax.servlet.jsp.tagext.BodyTag`

EVAL\_BODY\_BUFFERED, EVAL\_BODY\_TAG

### Fields from interface `javax.servlet.jsp.tagext.IterationTag`

EVAL\_BODY\_AGAIN

**Fields from interface javax.servlet.jsp.tagext.Tag**

EVAL\_BODY\_INCLUDE, EVAL\_PAGE, SKIP\_BODY, SKIP\_PAGE

## Constructor

## Summary

*Node()*

## Method

## Summary

```

public
    int doEndTag()
        Add this node to the parent.

public
    int doStartTag()
        Instantiate a new TreeNode.

public
    String getAction()
        Returns the Node's action.

public
    String getExpanded()
        Gets if the Node is expanded.

public
    String getHref()
        Gets the Node's href.

public
    String getIcon()
        Gets the Node icon URI.

public
    String getLabel()
        Gets the Node label.

public
    TreeNode getNode()

public
    String getTagName()
        Return the name of the Tag.

public
    String getTarget()
        Returns the window target.

protected
    void localRelease()
        Release any acquired resources.

public

```

```

void registerError(EvalErrorInfo error)
    This method will add an error to the errors begin tracked by this
    tag.

public
String registerTagError(String message)
    This will report an error from a tag.

public
    void setAction(String action)
        Set the Node's action.

public
    void setExpanded(String expanded)
        Sets the Node expansion state.

public
    void setHref(String href)
        Sets the Node's href.

public
    void setIcon(String icon)
        Sets the Node icon URI.

public
    void setLabel(String label)
        Sets the Node label.

public
    void setNode(TreeNode node)

public
    void setTarget(String target)
        Sets the window target.

```

#### Methods from **com.bea.wlw.netui.tags.AbstractBaseTag**

addTagIdMapping, evaluateAttributeToString, filter, filter,  
getJavaScriptUtils, getNearestForm, getScriptReporter, getUserLocale,  
release, renderAttribute, rewriteName, write

#### Methods from **javax.servlet.jsp.tagext.BodyTagSupport**

doAfterBody, doInitBody, getBodyContent, getPreviousOut, release,  
setBodyContent

#### Methods from **javax.servlet.jsp.tagext.TagSupport**

doAfterBody, doEndTag, doStartTag, findAncestorWithClass, getId,  
getParent, getValue, getValues, release, removeValue, setId,  
setPageContext, setParent, setValue

#### Methods from class **java.lang.Object**

clone, equals, finalize, getClass, hashCode, notify, notifyAll,  
toString, wait, wait, wait



**Methods from interface `javax.servlet.jsp.tagext.BodyTag`**

`doInitBody`, `setBodyContent`

**Methods from interface `javax.servlet.jsp.tagext.IterationTag`**

`doAfterBody`

**Methods from interface `javax.servlet.jsp.tagext.Tag`**

`doEndTag`, `doStartTag`, `getParent`, `release`, `setPageContext`, `setParent`

## Field Detail

### **action**

`protected String action`

---

### **expanded**

`protected String expanded`

---

### **href**

`protected String href`

---

### **icon**

`protected String icon`

---

### **label**

`protected String label`

---

### **node**

`protected TreeNode node`

---

### **target**

`protected String target`

## Constructor Detail

### Node

```
public Node()
```

## Method Detail

### doEndTag() Method

```
public int doEndTag()  
    throws JspException
```

Add this node to the parent.

#### *Overrides*

```
BodyTagSupport.doEndTag( )
```

### Exceptions

*JspException*  
if a JSP exception has occurred

---

### doStartTag() Method

```
public int doStartTag()  
    throws JspException
```

Instantiate a new TreeNode.

#### *Overrides*

```
BodyTagSupport.doStartTag( )
```

### Exceptions

*JspException*  
if a JSP exception has occurred

---

### getAction() Method

```
public String getAction()
```

Returns the Node's action.

## Returns

*the action.*

---

## getExpanded() Method

```
public String getExpanded()
```

Gets if the Node is expanded.

## Returns

*the expanded state.*

---

## getHref() Method

```
public String getHref()
```

Gets the Node's href.

## Returns

*the href.*

---

## getIcon() Method

```
public String getIcon()
```

Gets the Node icon URI.

## Returns

*the icon URI.*

---

## getLabel() Method

```
public String getLabel()
```

Gets the Node label.

## Returns

*the label.*

---

## getNode() Method

```
public TreeNode getNode()
```

---

## getTagName() Method

```
public String getTagName()
```

Return the name of the Tag.

### *Overrides*

```
AbstractBaseTag.getTagName()
```

---

## getTarget() Method

```
public String getTarget()
```

Returns the window target.

### **Returns**

*the window target.*

---

## localRelease() Method

```
protected void localRelease()
```

Release any acquired resources.

### *Overrides*

```
AbstractBaseTag.localRelease()
```

---

## registerError(EvalErrorInfo) Method

```
public void registerError(EvalErrorInfo error)
```

*Description copied from AbstractBaseTag.registerError(EvalErrorInfo)*

This method will add an error to the errors begin tracked by this tag.

### *Overrides*

```
AbstractBaseTag.registerError(EvalErrorInfo)
```

---

## registerTagError(String) Method

```
public String registerTagError(String message)
```

*Description copied from AbstractBaseTag.registerTagError(String)*

This will report an error from a tag. The error will contain a message. If error reporting is turned off, the message will be returned and the caller should throw a JspException to report the error.

### Overrides

```
AbstractBaseTag.registerTagError(String)
```

### Parameters

*message*

– the message to register with the error

### Returns

*null*

---

## setAction(String) Method

```
public void setAction(String action)
```

Set the Node's action.

### Parameters

*action*

– the action.

---

## setExpanded(String) Method

```
public void setExpanded(String expanded)
```

Sets the Node expansion state.

### Parameters

*expanded*

– the expanded state.

---

## setHref(String) Method

```
public void setHref(String href)
```

Sets the Node's href.

**Parameters***href*

– the href

**setIcon(String) Method**

```
public void setIcon(String icon)
```

Sets the Node icon URI.

**Parameters***icon*

– the icon URI

**setLabel(String) Method**

```
public void setLabel(String label)
```

Sets the Node label.

**Parameters***label*

– the label.

**setNode(TreeNode) Method**

```
public void setNode(TreeNode node)
```

**setTarget(String) Method**

```
public void setTarget(String target)
```

Sets the window target.

**Parameters***target*

– the window target.

# Parameter Class

*public class **Parameter***

extends AbstractBaseTag

Writes a URL parameter to a URL on its parent tag. The parent tag must implement URLParams.

Attribute Descriptions			
Attribute	Required	Runtime Expression Evaluation	Data Bindable
name	Yes	No	No
	The name of the parameter.		
value	Yes	No	Read Only
	A literal value or an expression that resolves to a bean property.		

## Related Topics

<netui:parameter> Tag Sample  
ParameterMap

---

## Hierarchy

```

Object
  TagSupport
    BodyTagSupport
      AbstractBaseTag
        Parameter
  
```

## All Implemented Interfaces

BodyTag, IterationTag, Serializable, Tag

---

## Field Summary

protected String

**name**

String

protected Object

**paramValue**

Object

protected String

**value**  
String

#### Fields from `com.bea.wlw.netui.tags.AbstractBaseTag`

ATTR\_GENERAL, ATTR\_GENERAL\_EXPRESSION, ATTR\_JAVASCRIPT, ATTR\_STYLE, CLASS, ID, JAVASCRIPT\_STATUS, NAME, NETUI\_UNIQUE\_CNT, ONCLICK, ONDBLCLICK, ONKEYDOWN, ONKEYPRESS, ONKEYUP, ONMOUSEDOWN, ONMOUSEMOVE, ONMOUSEOUT, ONMOUSEOVER, ONMOUSEUP, STYLE, TABINDEX

#### Fields from `javax.servlet.jsp.tagext.BodyTagSupport`

bodyContent

#### Fields from `javax.servlet.jsp.tagext.TagSupport`

id, pageContext

#### Fields from interface `javax.servlet.jsp.tagext.BodyTag`

EVAL\_BODY\_BUFFERED, EVAL\_BODY\_TAG

#### Fields from interface `javax.servlet.jsp.tagext.IterationTag`

EVAL\_BODY\_AGAIN

#### Fields from interface `javax.servlet.jsp.tagext.Tag`

EVAL\_BODY\_INCLUDE, EVAL\_PAGE, SKIP\_BODY, SKIP\_PAGE

## Constructor Summary

*Parameter()*

## Method Summary

```
public
    int doStartTag()
        Add the URL parameter to the Parameter's
        parent.

public
    String getName()
        Gets the name of the URL parameter.

public
    String getTagName()
        Return the name of the Tag.
```



```

    public getValue()
        String          Gets the value of the URL parameter.
protected
    void localRelease()
        Release any acquired resources.

    public
        void setName(String name)
            Sets the name of the URL parameter.

    public
        void setValue(String value)
            Sets the value of the URL parameter.

```

#### Methods from **com.bea.wlw.netui.tags.AbstractBaseTag**

```

addTagIdMapping, evaluateAttributeToString, filter, filter,
getJavaScriptUtils, getNearestForm, getScriptReporter, getUserLocale,
registerTagError, release, renderAttribute, rewriteName, write

```

#### Methods from **javax.servlet.jsp.tagext.BodyTagSupport**

```

doAfterBody, doEndTag, doInitBody, getBodyContent, getPreviousOut,
release, setBodyContent

```

#### Methods from **javax.servlet.jsp.tagext.TagSupport**

```

doAfterBody, doEndTag, doStartTag, findAncestorWithClass, getId,
getParent, getValue, getValues, release, removeValue, setId,
setPageContext, setParent, setValue

```

#### Methods from class **java.lang.Object**

```

clone, equals, finalize, getClass, hashCode, notify, notifyAll,
toString, wait, wait, wait

```

#### Methods from interface **javax.servlet.jsp.tagext.BodyTag**

```

doInitBody, setBodyContent

```

#### Methods from interface **javax.servlet.jsp.tagext.IterationTag**

```

doAfterBody

```

#### Methods from interface **javax.servlet.jsp.tagext.Tag**

```

doEndTag, doStartTag, getParent, release, setPageContext, setParent

```

## Field Detail

### name

`protected String name`

---

### paramValue

`protected Object paramValue`

---

### value

`protected String value`

## Constructor Detail

### Parameter

`public Parameter()`

## Method Detail

### doStartTag() Method

`public int doStartTag()  
 throws JspException`

Add the URL parameter to the Parameter's parent.

#### *Overrides*

`BodyTagSupport.doStartTag()`

### Exceptions

*JspException*  
if a JSP exception has occurred

---

### getName() Method

`public String getName()`

Gets the name of the URL parameter.

### Returns

*the parameter name.*

---

## getTagName() Method

```
public String getTagName()
```

Return the name of the Tag.

### Overrides

```
AbstractBaseTag.getTagName()
```

---

## getValue() Method

```
public String getValue()
```

Gets the value of the URL parameter.

### Returns

*the parameter value.*

---

## localRelease() Method

```
protected void localRelease()
```

Release any acquired resources.

### Overrides

```
AbstractBaseTag.localRelease()
```

---

## setName(String) Method

```
public void setName(String name)
```

Sets the name of the URL parameter.

### Parameters

*name*

– the parameter name.

---

## **setValue(String) Method**

```
public void setValue(String value)
```

Sets the value of the URL parameter. This can be an expression.

### **Parameters**

*value*

– the parameter value.

# ParameterMap Class

*public class* **ParameterMap**

extends AbstractBaseTag

Writes each in a map of URL parameters to a URL on its parent tag. The parent tag must implement URLParams.

Attribute Descriptions			
Attribute	Required	Runtime Expression Evaluation	Data Bindable
map	Yes	No	Read Only
	An expression pointing to a map of parameters.		

### Related Topics

Parameter  
<netui:parameterMap> Tag Sample

---

### Hierarchy

Object  
  TagSupport  
    BodyTagSupport  
      AbstractBaseTag  
        ParameterMap

### All Implemented Interfaces

BodyTag, IterationTag, Serializable, Tag

---

## Field Summary

protected String  
    **map**  
        String  
protected Map  
    **paramMap**  
        Map

Fields from com.bea.wlw.netui.tags.AbstractBaseTag

ATTR\_GENERAL, ATTR\_GENERAL\_EXPRESSION, ATTR\_JAVASCRIPT, ATTR\_STYLE, CLASS, ID, JAVASCRIPT\_STATUS, NAME, NETUI\_UNIQUE\_CNT, ONCLICK, ONDBLCLICK, ONKEYDOWN, ONKEYPRESS, ONKEYUP, ONMOUSEDOWN, ONMOUSEMOVE, ONMOUSEOUT, ONMOUSEOVER, ONMOUSEUP, STYLE, TABINDEX

**Fields from `javax.servlet.jsp.tagext.BodyTagSupport`**

bodyContent

**Fields from `javax.servlet.jsp.tagext.TagSupport`**

id, pageContext

**Fields from interface `javax.servlet.jsp.tagext.BodyTag`**

EVAL\_BODY\_BUFFERED, EVAL\_BODY\_TAG

**Fields from interface `javax.servlet.jsp.tagext.IterationTag`**

EVAL\_BODY\_AGAIN

**Fields from interface `javax.servlet.jsp.tagext.Tag`**

EVAL\_BODY\_INCLUDE, EVAL\_PAGE, SKIP\_BODY, SKIP\_PAGE

## Constructor Summary

*ParameterMap()*

## Method Summary

```
public
    int doStartTag()
        Add each parameter in the URL parameter map to the Parameter's
        parent.

public
    String getMap()
        Gets the map expression.

public
    String getTagName()
        Return the name of the Tag.

protected
    void localRelease()
        Release any acquired resources.
```

```
public setMap(String map)
void          Sets the map expression.
```

#### Methods from `com.bea.wlw.netui.tags.AbstractBaseTag`

```
addTagIdMapping, evaluateAttributeToString, filter, filter,
getJavaScriptUtils, getNearestForm, getScriptReporter, getUserLocale,
registerTagError, release, renderAttribute, rewriteName, write
```

#### Methods from `javax.servlet.jsp.tagext.BodyTagSupport`

```
doAfterBody, doEndTag, doInitBody, getBodyContent, getPreviousOut,
release, setBodyContent
```

#### Methods from `javax.servlet.jsp.tagext.TagSupport`

```
doAfterBody, doEndTag, doStartTag, findAncestorWithClass, getId,
getParent, getValue, getValues, release, removeValue, setId,
setPageContext, setParent, setValue
```

#### Methods from class `java.lang.Object`

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll,
toString, wait, wait, wait
```

#### Methods from interface `javax.servlet.jsp.tagext.BodyTag`

```
doInitBody, setBodyContent
```

#### Methods from interface `javax.servlet.jsp.tagext.IterationTag`

```
doAfterBody
```

#### Methods from interface `javax.servlet.jsp.tagext.Tag`

```
doEndTag, doStartTag, getParent, release, setPageContext, setParent
```

## Field Detail

### map

```
protected String map
```

---

### paramMap

```
protected Map paramMap
```

## Constructor Detail

### ParameterMap

```
public ParameterMap()
```

## Method Detail

### doStartTag() Method

```
public int doStartTag()  
    throws JspException
```

Add each parameter in the URL parameter map to the Parameter's parent.

#### *Overrides*

```
BodyTagSupport.doStartTag()
```

### Exceptions

*JspException*  
if a JSP exception has occurred

---

### getMap() Method

```
public String getMap()
```

Gets the map expression.

### Returns

*the map expression.*

---

### getTagName() Method

```
public String getTagName()
```

Return the name of the Tag.

#### *Overrides*

```
AbstractBaseTag.getTagName()
```

---



## localRelease() Method

```
protected void localRelease()
```

Release any acquired resources.

### *Overrides*

```
AbstractBaseTag.localRelease( )
```

---

## setMap(String) Method

```
public void setMap(String map)  
    throws JspException
```

Sets the map expression.

### Parameters

*map*  
– the map expression.

### Exceptions

*JspException*

# RadioButtonGroup Class

*public class* **RadioButtonGroup**

extends `OptionsDataSourceTag`

Groups a collection of `RadioButtonOptions`, and handles databinding of their values. If `RadioButtonGroup` uses any `Format` tags, it must have those tags come before above any nested `RadioButtonOption` tags.

Attribute Descriptions			
Attribute	Required	Runtime Expression Evaluation	Data Bindable
<b>dataSource</b>	Yes	No	Read / Write
	The name of a form bean property or an expression that contains the <code>RadioButtonGroup</code> state		
<b>defaultValue</b>	No	No	Read Only
	A literal value or an expression that is used to set the <code>RadioButtonGroup</code> 's initial state		
<b>disabled</b>	No	No	Read Only
	A boolean value which if true will cause the <code>RadioButtonGroup</code> to be disabled.		
<b>labelStyle</b>	No	No	No
	The style of the label for each contained <code>RadioButtonOption</code>		
<b>labelStyleClass</b>	No	No	No
	The style class of the label for each contained <code>RadioButtonOption</code>		
<b>optionsDataSource</b>	No	No	Read Only
	An expression which is used to generate a set of dynamic <code>RadioButtonOptions</code>		
<b>tagId</b>	No	No	No
	The ID of this <code>radioButtonGroup</code> used by the enclosing html tag to get the tag's name for javascript purposes		

## Related Topics

## <netui:radioButton> Tag Samples

### RadioButtonOption

---

#### *Hierarchy*

```

Object
  TagSupport
    BodyTagSupport
      AbstractBaseTag
        DataSourceTag
          DefaultableDataSourceTag
            OptionsDataSourceTag
              RadioButtonGroup

```

#### *All Implemented Interfaces*

BodyTag, IterationTag, Serializable, Tag

---

## Field Summary

```

protected String
    defaultRadio
        String

protected String
    labelStyle
        String

protected String
    labelStyleClass
        String

protected String
    match
        The actual values we will match against, calculated in doStartTag().

    public
    static RADIOBUTTONGROUP_KEY
        final String String

protected String
    saveBody
        The saved body content of this tag.

```

#### Fields from `com.bea.wlw.netui.tags.AbstractBaseTag`

ATTR\_GENERAL, ATTR\_GENERAL\_EXPRESSION, ATTR\_JAVASCRIPT, ATTR\_STYLE, CLASS, ID, JAVASCRIPT\_STATUS, NAME, NETUI\_UNIQUE\_CNT, ONCLICK, ONDBLCLICK, ONKEYDOWN, ONKEYPRESS, ONKEYUP, ONMOUSEDOWN, ONMOUSEMOVE, ONMOUSEOUT, ONMOUSEOVER, ONMOUSEUP, STYLE, TABINDEX

#### Fields from `javax.servlet.jsp.tagext.BodyTagSupport`

bodyContent

#### Fields from `javax.servlet.jsp.tagext.TagSupport`

id, pageContext

#### Fields from interface `javax.servlet.jsp.tagext.BodyTag`

EVAL\_BODY\_BUFFERED, EVAL\_BODY\_TAG

#### Fields from interface `javax.servlet.jsp.tagext.IterationTag`

EVAL\_BODY\_AGAIN

#### Fields from interface `javax.servlet.jsp.tagext.Tag`

EVAL\_BODY\_INCLUDE, EVAL\_PAGE, SKIP\_BODY, SKIP\_PAGE

## Constructor Summary

*RadioButtonGroup()*

## Method Summary

```

public
    int doAfterBody()
        Save the body content of the RadioButtonGroup.

public
    int doEndTag()
        Render the set of RadioButtonOptions.

public
    int doStartTag()
        Determine the match for the RadioButtonGroup

protected
    Object evaluateOptionsDataSource()
        This method will evaluate the optionsDataSource and return an
        object.

public
    String getDisabled()
        Returns the boolean value or expression indicating the disable state
        of the RadioButtonGroup.

public
    String getLabelStyle()
        Return the label style for each contained CheckBoxOption..

```

```

public getLabelStyleClass()
String      Return the label style class for each contained CheckBoxOption..

public
String getStyle()
            Gets the style of the rendered html tag.

public
String getStyleClass()
            Gets the style class of the rendered html tag.

public
String getTagId()
            Return the ID of the radiobutton group.

public
String getTagName()
            Return the name of the Tag.

protected
void localRelease()
            Release any acquired resources.

public
String qualifyDataSource()

public
void setDisabled(String disabled)
            Set the disable state either with the literal "true" or "false" or with an
            expression.

public
void setLabelStyle(String labelStyle)
            Set the label style for each contained CheckBoxOption..

public
void setLabelStyleClass(String labelStyleClass)
            Set the label style class for each contained CheckBoxOption..

public
void setStyle(String style)
            Sets the style of the rendered html tag.

public
void setStyleClass(String styleClass)
            Sets the style class of the rendered html tag.

public
void setTagId(String tagId)
            Set the ID of the radiobutton group.

```

#### Methods from **com.bea.wlw.netui.tags.OptionsDataSourceTag**

getOptionsDataSource, setOptionsDataSource

#### Methods from **com.bea.wlw.netui.tags.DefaultableDataSourceTag**

getDefaultValue, localRelease, setDefaultValue

#### Methods from **com.bea.wlw.netui.tags.DataSourceTag**

`getDataSource, localRelease, setDataSource`

**Methods from `com.bea.wlw.netui.tags.AbstractBaseTag`**

`addTagIdMapping, evaluateAttributeToString, filter, filter,  
getJavaScriptUtils, getNearestForm, getScriptReporter, getUserLocale,  
localRelease, registerTagError, release, renderAttribute, rewriteName,  
write`

**Methods from `javax.servlet.jsp.tagext.BodyTagSupport`**

`doInitBody, getBodyContent, getPreviousOut, release, setBodyContent`

**Methods from `javax.servlet.jsp.tagext.TagSupport`**

`doAfterBody, doEndTag, doStartTag, findAncestorWithClass, getId,  
getParent, getValue, getValues, release, removeValue, setId,  
setPageContext, setParent, setValue`

**Methods from class `java.lang.Object`**

`clone, equals, finalize, getClass, hashCode, notify, notifyAll,  
toString, wait, wait, wait`

**Methods from interface `javax.servlet.jsp.tagext.BodyTag`**

`doInitBody, setBodyContent`

**Methods from interface `javax.servlet.jsp.tagext.IterationTag`**

`doAfterBody`

**Methods from interface `javax.servlet.jsp.tagext.Tag`**

`doEndTag, doStartTag, getParent, release, setPageContext, setParent`

## Field Detail

### defaultRadio

`protected String defaultRadio`

---

### labelStyle

`protected String labelStyle`

---

## labelStyleClass

```
protected String labelStyleClass
```

---

## match

```
protected String match
```

The actual values we will match against, calculated in doStartTag().

---

## RADIOBUTTONGROUP\_KEY

```
public static final String RADIOBUTTONGROUP_KEY
```

---

## saveBody

```
protected String saveBody
```

The saved body content of this tag.

## Constructor Detail

### RadioButtonGroup

```
public RadioButtonGroup()
```

## Method Detail

### doAfterBody() Method

```
public int doAfterBody()  
    throws JspException
```

Save the body content of the RadioButtonGroup.

#### *Overrides*

```
BodyTagSupport.doAfterBody()
```

### Exceptions

*JspException*

if a JSP exception has occurred

## doEndTag() Method

```
public int doEndTag()  
    throws JspException
```

Render the set of RadioButtonOptions.

### *Overrides*

```
BodyTagSupport.doEndTag( )
```

### Exceptions

*JspException*  
if a JSP exception has occurred

---

## doStartTag() Method

```
public int doStartTag()  
    throws JspException
```

Determine the match for the RadioButtonGroup

### *Overrides*

```
BodyTagSupport.doStartTag( )
```

### Exceptions

*JspException*  
if a JSP exception has occurred

---

## evaluateOptionsDataSource() Method

```
protected Object evaluateOptionsDataSource()  
    throws JspException
```

This method will evaluate the optionsDataSource and return an object. If the object return from evaluating the expression results in a Map that is returned. Otherwise we will return an iterator. This routine will always return either a valid Iterator or Map.

### *Overrides*

```
OptionsDataSourceTag.evaluateOptionsDataSource( )
```

### Exceptions

*JspException*

---



## getDisabled() Method

```
public String getDisabled()
```

Returns the boolean value or expression indicating the disable state of the RadioButtonGroup.

### Returns

*the disabled state (true or false) or an expression*

---

## getLabelStyle() Method

```
public String getLabelStyle()
```

Return the label style for each contained CheckBoxOption..

### Returns

*the label style*

---

## getLabelStyleClass() Method

```
public String getLabelStyleClass()
```

Return the label style class for each contained CheckBoxOption..

### Returns

*the label style*

---

## getStyle() Method

```
public String getStyle()
```

Gets the style of the rendered html tag.

### Returns

*the style.*

---

## getStyleClass() Method

```
public String getStyleClass()
```

Gets the style class of the rendered html tag.

**Returns**

*the style class.*

---

**getTagId() Method**

```
public String getTagId()
```

Return the ID of the radiobutton group.

**Returns**

*the ID.*

---

**getTagName() Method**

```
public String getTagName()
```

Return the name of the Tag.

**Overrides**

```
AbstractBaseTag.getTagName()
```

---

**localRelease() Method**

```
protected void localRelease()
```

Release any acquired resources.

**Overrides**

```
OptionsDataSourceTag.localRelease()
```

---

**qualifyDataSource() Method**

```
public String qualifyDataSource()  
    throws JspException
```

**Exceptions**

*JspException*

---

**setDisabled(String) Method**

```
public void setDisabled(String disabled)
```

Set the disable state either with the literal "true" or "false" or with an expression.

### Parameters

*disabled*

– true or false or an expression

---

## setLabelStyle(String) Method

```
public void setLabelStyle(String labelStyle)
```

Set the label style for each contained CheckBoxOption..

### Parameters

*labelStyle*

– the label style

---

## setLabelStyleClass(String) Method

```
public void setLabelStyleClass(String labelStyleClass)
```

Set the label style class for each contained CheckBoxOption..

### Parameters

*labelStyleClass*

– the label style

---

## setStyle(String) Method

```
public void setStyle(String style)
```

Sets the style of the rendered html tag.

### Parameters

*style*

– the html style.

---

## setStyleClass(String) Method

```
public void setStyleClass(String styleClass)
```

Sets the style class of the rendered html tag.

## Parameters

*styleClass*  
– the html style class.

---

## setTagId(String) Method

```
public void setTagId(String tagId)
```

Set the ID of the radiobutton group.

## Parameters

*tagId*  
– the ID.

# RadioButtonOption Class

*public class **RadioButtonOption***

extends AbstractBaseTag  
implements IAttributeConsumer

A radio button whose state is determined by its enclosing RadioButtonGroup.

Attribute Descriptions			
Attribute	Required	Runtime Expression Evaluation	Data Bindable
accessKey	No	No	No
	Define the keyboard navigation key for the element.		
alt	No	No	Read Only
	Define the alt attribute of the <Input> element.		
labelStyle	No	No	No
	The style of the label for each contained CheckBoxOption		
labelStyleClass	No	No	No
	The style class of the label for each contained CheckBoxOption		
tagId	No	No	No
	The id of this control. The real id written to the HTML stream may be changed by the container (Portal), but may be looked up with the 'tagId' value through the JavaScript method <code>getNetuiTagName( tagId, tag )</code> method written by the <netui:html> tag.		
value	Yes	No	Read Only
	An expression to a bean property, or a literal, which is the value of this option		

## Related Topics

<netui:radionButton> Tag Samples  
RadioButtonGroup

## Hierarchy

```

Object
TagSupport
BodyTagSupport
AbstractBaseTag
RadioButtonOption

```

### *All Implemented Interfaces*

BodyTag, IAttributeConsumer, IterationTag, Serializable, Tag

---

## Field Summary

```

    public
    static CHECKED
final String      String
protected String

    labelStyle
      String

    public
    static LABELSTYLE
final String      String
protected String

    labelStyleClass
      String

    public
    static LABELSTYLECLASS
final String      String
protected String

    text
      The message text to be displayed to the user for this tag (if any)

    public
    static TYPE
final String      String
protected String

    value
      The server value for this radiobutton, also used to match against the current property
      value to determine whether this radiobutton should be marked as selected.

    public
    static VALUE
final String      String

```

### Fields from `com.bea.wlw.netui.tags.AbstractBaseTag`

ATTR\_GENERAL, ATTR\_GENERAL\_EXPRESSION, ATTR\_JAVASCRIPT, ATTR\_STYLE, CLASS, ID, JAVASCRIPT\_STATUS, NAME, NETUI\_UNIQUE\_CNT, ONCLICK, ONDBLCLICK, ONKEYDOWN, ONKEYPRESS, ONKEYUP, ONMOUSEDOWN, ONMOUSEMOVE, ONMOUSEOUT, ONMOUSEOVER, ONMOUSEUP, STYLE, TABINDEX

**Fields from `javax.servlet.jsp.tagext.BodyTagSupport`**`bodyContent`**Fields from `javax.servlet.jsp.tagext.TagSupport`**`id`, `pageContext`**Fields from interface `javax.servlet.jsp.tagext.BodyTag`**`EVAL_BODY_BUFFERED`, `EVAL_BODY_TAG`**Fields from interface `javax.servlet.jsp.tagext.IterationTag`**`EVAL_BODY_AGAIN`**Fields from interface `javax.servlet.jsp.tagext.Tag`**`EVAL_BODY_INCLUDE`, `EVAL_PAGE`, `SKIP_BODY`, `SKIP_PAGE`

## Constructor Summary

*`RadioButtonOption()`*

## Method Summary

```

public
    int doAfterBody()
        Process the body text of this tag (if any).

public
    int doEndTag()
        Process the end of this tag.

public
    int doStartTag()
        Process the start of this tag.

public
    String getAccessKey()
        Gets the accessKey attribute value.

public
    String getAlt()
        Gets the alt attribute on the generate <input tag.

public
    String getLabelStyle()
        Return the label style for each contained RadioButtonOption.

```

```
public getLabelStyleClass()
String      Return the label style class for each contained
             RadioButtonOption.

public
String getOnBlur()
             Gets the onBlur javascript event.

public
String getOnChange()
             Gets the onChange javascript event.

public
String getOnClick()
             Gets the onClick javascript event.

public
String getOnDbClick()
             Gets the onDbClick javascript event.

public
String getOnFocus()
             Gets the onFocus javascript event.

public
String getOnKeyDown()
             Gets the onKeyDown javascript event.

public
String getOnKeyPress()
             Gets the onKeyPress javascript event.

public
String getOnKeyUp()
             Gets the onKeyUp javascript event.

public
String getOnMouseDown()
             Gets the onMouseDown javascript event.

public
String getOnMouseMove()
             Gets the onMouseMove javascript event.

public
String getOnMouseOut()
             Gets the onMouseOut javascript event.

public
String getOnMouseOver()
             Gets the onMouseOver javascript event.

public
String getOnMouseUp()
             Gets the onMouseUp javascript event.

public
String getOnSelect()
             Gets the onSelect javascript event.

public
String getStyle()
```



Gets the style of the rendered html tag.

```
public
String getStyleClass()
```

Gets the style class of the rendered html tag.

```
public
String getTabIndex()
```

Gets the tabIndex of the rendered html tag.

```
public
String getTagId()
```

Return the ID of the tag.

```
public
String getTagName()
```

Return the name of the Tag.

```
public
String getValue()
```

Return the value of this RadioButtonOption.

```
protected
void localRelease()
```

Release any acquired resources.

```
public
void setAccessKey(String accessKey)
```

Sets the accessKey attribute value.

```
public
void setAlt(String alt)
```

Sets the alt attribute value.

```
public
void setAttribute(String name, String value)
```

Set an attribute value.

```
public
void setLabelStyle(String labelStyle)
```

Set the label style for each contained RadioButtonOption.

```
public
void setLabelStyleClass(String labelStyleClass)
```

Set the label style class for each contained RadioButtonOption.

```
public
void setOnBlur(String onblur)
```

Sets the onBlur javascript event.

```
public
void setOnChange(String onchange)
```

Sets the onChange javascript event.

```
public
void setOnClick(String onclick)
```

Sets the onClick javascript event.

```
public
void setOnDbClick(String ondblclick)
```

Sets the onDbClick javascript event.

```

public setOnFocus(String onBlur)
    void          Sets the onFocus javascript event.
public
    void setOnKeyDown(String onkeydown)
        Sets the onKeyDown javascript event.
public
    void setOnKeyPress(String onkeypress)
        Sets the onKeyPress javascript event.
public
    void setOnKeyUp(String onkeyup)
        Sets the onKeyUp javascript event.
public
    void setOnMouseDown(String onmousedown)
        Sets the onMouseDown javascript event.
public
    void setOnMouseMove(String onmousemove)
        Sets the onMouseMove javascript event.
public
    void setOnMouseOut(String onmouseout)
        Sets the onMouseOut javascript event.
public
    void setOnMouseOver(String onmouseover)
        Sets the onMouseOver javascript event.
public
    void setOnMouseUp(String onmouseup)
        Sets the onMouseUp javascript event.
public
    void setOnSelect(String onselect)
        Sets the onSelect javascript event.
public
    void setStyle(String style)
        Sets the style of the rendered html tag.
public
    void setStyleClass(String styleClass)
        Sets the style class of the rendered html tag.
public
    void setTabIndex(String tabindex)
        Sets the tabIndex of the rendered html tag.
public
    void setTagId(String tagId)
        Set the ID of the tag.
public
    void setValue(String value)
        Set the value of this RadioButtonOption.

```

Methods from **com.bea.wlw.netui.tags.AbstractBaseTag**

`addTagIdMapping, evaluateAttributeToString, filter, filter,  
getJavaScriptUtils, getNearestForm, getScriptReporter, getUserLocale,  
registerTagError, release, renderAttribute, rewriteName, write`

**Methods from `javax.servlet.jsp.tagext.BodyTagSupport`**

`doInitBody, getBodyContent, getPreviousOut, release, setBodyContent`

**Methods from `javax.servlet.jsp.tagext.TagSupport`**

`doAfterBody, doEndTag, doStartTag, findAncestorWithClass, getId,  
getParent, getValue, getValues, release, removeValue, setId,  
setPageContext, setParent, setValue`

**Methods from class `java.lang.Object`**

`clone, equals, finalize, getClass, hashCode, notify, notifyAll,  
toString, wait, wait, wait`

**Methods from interface `javax.servlet.jsp.tagext.BodyTag`**

`doInitBody, setBodyContent`

**Methods from interface `com.bea.wlw.netui.tags.IAttributeConsumer`**

`setAttribute`

**Methods from interface `javax.servlet.jsp.tagext.IterationTag`**

`doAfterBody`

**Methods from interface `javax.servlet.jsp.tagext.Tag`**

`doEndTag, doStartTag, getParent, release, setPageContext, setParent`

## Field Detail

### CHECKED

`public static final String CHECKED`

---

### labelStyle

`protected String labelStyle`

---

## LABELSTYLE

```
public static final String LABELSTYLE
```

---

## labelStyleClass

```
protected String labelStyleClass
```

---

## LABELSTYLECLASS

```
public static final String LABELSTYLECLASS
```

---

## text

```
protected String text
```

The message text to be displayed to the user for this tag (if any)

---

## TYPE

```
public static final String TYPE
```

---

## value

```
protected String value
```

The server value for this radiobutton, also used to match against the current property value to determine whether this radiobutton should be marked as selected.

---

## VALUE

```
public static final String VALUE
```

## Constructor Detail

### RadioButtonOption

```
public RadioButtonOption()
```

## Method Detail

### doAfterBody() Method

```
public int doAfterBody()  
    throws JspException
```

Process the body text of this tag (if any).

#### *Overrides*

```
BodyTagSupport.doAfterBody( )
```

#### Exceptions

*JspException*  
if a JSP exception has occurred

---

### doEndTag() Method

```
public int doEndTag()  
    throws JspException
```

Process the end of this tag.

#### *Overrides*

```
BodyTagSupport.doEndTag( )
```

#### Exceptions

*JspException*  
if a JSP exception has occurred

---

### doStartTag() Method

```
public int doStartTag()  
    throws JspException
```

Process the start of this tag.

#### *Overrides*

```
BodyTagSupport.doStartTag( )
```

#### Exceptions

*JspException*  
if a JSP exception has occurred

---

## getAccessKey() Method

```
public String getAccessKey()
```

Gets the accessKey attribute value.

### Returns

*the accessKey value.*

---

## getAlt() Method

```
public String getAlt()
```

Gets the alt attribute on the generate <input tag.

### Returns

*the alt value.*

---

## getLabelStyle() Method

```
public String getLabelStyle()
```

Return the label style for each contained RadioButtonOption.

### Returns

*the label style*

---

## getLabelStyleClass() Method

```
public String getLabelStyleClass()
```

Return the label style class for each contained RadioButtonOption.

### Returns

*the label style*

---

## getOnBlur() Method

```
public String getOnBlur()
```

Gets the onBlur javascript event.

## Returns

*the onBlur event.*

---

## getOnChange() Method

```
public String getOnChange()
```

Gets the onChange javascript event.

## Returns

*the onChange event.*

---

## getOnClick() Method

```
public String getOnClick()
```

Gets the onClick javascript event.

## Returns

*the onClick event.*

---

## getOnDbClick() Method

```
public String getOnDbClick()
```

Gets the onDbClick javascript event.

## Returns

*the onDbClick event.*

---

## getOnFocus() Method

```
public String getOnFocus()
```

Gets the onFocus javascript event.

## Returns

*the onFocus event.*

---

## getOnKeyDown() Method

```
public String getOnKeyDown()
```

Gets the onKeyDown javascript event.

### Returns

*the onKeyDown event.*

---

## getOnKeyPress() Method

```
public String getOnKeyPress()
```

Gets the onKeyPress javascript event.

### Returns

*the onKeyPress event.*

---

## getOnKeyUp() Method

```
public String getOnKeyUp()
```

Gets the onKeyUp javascript event.

### Returns

*the onKeyUp event.*

---

## getOnMouseDown() Method

```
public String getOnMouseDown()
```

Gets the onMouseDown javascript event.

### Returns

*the onMouseDown event.*

---

## getOnMouseMove() Method

```
public String getOnMouseMove()
```

Gets the onMouseMove javascript event.



## Returns

*the onMouseMove event.*

---

## getOnMouseOut() Method

```
public String getOnMouseOut()
```

Gets the onMouseOut javascript event.

## Returns

*the onMouseOut event.*

---

## getOnMouseOver() Method

```
public String getOnMouseOver()
```

Gets the onMouseOver javascript event.

## Returns

*the onMouseOver event.*

---

## getOnMouseUp() Method

```
public String getOnMouseUp()
```

Gets the onMouseUp javascript event.

## Returns

*the onMouseUp event.*

---

## getOnSelect() Method

```
public String getOnSelect()
```

Gets the onSelect javascript event.

## Returns

*the onSelect event.*

---

## getStyle() Method

```
public String getStyle()
```

Gets the style of the rendered html tag.

### Returns

*the style.*

---

## getStyleClass() Method

```
public String getStyleClass()
```

Gets the style class of the rendered html tag.

### Returns

*the style class.*

---

## getTabIndex() Method

```
public String getTabIndex()
```

Gets the tabIndex of the rendered html tag.

### Returns

*the tabindex.*

---

## getTagId() Method

```
public String getTagId()
```

Return the ID of the tag. The id may be rewritten by the container (such as a portal) to make sure it is unique. JavaScript may lookup the actual id of the element by looking it up in the netui\_names table written into the HTML.

### Returns

*the tagId.*

---

## getTagName() Method

```
public String getTagName()
```

Return the name of the Tag.

#### ***Overrides***

`AbstractBaseTag.getTagName()`

---

### **getValue() Method**

`public String getValue()`

Return the value of this RadioButtonOption.

#### **Returns**

*the RadioButtonOption value*

---

### **localRelease() Method**

`protected void localRelease()`

Release any acquired resources.

#### ***Overrides***

`AbstractBaseTag.localRelease()`

---

### **setAccessKey(String) Method**

`public void setAccessKey(String accessKey)`

Sets the accessKey attribute value. This should key value of the keyboard navigation key. It is recommended not to use the following values because there are often used by browsers A, C, E, F, G, H, V, left arrow, and right arrow.

#### **Parameters**

*accessKey*  
– the accessKey value.

---

### **setAlt(String) Method**

`public void setAlt(String alt)`

Sets the alt attribute value.

**Parameters***alt*

– the alt value.

**setAttribute(String, String) Method**

```
public void setAttribute(String name,
                        String value)
    throws JspException
```

Set an attribute value. The *name* represents the name of the attribute. The *value* represents the value and may contain a netui expression. This method may result in errors being generated. This requires that the tag buffer its body and write attributes in the end tag. For the radiobuttonoption tag it is not legal to set the *id*, *name*, *type*, *value*, or *checked* attributes with this method.

**Parameters***name*

The name of the attribute. This value may not be null or the empty string.

*value*

The value of the attribute. This may contain a netui expression.

**Exceptions***JspException*

A JspException may be thrown if there is an error setting the attribute.

**setLabelStyle(String) Method**

```
public void setLabelStyle(String labelStyle)
```

Set the label style for each contained RadioButtonOption. The label style here will override a labelStyle at the RadioButtonGroup level.

**Parameters***labelStyle*

– the label style

**setLabelStyleClass(String) Method**

```
public void setLabelStyleClass(String labelStyleClass)
```

Set the label style class for each contained RadioButtonOption. The label style class here will override a labelStyleClass at the RadioButtonGroup level.

**Parameters**

*labelStyleClass*

– the label style

---

**setOnBlur(String) Method**

```
public void setOnBlur(String onblur)
```

Sets the onBlur javascript event.

**Parameters**

*onblur*

– the onBlur event.

---

**setOnChange(String) Method**

```
public void setOnChange(String onchange)
```

Sets the onChange javascript event.

**Parameters**

*onchange*

– the onChange event.

---

**setOnClick(String) Method**

```
public void setOnClick(String onclick)
```

Sets the onClick javascript event.

**Parameters**

*onclick*

– the onClick event.

---

**setOnDbClick(String) Method**

```
public void setOnDbClick(String ondblclick)
```

Sets the onDbClick javascript event.

## Parameters

*ondblclick*

– the onDbClick event.

---

## setOnFocus(String) Method

```
public void setOnFocus(String onblur)
```

Sets the onFocus javascript event.

## Parameters

*onblur*

– the onFocus event.

---

## setOnKeyDown(String) Method

```
public void setOnKeyDown(String onkeydown)
```

Sets the onKeyDown javascript event.

## Parameters

*onkeydown*

– the onKeyDown event.

---

## setOnKeyPress(String) Method

```
public void setOnKeyPress(String onkeypress)
```

Sets the onKeyPress javascript event.

## Parameters

*onkeypress*

– the onKeyPress event.

---

## setOnKeyUp(String) Method

```
public void setOnKeyUp(String onkeyup)
```

Sets the onKeyUp javascript event.

## Parameters

*onkeyup*

– the onKeyUp event.

---

## setOnMouseDown(String) Method

```
public void setOnMouseDown(String onmousedown)
```

Sets the onMouseDown javascript event.

## Parameters

*onmousedown*

– the onMouseDown event.

---

## setOnMouseMove(String) Method

```
public void setOnMouseMove(String onmousemove)
```

Sets the onMouseMove javascript event.

## Parameters

*onmousemove*

– the onMouseMove event.

---

## setOnMouseOut(String) Method

```
public void setOnMouseOut(String onmouseout)
```

Sets the onMouseOut javascript event.

## Parameters

*onmouseout*

– the onMouseOut event.

---

## setOnMouseOver(String) Method

```
public void setOnMouseOver(String onmouseover)
```

Sets the onMouseOver javascript event.

## Parameters

*onmouseover*

– the onMouseOver event.

---

## setOnMouseUp(String) Method

```
public void setOnMouseUp(String onmouseup)
```

Sets the onMouseUp javascript event.

## Parameters

*onmouseup*

– the onMouseUp event.

---

## setOnSelect(String) Method

```
public void setOnSelect(String onselect)
```

Sets the onSelect javascript event.

## Parameters

*onselect*

– the onSelect event.

---

## setStyle(String) Method

```
public void setStyle(String style)
```

Sets the style of the rendered html tag.

## Parameters

*style*

– the html style.

---

## setStyleClass(String) Method

```
public void setStyleClass(String styleClass)
```

Sets the style class of the rendered html tag.



## Parameters

*styleClass*

– the html style class.

---

## setTabIndex(String) Method

```
public void setTabIndex(String tabIndex)
```

Sets the tabIndex of the rendered html tag.

## Parameters

*tabindex*

– the tab index.

---

## setTagId(String) Method

```
public void setTagId(String tagId)
```

Set the ID of the tag.

## Parameters

*tagId*

– the tagId.

---

## setValue(String) Method

```
public void setValue(String value)
```

Set the value of this RadioButtonOption.

## Parameters

*value*

– the RadioButtonOption value

---

# RewriteName Class

*public class RewriteName*

extends `AbstractBaseTag`

Allow a tag name, typically either an `id` or `name` attribute, to participate in URL rewriting. Some containers such as WebLogic Portal rewrite names so they are unique. This tag will cause the name to be made available from the `getNetuiTagName( tagId, tag )` JavaScript function (which is written to the page by the `<netui:html>` tag).

Attribute Descriptions			
Attribute	Required	Runtime Expression Evaluation	Data Bindable
name	Yes	No	No
	The value of the name which will be rewritten. The real name will be output and may be looked up using 'name'.		
resultId	No	No	No
	A name that will cause the real name to be stored into the page context under this name.		

## Related Topics

`<netui:rewrite...>` Tag Samples  
[RewriteURL](#)

---

## Hierarchy

```

Object
  TagSupport
    BodyTagSupport
      AbstractBaseTag
        RewriteName
  
```

## All Implemented Interfaces

`BodyTag`, `IterationTag`, `Serializable`, `Tag`

---

## Field Summary

Fields from `com.bea.wlw.netui.tags.AbstractBaseTag`

ATTR\_GENERAL, ATTR\_GENERAL\_EXPRESSION, ATTR\_JAVASCRIPT, ATTR\_STYLE, CLASS, ID, JAVASCRIPT\_STATUS, NAME, NETUI\_UNIQUE\_CNT, ONCLICK, ONDBLCLICK, ONKEYDOWN, ONKEYPRESS, ONKEYUP, ONMOUSEDOWN, ONMOUSEMOVE, ONMOUSEOUT, ONMOUSEOVER, ONMOUSEUP, STYLE, TABINDEX

### Fields from `javax.servlet.jsp.tagext.BodyTagSupport`

`bodyContent`

### Fields from `javax.servlet.jsp.tagext.TagSupport`

`id`, `pageContext`

### Fields from interface `javax.servlet.jsp.tagext.BodyTag`

`EVAL_BODY_BUFFERED`, `EVAL_BODY_TAG`

### Fields from interface `javax.servlet.jsp.tagext.IterationTag`

`EVAL_BODY_AGAIN`

### Fields from interface `javax.servlet.jsp.tagext.Tag`

`EVAL_BODY_INCLUDE`, `EVAL_PAGE`, `SKIP_BODY`, `SKIP_PAGE`

## Constructor Summary

*`RewriteName()`*

## Method Summary

```
public
    int doStartTag()
        Pass the name attribute to the URLRewriter and output the returned value.
```

```
public
    String getName()
        Gets the name of that should be rewritten.
```

```
public
    String getResultId()
        Gets the resultId.
```

```
public
    String getTagName()
        Return the name of the Tag.
```

```
protected
    void localRelease()
```

Release any acquired resources.

```
public
    void setName(String name)
        Sets the name to be rewritten.

public
    void setResultId(String resultId)
        Sets the resultId which, if non-null, will store the real name in the page
        context under the resultId name;
```

### Methods from `com.bea.wlw.netui.tags.AbstractBaseTag`

`addTagIdMapping`, `evaluateAttributeToString`, `filter`, `filter`,  
`getJavaScriptUtils`, `getNearestForm`, `getScriptReporter`, `getUserLocale`,  
`registerTagError`, `release`, `renderAttribute`, `rewriteName`, `write`

### Methods from `javax.servlet.jsp.tagext.BodyTagSupport`

`doAfterBody`, `doEndTag`, `doInitBody`, `getBodyContent`, `getPreviousOut`,  
`release`, `setBodyContent`

### Methods from `javax.servlet.jsp.tagext.TagSupport`

`doAfterBody`, `doEndTag`, `doStartTag`, `findAncestorWithClass`, `getId`,  
`getParent`, `getValue`, `getValues`, `release`, `removeValue`, `setId`,  
`setPageContext`, `setParent`, `setValue`

### Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`,  
`toString`, `wait`, `wait`, `wait`

### Methods from interface `javax.servlet.jsp.tagext.BodyTag`

`doInitBody`, `setBodyContent`

### Methods from interface `javax.servlet.jsp.tagext.IterationTag`

`doAfterBody`

### Methods from interface `javax.servlet.jsp.tagext.Tag`

`doEndTag`, `doStartTag`, `getParent`, `release`, `setPageContext`, `setParent`

## Constructor Detail

### RewriteName

```
public RewriteName()
```

## Method Detail

### doStartTag() Method

```
public int doStartTag()  
    throws JspException
```

Pass the name attribute to the URLRewriter and output the returned value. Updates the HTML tag to output the mapping.

#### *Overrides*

```
BodyTagSupport.doStartTag( )
```

### Exceptions

*JspException*  
if a JSP exception has occurred

---

### getName() Method

```
public String getName()
```

Gets the name of that should be rewritten.

### Returns

*the parameter name.*

---

### getResultId() Method

```
public String getResultId()
```

Gets the resultId. This is the name that the real name that will be used to store the real name in the page context.

### Returns

*the resultId*

---

### getTagName() Method

```
public String getTagName()
```

Return the name of the Tag.

#### *Overrides*

```
AbstractBaseTag.getTagName( )
```

## localRelease() Method

```
protected void localRelease()
```

Release any acquired resources.

### *Overrides*

```
AbstractBaseTag.localRelease()
```

---

## setName(String) Method

```
public void setName(String name)
```

Sets the name to be rewritten.

### Parameters

*name*

– the parameter name.

---

## setResultId(String) Method

```
public void setResultId(String resultId)
```

Sets the resultId which, if non-null, will store the real name in the page context under the resultId name;

### Parameters

*resultId*

– the parameter name.

# RewriteURL Class

*public class RewriteURL*

extends AbstractBaseTag

Allow a name, typically either an id or name attribute, to participate in URL rewriting. Some containers such as WebLogic Portal rewrite names so they are unique. This tag will cause the name to be made available from `getNetuiTagName( tagId, tag )` JavaScript which is output from the `<netui:html>` tag.

Attribute Descriptions			
Attribute	Required	Runtime Expression Evaluation	Data Bindable
URL	No	No	No
	The value of the URL that will be rewritten.		

## Related Topics

`<netui:rewrite...>` Tag Samples

RewriteName

---

## Hierarchy

```

Object
  TagSupport
    BodyTagSupport
      AbstractBaseTag
        RewriteURL
  
```

## All Implemented Interfaces

BodyTag, IterationTag, Serializable, Tag

---

## Field Summary

protected String

**url**

String

## Fields from com.bea.wlw.netui.tags.AbstractBaseTag

ATTR\_GENERAL, ATTR\_GENERAL\_EXPRESSION, ATTR\_JAVASCRIPT, ATTR\_STYLE, CLASS, ID, JAVASCRIPT\_STATUS, NAME, NETUI\_UNIQUE\_CNT, ONCLICK, ONDBLCLICK, ONKEYDOWN, ONKEYPRESS, ONKEYUP, ONMOUSEDOWN, ONMOUSEMOVE,

ONMOUSEOUT, ONMOUSEOVER, ONMOUSEUP, STYLE, TABINDEX

#### Fields from `javax.servlet.jsp.tagext.BodyTagSupport`

`bodyContent`

#### Fields from `javax.servlet.jsp.tagext.TagSupport`

`id`, `pageContext`

#### Fields from interface `javax.servlet.jsp.tagext.BodyTag`

`EVAL_BODY_BUFFERED`, `EVAL_BODY_TAG`

#### Fields from interface `javax.servlet.jsp.tagext.IterationTag`

`EVAL_BODY_AGAIN`

#### Fields from interface `javax.servlet.jsp.tagext.Tag`

`EVAL_BODY_INCLUDE`, `EVAL_PAGE`, `SKIP_BODY`, `SKIP_PAGE`

## Constructor Summary

***RewriteURL()***

## Method Summary

```
public
    int doStartTag()
        Add the URL parameter to the Parameter's
        parent.

public
    String getTagName()
        Return the name of the Tag.

public
    String getURL()
        Gets the name of that should be rewritten.

protected
    void localRelease()
        Release any acquired resources.

public
    void setURL(String url)
        Sets the name to be rewritten.
```



**Methods from `com.bea.wlw.netui.tags.AbstractBaseTag`**

`addTagIdMapping`, `evaluateAttributeToString`, `filter`, `filter`,  
`getJavaScriptUtils`, `getNearestForm`, `getScriptReporter`, `getUserLocale`,  
`registerTagError`, `release`, `renderAttribute`, `rewriteName`, `write`

**Methods from `javax.servlet.jsp.tagext.BodyTagSupport`**

`doAfterBody`, `doEndTag`, `doInitBody`, `getBodyContent`, `getPreviousOut`,  
`release`, `setBodyContent`

**Methods from `javax.servlet.jsp.tagext.TagSupport`**

`doAfterBody`, `doEndTag`, `doStartTag`, `findAncestorWithClass`, `getId`,  
`getParent`, `getValue`, `getValues`, `release`, `removeValue`, `setId`,  
`setPageContext`, `setParent`, `setValue`

**Methods from class `java.lang.Object`**

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`,  
`toString`, `wait`, `wait`, `wait`

**Methods from interface `javax.servlet.jsp.tagext.BodyTag`**

`doInitBody`, `setBodyContent`

**Methods from interface `javax.servlet.jsp.tagext.IterationTag`**

`doAfterBody`

**Methods from interface `javax.servlet.jsp.tagext.Tag`**

`doEndTag`, `doStartTag`, `getParent`, `release`, `setPageContext`, `setParent`

## Field Detail

### `url`

`protected String url`

## Constructor Detail

### `RewriteURL`

`public RewriteURL()`

## Method Detail

### doStartTag() Method

```
public int doStartTag()  
    throws JspException
```

Add the URL parameter to the Parameter's parent.

#### *Overrides*

```
BodyTagSupport.doStartTag( )
```

### Exceptions

*JspException*  
if a JSP exception has occurred

---

### getTagName() Method

```
public String getTagName()
```

Return the name of the Tag.

#### *Overrides*

```
AbstractBaseTag.getTagName( )
```

---

### getURL() Method

```
public String getURL()
```

Gets the name of that should be rewritten.

### Returns

*the parameter name.*

---

### localRelease() Method

```
protected void localRelease()
```

Release any acquired resources.

#### *Overrides*

```
AbstractBaseTag.localRelease( )
```

---

## **setURL(String) Method**

```
public void setURL(String url)
```

Sets the name to be rewritten.

### **Parameters**

*url*

– the parameter name.

# ScriptContainer Class

*public class ScriptContainer*

extends AbstractBaseTag  
implements ScriptReporter

Acts as a container that will bundle up JavaScript created by other NetUI tags, and output it within a single `<script>` tag. This is especially needed for Portal web applications, because they often cannot rely on having `<html> ... </html>` tags to provide a default container. In a portlet, some JSP pages might be included into other JSP pages. Having redundant `<html> ... </html>` tags in the rendered portlet JSP can result in display problems for some browsers. However, omitting the `<html>` tag (and the container it provides) can result in cluttered code, especially where Javascript appears in the file. To solve this issue, WebLogic Workshop provides the `<netui:scriptContainer>` tag.

Attribute Descriptions			
Attribute	Required	Runtime Expression Evaluation	Data Bindable
scopeID	No	No	No
	The ID that is associated with the code methods.		

## Related Topics

`<netui:scriptContainer>` Tag Sample

---

## Hierarchy

```

Object
  TagSupport
    BodyTagSupport
      AbstractBaseTag
        ScriptContainer
  
```

## All Implemented Interfaces

BodyTag, IterationTag, ScriptReporter, Serializable, Tag

## Direct Known Subclasses

Html

---

## Field Summary

### Fields from `com.bea.wlw.netui.tags.AbstractBaseTag`

ATTR\_GENERAL, ATTR\_GENERAL\_EXPRESSION, ATTR\_JAVASCRIPT, ATTR\_STYLE, CLASS, ID, JAVASCRIPT\_STATUS, NAME, NETUI\_UNIQUE\_CNT, ONCLICK, ONDBLCLICK, ONKEYDOWN, ONKEYPRESS, ONKEYUP, ONMOUSEDOWN, ONMOUSEMOVE, ONMOUSEOUT, ONMOUSEOVER, ONMOUSEUP, STYLE, TABINDEX

### Fields from `javax.servlet.jsp.tagext.BodyTagSupport`

bodyContent

### Fields from `javax.servlet.jsp.tagext.TagSupport`

id, pageContext

### Fields from interface `javax.servlet.jsp.tagext.BodyTag`

EVAL\_BODY\_BUFFERED, EVAL\_BODY\_TAG

### Fields from interface `javax.servlet.jsp.tagext.IterationTag`

EVAL\_BODY\_AGAIN

### Fields from interface `javax.servlet.jsp.tagext.Tag`

EVAL\_BODY\_INCLUDE, EVAL\_PAGE, SKIP\_BODY, SKIP\_PAGE

## Constructor Summary

*ScriptContainer()*

## Method Summary

```
public
    void addScriptCode(String s)
        This method will add Script as top level code that runs when the page is loaded.

public
    void addScriptFunction(String s)
        This method will add Script as a function.

public
    void addTagId(String tagId, String tagName)
```

## Page Flow API Reference

Adds a tagID and tagName to the Html's getId javascript function.

```
public
    int doAfterBody()
        Save any body content of this tag, which will generally be the option(s) representing
        the values displayed to the user.

public
    int doEndTag()
        Write out the body content and report any errors that occurred.

public
    int doStartTag()

public
    String getScopeId()
        return the scopeId associated with the ScriptContainer

public
    String getTagName()
        Returns the name of the Tag.

protected
    void localRelease()
        Release any acquired resources.

public
    String nextTagIdRef(String tagId)

protected
    void processIdMap()

protected
    String replaceTagNameRefs(String content)

public
    String resolveTagIdRef(String tagIdRef)

public
    void setScopeId(String scopeId)
        Set the scopeId associated with the code methods

protected
    void writeScriptBlock()
```

### Methods from `com.bea.wlw.netui.tags.AbstractBaseTag`

addTagIdMapping, evaluateAttributeToString, filter, filter,  
getJavaScriptUtils, getNearestForm, getScriptReporter, getUserLocale,  
registerTagError, release, renderAttribute, rewriteName, write

### Methods from `javax.servlet.jsp.tagext.BodyTagSupport`

doInitBody, getBodyContent, getPreviousOut, release, setBodyContent

### Methods from `javax.servlet.jsp.tagext.TagSupport`

doAfterBody, doEndTag, doStartTag, findAncestorWithClass, getId,  
getParent, getValue, getValues, release, removeValue, setId,  
setPageContext, setParent, setValue

### Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

### Methods from interface `javax.servlet.jsp.tagext.BodyTag`

`doInitBody`, `setBodyContent`

### Methods from interface `javax.servlet.jsp.tagext.IterationTag`

`doAfterBody`

### Methods from interface `com.bea.wlw.netui.tags.ScriptReporter`

`addScriptCode`, `addScriptFunction`, `addTagId`

### Methods from interface `javax.servlet.jsp.tagext.Tag`

`doEndTag`, `doStartTag`, `getParent`, `release`, `setPageContext`, `setParent`

## Constructor Detail

### ScriptContainer

```
public ScriptContainer()
```

## Method Detail

### addScriptCode(String) Method

```
public void addScriptCode(String s)
```

This method will add Script as top level code that runs when the page is loaded.

#### Parameters

*s*  
the text of the script

---

### addScriptFunction(String) Method

```
public void addScriptFunction(String s)
```

This method will add Script as a function.

**Parameters**

*s*  
the text of the function

---

**addTagId(String, String) Method**

```
public void addTagId(String tagId,
                    String tagName)
```

Adds a tagID and tagName to the Html's getId javascript function.

**Parameters**

*tagId*  
– the id of a child tag.

*tagName*  
– the name of a child tag.

---

**doAfterBody() Method**

```
public int doAfterBody()
    throws JspException
```

Save any body content of this tag, which will generally be the option(s) representing the values displayed to the user.

**Overrides**

```
BodyTagSupport.doAfterBody()
```

**Exceptions**

*JspException*  
if a JSP exception has occurred

---

**doEndTag() Method**

```
public int doEndTag()
    throws JspException
```

Write out the body content and report any errors that occurred.

**Overrides**

```
BodyTagSupport.doEndTag()
```



**Exceptions***JspException*if a JSP exception has occurred

---

**doStartTag() Method**

```
public int doStartTag()
    throws JspException
```

*Overrides*`BodyTagSupport.doStartTag()`**Exceptions***JspException*

---

**getScopeId() Method**

```
public String getScopeId()
```

return the scopeId associated with the ScriptContainer

---

**getTagName() Method**

```
public String getTagName()
```

Returns the name of the Tag.

*Overrides*`AbstractBaseTag.getTagName()`

---

**localRelease() Method**

```
protected void localRelease()
```

Release any acquired resources.

*Overrides*`AbstractBaseTag.localRelease()`

---

**nextTagIdRef(String) Method**

```
public String nextTagIdRef(String tagId)
```

---

## processIdMap() Method

```
protected void processIdMap()
```

---

## replaceTagNameRefs(String) Method

```
protected String replaceTagNameRefs(String content)
    throws JspException
```

### Exceptions

*JspException*

---

## resolveTagIdRef(String) Method

```
public String resolveTagIdRef(String tagIdRef)
```

---

## setScopeId(String) Method

```
public void setScopeId(String scopeId)
```

Set the scopeId associated with the code methods

---

## writeScriptBlock() Method

```
protected void writeScriptBlock()
    throws JspException
```

### Exceptions

*JspException*

## Select Class

*public class Select*

extends `OptionsDataSourceTag`  
implements `Formattable`, `IAttributeConsumer`

Renders a select containing a set of `SelectOptions`. `Select` binds to an `Iterator` of `Strings`. If `Select` uses any `Format` tags, it must have those tags come before any nested `SelectOption` tags.

Attribute Descriptions			
Attribute	Required	Runtime Expression Evaluation	Data Bindable
<b>accessKey</b>	No	No	No
	Define the keyboard navigation key for the element.		
<b>dataSource</b>	Yes	No	Read / Write
	The name of a form bean property or an expression that contains the <code>Select</code> state.		
<b>defaultValue</b>	No	No	Read Only
	A literal value or an expression that is used to set the <code>Select</code> 's initial state.		
<b>disabled</b>	No	No	Read Only
	A boolean value which if true will cause the <code>Select</code> to be disabled.		
<b>multiple</b>	No	No	No
	Whether or not multi-selection is enabled.		
<b>nullable</b>	No	No	No
	Whether a (null) option should be added to the bottom of the list. If the <code>Select</code> has <code>multiple=true</code> , (null) won't be shown regardless.		
<b>nullableOptionText</b>	No	No	No
	If <code>nullable</code> is true, then this is optional text for the option. The default is to use the empty string, "", as the value.		
<b>optionsDataSource</b>	No	No	Read Only

	An expression which is used to generate a set of dynamic SelectOption tags.		
<b>tagId</b>	No	No	No
	The ID of this select used by the enclosing html tag to get the tag's name for javascript and focus purposes.		
<b>size</b>	No	No	No
	The number of visible options.		

**Related Topics**

[<netui:select> Tag Sample](#)  
[Dynamically Populating the Options in a Dropdown Sample](#)  
[SelectOption](#)

---

***Hierarchy***

```

Object
  TagSupport
    BodyTagSupport
      AbstractBaseTag
        DataSourceTag
          DefaultableDataSourceTag
            OptionsDataSourceTag
              Select
  
```

***All Implemented Interfaces***

BodyTag, Formattable, IAttributeConsumer, IterationTag, Serializable, Tag

---

## Field

### Summary

```

public
static NULL_VALUE
final String      String
  
```

**Fields from com.bea.wlw.netui.tags.AbstractBaseTag**

ATTR\_GENERAL, ATTR\_GENERAL\_EXPRESSION, ATTR\_JAVASCRIPT, ATTR\_STYLE, CLASS, ID, JAVASCRIPT\_STATUS, NAME, NETUI\_UNIQUE\_CNT, ONCLICK, ONDBLCLICK, ONKEYDOWN, ONKEYPRESS, ONKEYUP, ONMOUSEDOWN, ONMOUSEMOVE, ONMOUSEOUT, ONMOUSEOVER, ONMOUSEUP, STYLE, TABINDEX

**Fields from javax.servlet.jsp.tagext.BodyTagSupport**

bodyContent

**Fields from `javax.servlet.jsp.tagext.TagSupport`**`id`, `pageContext`**Fields from interface `javax.servlet.jsp.tagext.BodyTag`**`EVAL_BODY_BUFFERED`, `EVAL_BODY_TAG`**Fields from interface `javax.servlet.jsp.tagext.IterationTag`**`EVAL_BODY_AGAIN`**Fields from interface `javax.servlet.jsp.tagext.Tag`**`EVAL_BODY_INCLUDE`, `EVAL_PAGE`, `SKIP_BODY`, `SKIP_PAGE`

## Constructor Summary

*Select()*

## Method Summary

```

    public
        void addFormatter(FormatTag.Formatter formatter)
            Adds a FormatTag.Formatter to the Select's set of formatters

    protected
        boolean containsString(List list, String value)
            Return true, if the list contains an Object.toString().equals(value).

    public
        int doAfterBody()
            Save any body content of this tag, which will generally be the option(s) representing
            the values displayed to the user.

    public
        int doEndTag()
            Render the end of this select.

    public
        int doStartTag()
            Render the beginning of this select.

    protected
        Object evaluateDefaultValue()
            Evaluate the defaultValues

    public
        String getAccessKey()
            Gets the accessKey attribute value.

```

```

public getDisabled()
String      Returns the boolean value or expression indicating the disable state of the Select.

public
boolean getMultiple()
           Gets whether multiple selections are allowed.

public
String getNullableOptionText()
           If the nullable option is true, this is the text of that
           option.

public
String getOnBlur()
           Gets the onBlur javascript event.

public
String getOnChange()
           Gets the onChange javascript event.

public
String getOnClick()
           Gets the onClick javascript event.

public
String getOnDbClick()
           Gets the onDbClick javascript event.

public
String getOnFocus()
           Gets the onFocus javascript event.

public
String getOnKeyDown()
           Gets the onKeyDown javascript event.

public
String getOnKeyPress()
           Gets the onKeyPress javascript event.

public
String getOnKeyUp()
           Gets the onKeyUp javascript event.

public
String getOnMouseDown()
           Gets the onMouseDown javascript event.

public
String getOnMouseMove()
           Gets the onMouseMove javascript event.

public
String getOnMouseOut()
           Gets the onMouseOut javascript event.

public
String getOnMouseOver()
           Gets the onMouseOver javascript event.

public
String getOnMouseUp()

```

Gets the onMouseUp javascript event.

```
public
String getOnSelect()
```

Gets the onSelect javascript event.

```
public
String getSize()
```

Gets how many options are displayed.

```
public
String getStyle()
```

Gets the style of the rendered html tag.

```
public
String getStyleClass()
```

Gets the style class of the rendered html tag.

```
public
String getTabIndex()
```

Gets the tabIndex of the rendered html tag.

```
public
String getTagId()
```

Return the ID of the select.

```
public
String getTagName()
```

Return the name of the Tag.

```
public
boolean isMatched(String value)
```

Does the specified value match one of those we are looking for?

```
public
boolean isNullable()
```

Gets whether a null option is desired.

```
protected
void localRelease()
```

Release any acquired resources.

```
public
void setAccessKey(String accessKey)
```

Sets the accessKey attribute value.

```
public
void setAttribute(String name, String value)
```

Set an attribute value.

```
public
void setDisabled(String disabled)
```

Set the disable state either with the literal "true" or "false" or with an expression.

```
public
void setMultiple(boolean multiple)
```

Set whether multiple selections are allowed.

```
public
void setNullable(boolean nullable)
```

Set whether a null option is desired.

```

public setNullableOptionText(String nullableOptionText)
    void          Set the text of the nullable option.
public
    void setOnBlur(String onblur)
        Sets the onBlur javascript event.
public
    void setOnChange(String onchange)
        Sets the onChange javascript event.
public
    void setOnClick(String onclick)
        Sets the onClick javascript event.
public
    void setOnDbClick(String ondblclick)
        Sets the onDbClick javascript event.
public
    void setOnFocus(String onblur)
        Sets the onFocus javascript event.
public
    void setOnKeyDown(String onkeydown)
        Sets the onKeyDown javascript event.
public
    void setOnKeyPress(String onkeypress)
        Sets the onKeyPress javascript event.
public
    void setOnKeyUp(String onkeyup)
        Sets the onKeyUp javascript event.
public
    void setOnMouseDown(String onmousedown)
        Sets the onMouseDown javascript event.
public
    void setOnMouseMove(String onmousemove)
        Sets the onMouseMove javascript event.
public
    void setOnMouseOut(String onmouseout)
        Sets the onMouseOut javascript event.
public
    void setOnMouseOver(String onmouseover)
        Sets the onMouseOver javascript event.
public
    void setOnMouseUp(String onmouseup)
        Sets the onMouseUp javascript event.
public
    void setOnSelect(String onselect)
        Sets the onSelect javascript event.
public
    void setSize(String size)
        Sets how many options are displayed.

```



```

public setStyle(String style)
    void          Sets the style of the rendered html tag.
public
    void setStyleClass(String styleClass)
        Sets the style class of the rendered html tag.
public
    void setTabIndex(String tabIndex)
        Sets the tabIndex of the rendered html tag.
public
    void setTagId(String tagId)
        Set the ID of the select.

```

**Methods from com.bea.wlw.netui.tags.OptionsDataSourceTag**

getOptionsDataSource, setOptionsDataSource

**Methods from com.bea.wlw.netui.tags.DefaultableDataSourceTag**

getDefaultValue, localRelease, setDefaultValue

**Methods from com.bea.wlw.netui.tags.DataSourceTag**

getDataSource, localRelease, setDataSource

**Methods from com.bea.wlw.netui.tags.AbstractBaseTag**

addTagIdMapping, evaluateAttributeToString, filter, filter,  
getJavaScriptUtils, getNearestForm, getScriptReporter, getUserLocale,  
localRelease, registerTagError, release, renderAttribute, rewriteName,  
write

**Methods from javax.servlet.jsp.tagext.BodyTagSupport**

doInitBody, getBodyContent, getPreviousOut, release, setBodyContent

**Methods from javax.servlet.jsp.tagext.TagSupport**

doAfterBody, doEndTag, doStartTag, findAncestorWithClass, getId,  
getParent, getValue, getValues, release, removeValue, setId,  
setPageContext, setParent, setValue

**Methods from class java.lang.Object**

clone, equals, finalize, getClass, hashCode, notify, notifyAll,  
toString, wait, wait, wait

**Methods from interface javax.servlet.jsp.tagext.BodyTag**

doInitBody, setBodyContent

**Methods from interface `com.bea.wlw.netui.tags.html.Formattable`**

`addFormatter`

**Methods from interface `com.bea.wlw.netui.tags.IAttributeConsumer`**

`setAttribute`

**Methods from interface `javax.servlet.jsp.tagext.IterationTag`**

`doAfterBody`

**Methods from interface `javax.servlet.jsp.tagext.Tag`**

`doEndTag`, `doStartTag`, `getParent`, `release`, `setPageContext`, `setParent`

## Field Detail

### NULL\_VALUE

```
public static final String NULL_VALUE
```

## Constructor Detail

### Select

```
public Select()
```

## Method Detail

### **addFormatter(FormatTag.Formatter) Method**

```
public void addFormatter(FormatTag.Formatter formatter)
```

Adds a `FormatTag.Formatter` to the `Select`'s set of formatters

#### Parameters

*formatter*

– a `FormatTag.Formatter` added by a child `FormatTag`.

---

**containsString(List, String) Method**

```
protected boolean containsString(List list,
                                String value)
```

Return true, if the list contains an `Object.toString().equals(value)`.

---

**doAfterBody() Method**

```
public int doAfterBody()
    throws JspException
```

Save any body content of this tag, which will generally be the option(s) representing the values displayed to the user.

*Overrides*

```
BodyTagSupport.doAfterBody()
```

**Exceptions**

*JspException*  
if a JSP exception has occurred

---

**doEndTag() Method**

```
public int doEndTag()
    throws JspException
```

Render the end of this select.

*Overrides*

```
BodyTagSupport.doEndTag()
```

**Exceptions**

*JspException*  
if a JSP exception has occurred

---

**doStartTag() Method**

```
public int doStartTag()
    throws JspException
```

Render the beginning of this select.

*Overrides*

```
BodyTagSupport.doStartTag()
```

## Exceptions

*JspException*

if a JSP exception has occurred

---

## evaluateDefaultValue() Method

```
protected Object evaluateDefaultValue()
```

Evaluate the defaultValues

### Overrides

```
DefaultableDataSourceTag.evaluateDefaultValue()
```

---

## getAccessKey() Method

```
public String getAccessKey()
```

Gets the accessKey attribute value.

### Returns

*the accessKey value.*

---

## getDisabled() Method

```
public String getDisabled()
```

Returns the boolean value or expression indicating the disable state of the Select.

### Returns

*the disabled state (true or false) or an expression*

---

## getMultiple() Method

```
public boolean getMultiple()
```

Gets whether multiple selections are allowed.

### Returns

*the multiple value*

---

## getNullableOptionText() Method

```
public String getNullableOptionText()
```

If the nullable option is true, this is the text of that option. The default is "";

### Returns

*the text of the nullable option*

---

## getOnBlur() Method

```
public String getOnBlur()
```

Gets the onBlur javascript event.

### Returns

*the onBlur event.*

---

## getOnChange() Method

```
public String getOnChange()
```

Gets the onChange javascript event.

### Returns

*the onChange event.*

---

## getOnClick() Method

```
public String getOnClick()
```

Gets the onClick javascript event.

### Returns

*the onClick event.*

---

## getOnDbClick() Method

```
public String getOnDbClick()
```

Gets the onDbClick javascript event.

**Returns**

*the onDbClick event.*

---

**getOnFocus() Method**

```
public String getOnFocus()
```

Gets the onFocus javascript event.

**Returns**

*the onFocus event.*

---

**getOnKeyDown() Method**

```
public String getOnKeyDown()
```

Gets the onKeyDown javascript event.

**Returns**

*the onKeyDown event.*

---

**getOnKeyPress() Method**

```
public String getOnKeyPress()
```

Gets the onKeyPress javascript event.

**Returns**

*the onKeyPress event.*

---

**getOnKeyUp() Method**

```
public String getOnKeyUp()
```

Gets the onKeyUp javascript event.

**Returns**

*the onKeyUp event.*

---

## **getOnMouseDown() Method**

```
public String getOnMouseDown()
```

Gets the onMouseDown javascript event.

### **Returns**

*the onMouseDown event.*

---

## **getOnMouseMove() Method**

```
public String getOnMouseMove()
```

Gets the onMouseMove javascript event.

### **Returns**

*the onMouseMove event.*

---

## **getOnMouseOut() Method**

```
public String getOnMouseOut()
```

Gets the onMouseOut javascript event.

### **Returns**

*the onMouseOut event.*

---

## **getOnMouseOver() Method**

```
public String getOnMouseOver()
```

Gets the onMouseOver javascript event.

### **Returns**

*the onMouseOver event.*

---

## **getOnMouseUp() Method**

```
public String getOnMouseUp()
```

Gets the onMouseUp javascript event.

**Returns**

*the onMouseUp event.*

---

**getOnSelect() Method**

```
public String getOnSelect()
```

Gets the onSelect javascript event.

**Returns**

*the onSelect event.*

---

**getSize() Method**

```
public String getSize()
```

Gets how many options are displayed.

**Returns**

*the size*

---

**getStyle() Method**

```
public String getStyle()
```

Gets the style of the rendered html tag.

**Returns**

*the style.*

---

**getStyleClass() Method**

```
public String getStyleClass()
```

Gets the style class of the rendered html tag.

**Returns**

*the style class.*

---



## getTabIndex() Method

```
public String getTabIndex()
```

Gets the tabIndex of the rendered html tag.

### Returns

*the tabIndex.*

---

## getTagId() Method

```
public String getTagId()
```

Return the ID of the select.

### Returns

*the ID.*

---

## getTagName() Method

```
public String getTagName()
```

Return the name of the Tag.

### Overrides

```
AbstractBaseTag.getTagName()
```

---

## isMatched(String) Method

```
public boolean isMatched(String value)
```

Does the specified value match one of those we are looking for?

### Parameters

*value*

Value to be compared

---

## isNullable() Method

```
public boolean isNullable()
```

Gets whether a null option is desired.

**Returns**

*the nullable value*

---

**localRelease() Method**

```
protected void localRelease()
```

Release any acquired resources.

**Overrides**

```
OptionsDataSourceTag.localRelease()
```

---

**setAccessKey(String) Method**

```
public void setAccessKey(String accessKey)
```

Sets the accessKey attribute value. This should key value of the keyboard navigation key. It is recommended not to use the following values because there are often used by browsers A, C, E, F, G, H, V, left arrow, and right arrow.

**Parameters**

*accessKey*

- the accessKey value.

---

**setAttribute(String, String) Method**

```
public void setAttribute(String name,
                        String value)
    throws JspException
```

Set an attribute value. The name represents the name of the attribute. The value represents the value and may contain a netui expression. This method may result in errors being generated. This requires that the tag buffer its body and write attributes in the end tag. For the select tag it is not legal to set the id or name attributes with this method.

**Parameters**

*name*

The name of the attribute. This value may not be null or the empty string.

*value*

The value of the attribute. This may contain a netui expression.

## Exceptions

### *JspException*

A *JspException* may be thrown if there is an error setting the attribute.

---

## setEnabled(String) Method

```
public void setEnabled(String disabled)
```

Set the enable state either with the literal "true" or "false" or with an expression.

### Parameters

*disabled*

- true or false or an expression

---

## setMultiple(boolean) Method

```
public void setMultiple(boolean multiple)
```

Set whether multiple selections are allowed.

### Parameters

*multiple*

- the multiple value ("true" or "false")

---

## setNullable(boolean) Method

```
public void setNullable(boolean nullable)
```

Set whether a null option is desired.

### Parameters

*nullable*

- the nullable value

---

## setNullableOptionText(String) Method

```
public void setNullableOptionText(String nullableOptionText)
```

Set the text of the nullable option. If the nullable option is true, this is the text of that option. The default is "";

**Returns**

*the text of the nullable option*

---

**setOnBlur(String) Method**

```
public void setOnBlur(String onblur)
```

Sets the onBlur javascript event.

**Parameters**

*onblur*  
- the onBlur event.

---

**setOnChange(String) Method**

```
public void setOnChange(String onchange)
```

Sets the onChange javascript event.

**Parameters**

*onchange*  
- the onChange event.

---

**setOnClick(String) Method**

```
public void setOnClick(String onclick)
```

Sets the onClick javascript event.

**Parameters**

*onclick*  
- the onClick event.

---

**setOnDbClick(String) Method**

```
public void setOnDbClick(String ondblclick)
```

Sets the onDbClick javascript event.

**Parameters**

*ondblclick*  
- the onDblClick event.

---

**setOnFocus(String) Method**

```
public void setOnFocus(String onblur)
```

Sets the onFocus javascript event.

**Parameters**

*onblur*  
- the onFocus event.

---

**setOnKeyDown(String) Method**

```
public void setOnKeyDown(String onkeydown)
```

Sets the onKeyDown javascript event.

**Parameters**

*onkeydown*  
- the onKeyDown event.

---

**setOnKeyPress(String) Method**

```
public void setOnKeyPress(String onkeypress)
```

Sets the onKeyPress javascript event.

**Parameters**

*onkeypress*  
- the onKeyPress event.

---

**setOnKeyUp(String) Method**

```
public void setOnKeyUp(String onkeyup)
```

Sets the onKeyUp javascript event.

**Parameters**

*onkeyup*  
- the onKeyUp event.

---

**setOnMouseDown(String) Method**

```
public void setOnMouseDown(String onmousedown)
```

Sets the onMouseDown javascript event.

**Parameters**

*onmousedown*  
- the onMouseDown event.

---

**setOnMouseMove(String) Method**

```
public void setOnMouseMove(String onmousemove)
```

Sets the onMouseMove javascript event.

**Parameters**

*onmousemove*  
- the onMouseMove event.

---

**setOnMouseOut(String) Method**

```
public void setOnMouseOut(String onmouseout)
```

Sets the onMouseOut javascript event.

**Parameters**

*onmouseout*  
- the onMouseOut event.

---

**setOnMouseOver(String) Method**

```
public void setOnMouseOver(String onmouseover)
```

Sets the onMouseOver javascript event.

**Parameters**

*onmouseover*

- the onMouseOver event.

---

**setOnMouseUp(String) Method**

```
public void setOnMouseUp(String onmouseup)
```

Sets the onMouseUp javascript event.

**Parameters**

*onmouseup*

- the onMouseUp event.

---

**setOnSelect(String) Method**

```
public void setOnSelect(String onselect)
```

Sets the onSelect javascript event.

**Parameters**

*onselect*

- the onSelect event.

---

**setSize(String) Method**

```
public void setSize(String size)
```

Sets how many options are displayed.

**Parameters**

*size*

- the size (a number)

---

**setStyle(String) Method**

```
public void setStyle(String style)
```

Sets the style of the rendered html tag.

**Parameters**

*style*  
- the html style.

---

**setStyleClass(String) Method**

```
public void setStyleClass(String styleClass)
```

Sets the style class of the rendered html tag.

**Parameters**

*styleClass*  
- the html style class.

---

**setTabIndex(String) Method**

```
public void setTabIndex(String tabindex)
```

Sets the tabIndex of the rendered html tag.

**Parameters**

*tabindex*  
- the tab index.

---

**setTagId(String) Method**

```
public void setTagId(String tagId)
```

Set the ID of the select.

**Parameters**

*tagId*  
- the ID.



# SelectOption Class

*public class SelectOption*

extends AbstractBaseTag  
implements IAttributeConsumer

An option whose state is determined by its enclosing SelectOption.

Attribute Descriptions			
Attribute	Required	Runtime Expression Evaluation	Data Bindable
<b>disabled</b>	No	No	No
	The option's disabled state ("true" or "false")		
<b>id</b>	No	No	No
	<b>The id attribute has been deprecated; instead, use tagId.</b> The id attribute of the generated selectOption. You should use the tagId which will encode the id correctly for containers such as the Portal which needs to rewrite IDs to make them unique.		
<b>tagId</b>	No	No	No
	The id of this anchor. The real id written to the HTML stream may be changed by the container (Portal), but may be looked up with the 'tagId' value through the JavaScript method <code>getNetuiTagName( tagId, tag )</code> method written by the <code>&lt;netui:html&gt;</code> tag.		
<b>value</b>	Yes	No	Read Only
	An expression to a bean property, or a literal, which is the value of this option.		

## Related Topics

`<netui:select>` Tag Sample  
Select

---

## Hierarchy

```

Object
  TagSupport
    BodyTagSupport
      AbstractBaseTag
        SelectOption
  
```

*All Implemented Interfaces*BodyTag, IAttributeConsumer, IterationTag, Serializable, Tag

---

## Field Summary

```

    public
    static VALUE
final String      String

```

**Fields from com.bea.wlw.netui.tags.AbstractBaseTag**

```

ATTR_GENERAL, ATTR_GENERAL_EXPRESSION, ATTR_JAVASCRIPT, ATTR_STYLE,
CLASS, ID, JAVASCRIPT_STATUS, NAME, NETUI_UNIQUE_CNT, ONCLICK,
ONDBLCLICK, ONKEYDOWN, ONKEYPRESS, ONKEYUP, ONMOUSEDOWN, ONMOUSEMOVE,
ONMOUSEOUT, ONMOUSEOVER, ONMOUSEUP, STYLE, TABINDEX

```

**Fields from javax.servlet.jsp.tagext.BodyTagSupport**

```

bodyContent

```

**Fields from javax.servlet.jsp.tagext.TagSupport**

```

id, pageContext

```

**Fields from interface javax.servlet.jsp.tagext.BodyTag**

```

EVAL_BODY_BUFFERED, EVAL_BODY_TAG

```

**Fields from interface javax.servlet.jsp.tagext.IterationTag**

```

EVAL_BODY_AGAIN

```

**Fields from interface javax.servlet.jsp.tagext.Tag**

```

EVAL_BODY_INCLUDE, EVAL_PAGE, SKIP_BODY, SKIP_PAGE

```

## Constructor Summary

```

SelectOption()

```

## Method Summary

```

    public
    int doAfterBody()

```

Process the body text of this tag (if any).

```
public
    int doEndTag()
```

Process the end of this tag.

```
public
    int doStartTag()
```

Process the start of this tag.

```
public
    String getDisabled()
```

Gets if this option is disabled or not.

```
public
    String getId()
```

Gets the option id.

```
public
    String getOnClick()
```

Gets the onClick javascript event.

```
public
    String getOnDbClick()
```

Gets the onDbClick javascript event.

```
public
    String getOnKeyDown()
```

Gets the onKeyDown javascript event.

```
public
    String getOnKeyPress()
```

Gets the onKeyPress javascript event.

```
public
    String getOnKeyUp()
```

Gets the onKeyUp javascript event.

```
public
    String getOnMouseDown()
```

Gets the onMouseDown javascript event.

```
public
    String getOnMouseMove()
```

Gets the onMouseMove javascript event.

```
public
    String getOnMouseOut()
```

Gets the onMouseOut javascript event.

```
public
    String getOnMouseOver()
```

Gets the onMouseOver javascript event.

```
public
    String getOnMouseUp()
```

Gets the onMouseUp javascript event.

```
public
    String getStyle()
```

Gets the style of the rendered html tag.

```

public getStyleClass()
String      Gets the style class of the rendered html tag.

public
String getTabIndex()
           Gets the tabIndex of the rendered html tag.

public
String getTagId()
           Return the ID of the selectOption.

public
String getTagName()
           Return the name of the Tag.

public
String getValue()
           Return the value of this SelectOption.

protected
void localRelease()
           Release any acquired resources.

public
void setAttribute(String name, String value)
           Set an attribute value.

public
void setDisabled(String disabled)
           Set if this option is disabled or not.

public
void setId(String id)
           Sets the option id.

public
void setOnClick(String onclick)
           Sets the onClick javascript event.

public
void setOnDbClick(String ondblclick)
           Sets the onDbClick javascript event.

public
void setOnKeyDown(String onkeydown)
           Sets the onKeyDown javascript event.

public
void setOnKeyPress(String onkeypress)
           Sets the onKeyPress javascript event.

public
void setOnKeyUp(String onkeyup)
           Sets the onKeyUp javascript event.

public
void setOnMouseDown(String onmousedown)
           Sets the onMouseDown javascript event.

public
void setOnMouseMove(String onmousemove)
           Sets the onMouseMove javascript event.

```

```

public setOnMouseOut(String onmouseout)
    void          Sets the onMouseOut javascript event.
public
    void setOnMouseOver(String onmouseover)
        Sets the onMouseOver javascript event.
public
    void setOnMouseUp(String onmouseup)
        Sets the onMouseUp javascript event.
public
    void setStyle(String style)
        Sets the style of the rendered html tag.
public
    void setStyleClass(String styleClass)
        Sets the style class of the rendered html tag.
public
    void setTabIndex(String tabindex)
        Sets the tabIndex of the rendered html tag.
public
    void setTagId(String tagId)
        Set the ID of the anchor.
public
    void setValue(String value)
        Set the value of this SelectOption.

```

#### Methods from **com.bea.wlw.netui.tags.AbstractBaseTag**

```

addTagIdMapping, evaluateAttributeToString, filter, filter,
getJavaScriptUtils, getNearestForm, getScriptReporter, getUserLocale,
registerTagError, release, renderAttribute, rewriteName, write

```

#### Methods from **javax.servlet.jsp.tagext.BodyTagSupport**

```

doInitBody, getBodyContent, getPreviousOut, release, setBodyContent

```

#### Methods from **javax.servlet.jsp.tagext.TagSupport**

```

doAfterBody, doEndTag, doStartTag, findAncestorWithClass, getParent,
getValue, getValues, release, removeValue, setPageContext, setParent,
setValue

```

#### Methods from class **java.lang.Object**

```

clone, equals, finalize, getClass, hashCode, notify, notifyAll,
toString, wait, wait, wait

```

#### Methods from interface **javax.servlet.jsp.tagext.BodyTag**

```

doInitBody, setBodyContent

```

Methods from interface `com.bea.wlw.netui.tags.IAttributeConsumer`

`setAttribute`

Methods from interface `javax.servlet.jsp.tagext.IterationTag`

`doAfterBody`

Methods from interface `javax.servlet.jsp.tagext.Tag`

`doEndTag`, `doStartTag`, `getParent`, `release`, `setPageContext`, `setParent`

## Field Detail

### VALUE

```
public static final String VALUE
```

## Constructor Detail

### SelectOption

```
public SelectOption()
```

## Method Detail

### doAfterBody() Method

```
public int doAfterBody()  
    throws JspException
```

Process the body text of this tag (if any).

#### *Overrides*

```
BodyTagSupport.doAfterBody()
```

### Exceptions

*JspException*  
if a JSP exception has occurred

---

## doEndTag() Method

```
public int doEndTag()
    throws JspException
```

Process the end of this tag.

### Overrides

```
BodyTagSupport.doEndTag()
```

### Exceptions

*JspException*  
if a JSP exception has occurred

---

## doStartTag() Method

```
public int doStartTag()
    throws JspException
```

Process the start of this tag.

### Overrides

```
BodyTagSupport.doStartTag()
```

### Exceptions

*JspException*  
if a JSP exception has occurred

---

## getDisabled() Method

```
public String getDisabled()
```

Gets if this option is disabled or not.

### Returns

*the disabled state ("true" or "false")*

---

## getId() Method

```
public String getId()
```

Gets the option id.

### Overrides

```
TagSupport.getId()
```

**Returns**

*the option id.*

---

**getOnClick() Method**

```
public String getOnClick()
```

Gets the onClick javascript event.

**Returns**

*the onClick event.*

---

**getOnDbClick() Method**

```
public String getOnDbClick()
```

Gets the onDbClick javascript event.

**Returns**

*the onDbClick event.*

---

**getOnKeyDown() Method**

```
public String getOnKeyDown()
```

Gets the onKeyDown javascript event.

**Returns**

*the onKeyDown event.*

---

**getOnKeyPress() Method**

```
public String getOnKeyPress()
```

Gets the onKeyPress javascript event.

**Returns**

*the onKeyPress event.*

---



## getOnKeyUp() Method

```
public String getOnKeyUp()
```

Gets the onKeyUp javascript event.

### Returns

*the onKeyUp event.*

---

## getOnMouseDown() Method

```
public String getOnMouseDown()
```

Gets the onMouseDown javascript event.

### Returns

*the onMouseDown event.*

---

## getOnMouseMove() Method

```
public String getOnMouseMove()
```

Gets the onMouseMove javascript event.

### Returns

*the onMouseMove event.*

---

## getOnMouseOut() Method

```
public String getOnMouseOut()
```

Gets the onMouseOut javascript event.

### Returns

*the onMouseOut event.*

---

## getOnMouseOver() Method

```
public String getOnMouseOver()
```

Gets the onMouseOver javascript event.

## Returns

*the onMouseOver event.*

---

## getOnMouseUp() Method

```
public String getOnMouseUp()
```

Gets the onMouseUp javascript event.

## Returns

*the onMouseUp event.*

---

## getStyle() Method

```
public String getStyle()
```

Gets the style of the rendered html tag.

## Returns

*the style.*

---

## getStyleClass() Method

```
public String getStyleClass()
```

Gets the style class of the rendered html tag.

## Returns

*the style class.*

---

## getTabIndex() Method

```
public String getTabIndex()
```

Gets the tabIndex of the rendered html tag.

## Returns

*the tabIndex.*

---

## getTagId() Method

```
public String getTagId()
```

Return the ID of the selectOption. The id may be rewritten by the container (such as a portal) to make sure it is unique. JavaScript may lookup the actual id of the element by looking it up in the `netui_names` table written into the HTML.

### Returns

*the tagId.*

---

## getTagName() Method

```
public String getTagName()
```

Return the name of the Tag.

### Overrides

```
AbstractBaseTag.getTagName()
```

---

## getValue() Method

```
public String getValue()
```

Return the value of this SelectOption.

### Returns

*the SelectOption value*

---

## localRelease() Method

```
protected void localRelease()
```

Release any acquired resources.

### Overrides

```
AbstractBaseTag.localRelease()
```

---

## setAttribute(String, String) Method

```
public void setAttribute(String name,
                        String value)
    throws JspException
```

Set an attribute value. The `name` represents the name of the attribute. The `value` represents the value and may contain a netui expression. This method may result in errors being generated. This requires that the tag buffer its body and write attributes in the end tag. For the `selectoption` tag it is not legal to set the `id`, `name`, or `value` attributes with this method.

### Parameters

*name*

The name of the attribute. This value may not be null or the empty string.

*value*

The value of the attribute. This may contain a netui expression.

### Exceptions

*JspException*

A `JspException` may be thrown if there is an error setting the attribute.

---

## setEnabled(String) Method

```
public void setEnabled(String disabled)
```

Set if this option is disabled or not.

### Parameters

*disabled*

– "true" or "false"

---

## setId(String) Method

```
public void setId(String id)
```

Sets the option id.

### Overrides

```
TagSupport.setId(String)
```

### Parameters

*id*

– the option id

---

## setOnClick(String) Method

```
public void setOnClick(String onclick)
```

Sets the onClick javascript event.

### Parameters

*onclick*

– the onClick event.

---

## setOnDbClick(String) Method

```
public void setOnDbClick(String ondblclick)
```

Sets the onDbClick javascript event.

### Parameters

*ondblclick*

– the onDbClick event.

---

## setOnKeyDown(String) Method

```
public void setOnKeyDown(String onkeydown)
```

Sets the onKeyDown javascript event.

### Parameters

*onkeydown*

– the onKeyDown event.

---

## setOnKeyPress(String) Method

```
public void setOnKeyPress(String onkeypress)
```

Sets the onKeyPress javascript event.

### Parameters

*onkeypress*

– the onKeyPress event.

---

## setOnKeyUp(String) Method

```
public void setOnKeyUp(String onkeyup)
```

Sets the onKeyUp javascript event.

**Parameters***onkeyup*

– the onKeyUp event.

**setOnMouseDown(String) Method**

```
public void setOnMouseDown(String onmousedown)
```

Sets the onMouseDown javascript event.

**Parameters***onmousedown*

– the onMouseDown event.

**setOnMouseMove(String) Method**

```
public void setOnMouseMove(String onmousemove)
```

Sets the onMouseMove javascript event.

**Parameters***onmousemove*

– the onMouseMove event.

**setOnMouseOut(String) Method**

```
public void setOnMouseOut(String onmouseout)
```

Sets the onMouseOut javascript event.

**Parameters***onmouseout*

– the onMouseOut event.

**setOnMouseOver(String) Method**

```
public void setOnMouseOver(String onmouseover)
```

Sets the onMouseOver javascript event.

## Parameters

*onmouseover*

– the onMouseOver event.

---

## setOnMouseUp(String) Method

```
public void setOnMouseUp(String onmouseup)
```

Sets the onMouseUp javascript event.

## Parameters

*onmouseup*

– the onMouseUp event.

---

## setStyle(String) Method

```
public void setStyle(String style)
```

Sets the style of the rendered html tag.

## Parameters

*style*

– the html style.

---

## setStyleClass(String) Method

```
public void setStyleClass(String styleClass)
```

Sets the style class of the rendered html tag.

## Parameters

*styleClass*

– the html style class.

---

## setTabIndex(String) Method

```
public void setTabIndex(String tabindex)
```

Sets the tabIndex of the rendered html tag.

## Parameters

*tabindex*

– the tab index.

---

## setTagId(String) Method

```
public void setTagId(String tagId)
```

Set the ID of the anchor.

## Parameters

*tagId*

– the tagId.

---

## setValue(String) Method

```
public void setValue(String value)
```

Set the value of this SelectOption.

## Parameters

*value*

– the SelectOption value

---



# TextArea Class

*public class* **TextArea**

extends DefaultableDataSourceTag  
implements Formattable, IAttributeConsumer

Renders a databound TextArea with the given attributes.

Attribute Descriptions			
Attribute	Required	Runtime Expression Evaluation	Data Bindable
accessKey	No	No	No
	Define the keyboard navigation key for the element.		
cols	No	No	Read Only
	The number of columns in the TextArea		
dataSource	Yes	No	Read / Write
	The name of a form bean property or an expression that contains the TextArea state.		
defaultValue	No	No	Read Only
	A literal value or an expression that is used to set the TextArea's initial state. The default value is applied only if the dataSource returns null.		
disabled	No	No	Read Only
	A boolean specifying whether the TextArea is disabled.		
readOnly	No	No	No
	A boolean specifying whether the TextArea is read-only.		
rows	No	No	Read Only
	The number of rows in the TextArea.		
tagId	No	No	No
	The ID of this textarea used by the enclosing html tag to get the tag's name for JavaScript and focus purposes.		

## Related Topics

<netui:textArea> Sample

***Hierarchy***

```

Object
  TagSupport
    BodyTagSupport
      AbstractBaseTag
        DataSourceTag
          DefaultableDataSourceTag
            TextArea

```

***All Implemented Interfaces***

BodyTag, Formattable, IAttributeConsumer, IterationTag, Serializable, Tag

## Field

## Summary

```

protected String
    cols
        The number of columns in the
        TextArea.

public static
final String COLS
    String

protected String
    disabled
        Whether or not the TextArea is disabled

public static
final String DISABLED
    String

protected ArrayList
    formatters
        ArrayList

protected boolean
    readonly
        The read-only status of the TextArea

public static
final String READONLY
    String

protected String
    rows
        The number of row in the TextArea

public static
final String ROWS
    String

```

Fields from `com.bea.wlw.netui.tags.AbstractBaseTag`

ATTR\_GENERAL, ATTR\_GENERAL\_EXPRESSION, ATTR\_JAVASCRIPT, ATTR\_STYLE, CLASS, ID, JAVASCRIPT\_STATUS, NAME, NETUI\_UNIQUE\_CNT, ONCLICK, ONDBLCLICK, ONKEYDOWN, ONKEYPRESS, ONKEYUP, ONMOUSEDOWN, ONMOUSEMOVE, ONMOUSEOUT, ONMOUSEOVER, ONMOUSEUP, STYLE, TABINDEX

### Fields from `javax.servlet.jsp.tagext.BodyTagSupport`

bodyContent

### Fields from `javax.servlet.jsp.tagext.TagSupport`

id, pageContext

### Fields from interface `javax.servlet.jsp.tagext.BodyTag`

EVAL\_BODY\_BUFFERED, EVAL\_BODY\_TAG

### Fields from interface `javax.servlet.jsp.tagext.IterationTag`

EVAL\_BODY\_AGAIN

### Fields from interface `javax.servlet.jsp.tagext.Tag`

EVAL\_BODY\_INCLUDE, EVAL\_PAGE, SKIP\_BODY, SKIP\_PAGE

## Constructor Summary

*TextArea()*

## Method Summary

```
public
    void addFormatter(FormatTag.Formatter formatter)
        Adds a FormatTag.Formatter to the TextArea's set of
        formatters

public
    int doEndTag()
        Render the TextArea.

public
    int doStartTag()
        Prepare the TextArea's formatters.

public
    String getAccessKey()
        Gets the accessKey attribute value.
```

```
public getCols()
String      Gets the number of columns in the TextArea.

public
String getDisabled()
           Gets if this TextArea is disabled or not.

public
String getOnBlur()
           Gets the onBlur javascript event.

public
String getOnChange()
           Gets the onChange javascript event.

public
String getOnClick()
           Gets the onClick javascript event.

public
String getOnDbClick()
           Gets the onDbClick javascript event.

public
String getOnFocus()
           Gets the onFocus javascript event.

public
String getOnKeyDown()
           Gets the onKeyDown javascript event.

public
String getOnKeyPress()
           Gets the onKeyPress javascript event.

public
String getOnKeyUp()
           Gets the onKeyUp javascript event.

public
String getOnMouseDown()
           Gets the onMouseDown javascript event.

public
String getOnMouseMove()
           Gets the onMouseMove javascript event.

public
String getOnMouseOut()
           Gets the onMouseOut javascript event.

public
String getOnMouseOver()
           Gets the onMouseOver javascript event.

public
String getOnMouseUp()
           Gets the onMouseUp javascript event.

public
String getOnSelect()
           Gets the onSelect javascript event.
```

```

    public getReadonly()
boolean        Gets if this TextArea is read-only.

    public
String getRows()
                Gets the number of rows in the TextArea.

    public
String getStyle()
                Gets the style of the rendered html tag.

    public
String getStyleClass()
                Gets the style class of the rendered html tag.

    public
String getTabIndex()
                Gets the tabIndex of the rendered html tag.

    public
String getTagId()
                Return the ID of the textarea.

    public
String getTagName()
                Return the name of the Tag.

protected
    void localRelease()
                Release any acquired resources.

    public
    void setAccessKey(String accessKey)
                Sets the accessKey attribute value.

    public
    void setAttribute(String name, String value)
                Set an attribute value.

    public
    void setCols(String cols)
                Sets the number of columns in the TextArea.

    public
    void setDisabled(String disabled)
                Set if this TextArea is disabled or not.

    public
    void setOnBlur(String onblur)
                Sets the onBlur javascript event.

    public
    void setOnChange(String onchange)
                Sets the onChange javascript event.

    public
    void setOnClick(String onclick)
                Sets the onClick javascript event.

    public
    void setOnDbClick(String ondblclick)
                Sets the onDbClick javascript event.

```

```

public setOnFocus(String onBlur)
    void          Sets the onFocus javascript event.
public
    void setOnKeyDown(String onkeydown)
        Sets the onKeyDown javascript event.
public
    void setOnKeyPress(String onkeypress)
        Sets the onKeyPress javascript event.
public
    void setOnKeyUp(String onkeyup)
        Sets the onKeyUp javascript event.
public
    void setOnMouseDown(String onmousedown)
        Sets the onMouseDown javascript event.
public
    void setOnMouseMove(String onmousemove)
        Sets the onMouseMove javascript event.
public
    void setOnMouseOut(String onmouseout)
        Sets the onMouseOut javascript event.
public
    void setOnMouseOver(String onmouseover)
        Sets the onMouseOver javascript event.
public
    void setOnMouseUp(String onmouseup)
        Sets the onMouseUp javascript event.
public
    void setOnSelect(String onselect)
        Sets the onSelect javascript event.
public
    void setReadonly(boolean readonly)
        Set if this TextArea is read-only.
public
    void setRows(String rows)
        Sets the number of rows in the TextArea.
public
    void setStyle(String style)
        Sets the style of the rendered html tag.
public
    void setStyleClass(String styleClass)
        Sets the style class of the rendered html tag.
public
    void setTabIndex(String tabindex)
        Sets the tabIndex of the rendered html tag.
public
    void setTagId(String tagId)
        Set the ID of the textarea.

```

**Methods from `com.bea.wlw.netui.tags.DefaultableDataSourceTag`**

`getDefaultValue, setDefaultValue`

**Methods from `com.bea.wlw.netui.tags.DataSourceTag`**

`getDataSource, localRelease, setDataSource`

**Methods from `com.bea.wlw.netui.tags.AbstractBaseTag`**

`addTagIdMapping, evaluateAttributeToString, filter, filter,  
getJavaScriptUtils, getNearestForm, getScriptReporter, getUserLocale,  
localRelease, registerTagError, release, renderAttribute, rewriteName,  
write`

**Methods from `javax.servlet.jsp.tagext.BodyTagSupport`**

`doAfterBody, doInitBody, getBodyContent, getPreviousOut, release,  
setBodyContent`

**Methods from `javax.servlet.jsp.tagext.TagSupport`**

`doAfterBody, doEndTag, doStartTag, findAncestorWithClass, getId,  
getParent, getValue, getValues, release, removeValue, setId,  
setPageContext, setParent, setValue`

**Methods from class `java.lang.Object`**

`clone, equals, finalize, getClass, hashCode, notify, notifyAll,  
toString, wait, wait, wait`

**Methods from interface `javax.servlet.jsp.tagext.BodyTag`**

`doInitBody, setBodyContent`

**Methods from interface `com.bea.wlw.netui.tags.html.Formatable`**

`addFormatter`

**Methods from interface `com.bea.wlw.netui.tags.IAttributeConsumer`**

`setAttribute`

**Methods from interface `javax.servlet.jsp.tagext.IterationTag`**

`doAfterBody`

**Methods from interface `javax.servlet.jsp.tagext.Tag`**

`doEndTag, doStartTag, getParent, release, setPageContext, setParent`

## Field Detail

### cols

`protected String cols`

The number of columns in the TextArea.

---

### COLS

`public static final String COLS`

---

### disabled

`protected String disabled`

Whether or not the TextArea is disabled

---

### DISABLED

`public static final String DISABLED`

---

### formatters

`protected ArrayList formatters`

---

### readonly

`protected boolean readonly`

The read-only status of the TextArea

---

### READONLY

`public static final String READONLY`

---

### rows

`protected String rows`

The number of row in the TextArea

---



## ROWS

```
public static final String ROWS
```

## Constructor Detail

### TextArea

```
public TextArea()
```

## Method Detail

### addFormatter(FormatTag.Formatter) Method

```
public void addFormatter(FormatTag.Formatter formatter)
```

Adds a FormatTag.Formatter to the TextArea's set of formatters

#### Parameters

*formatter*

– a FormatTag.Formatter added by a child FormatTag.

---

### doEndTag() Method

```
public int doEndTag()  
    throws JspException
```

Render the TextArea.

#### Overrides

BodyTagSupport.doEndTag( )

#### Exceptions

*JspException*

if a JSP exception has occurred

---

### doStartTag() Method

```
public int doStartTag()  
    throws JspException
```

Prepare the TextArea's formatters.

**Overrides**

`BodyTagSupport.doStartTag()`

**Exceptions**

*JspException*

if a JSP exception has occurred

---

**getAccessKey() Method**

`public String getAccessKey()`

Gets the accessKey attribute value.

**Returns**

*the accessKey value.*

---

**getCols() Method**

`public String getCols()`

Gets the number of columns in the TextArea.

**Returns**

*the number of columns*

---

**getDisabled() Method**

`public String getDisabled()`

Gets if this TextArea is disabled or not.

**Returns**

*the disabled state (true or false)*

---

**getOnBlur() Method**

`public String getOnBlur()`

Gets the onBlur javascript event.

## Returns

*the onBlur event.*

---

## getOnChange() Method

```
public String getOnChange()
```

Gets the onChange javascript event.

## Returns

*the onChange event.*

---

## getOnClick() Method

```
public String getOnClick()
```

Gets the onClick javascript event.

## Returns

*the onClick event.*

---

## getOnDbClick() Method

```
public String getOnDbClick()
```

Gets the onDbClick javascript event.

## Returns

*the onDbClick event.*

---

## getOnFocus() Method

```
public String getOnFocus()
```

Gets the onFocus javascript event.

## Returns

*the onFocus event.*

---

## getOnKeyDown() Method

```
public String getOnKeyDown()
```

Gets the onKeyDown javascript event.

### Returns

*the onKeyDown event.*

---

## getOnKeyPress() Method

```
public String getOnKeyPress()
```

Gets the onKeyPress javascript event.

### Returns

*the onKeyPress event.*

---

## getOnKeyUp() Method

```
public String getOnKeyUp()
```

Gets the onKeyUp javascript event.

### Returns

*the onKeyUp event.*

---

## getOnMouseDown() Method

```
public String getOnMouseDown()
```

Gets the onMouseDown javascript event.

### Returns

*the onMouseDown event.*

---

## getOnMouseMove() Method

```
public String getOnMouseMove()
```

Gets the onMouseMove javascript event.

## Returns

*the onMouseMove event.*

---

## getOnMouseOut() Method

```
public String getOnMouseOut()
```

Gets the onMouseOut javascript event.

## Returns

*the onMouseOut event.*

---

## getOnMouseOver() Method

```
public String getOnMouseOver()
```

Gets the onMouseOver javascript event.

## Returns

*the onMouseOver event.*

---

## getOnMouseUp() Method

```
public String getOnMouseUp()
```

Gets the onMouseUp javascript event.

## Returns

*the onMouseUp event.*

---

## getOnSelect() Method

```
public String getOnSelect()
```

Gets the onSelect javascript event.

## Returns

*the onSelect event.*

---

## getReadOnly() Method

```
public boolean getReadOnly()
```

Gets if this TextArea is read-only.

### Returns

*the read-only state*

---

## getRows() Method

```
public String getRows()
```

Gets the number of rows in the TextArea.

### Returns

*the number of rows*

---

## getStyle() Method

```
public String getStyle()
```

Gets the style of the rendered html tag.

### Returns

*the style.*

---

## getStyleClass() Method

```
public String getStyleClass()
```

Gets the style class of the rendered html tag.

### Returns

*the style class.*

---

## getTabIndex() Method

```
public String getTabIndex()
```

Gets the tabIndex of the rendered html tag.

**Returns**

*the tabindex.*

---

**getTagId() Method**

```
public String getTagId()
```

Return the ID of the textarea.

**Returns**

*the ID.*

---

**getTagName() Method**

```
public String getTagName()
```

Return the name of the Tag.

**Overrides**

```
AbstractBaseTag.getTagName()
```

---

**localRelease() Method**

```
protected void localRelease()
```

Release any acquired resources.

**Overrides**

```
DefaultableDataSourceTag.localRelease()
```

---

**setAccessKey(String) Method**

```
public void setAccessKey(String accessKey)
```

Sets the accessKey attribute value. This should key value of the keyboard navigation key. It is recommended not to use the following values because there are often used by browsers A, C, E, F, G, H, V, left arrow, and right arrow.

**Parameters**

*accessKey*  
– the accessKey value.

---

## setAttribute(String, String) Method

```
public void setAttribute(String name,
                        String value)
    throws JspException
```

Set an attribute value. The `name` represents the name of the attribute. The `value` represents the value and may contain a netui expression. This method may result in errors being generated. This requires that the tag buffer its body and write attributes in the end tag. For the textarea tag it is not legal to set the `id` or `name` attributes with this method.

### Parameters

*name*

The name of the attribute. This value may not be null or the empty string.

*value*

The value of the attribute. This may contain a netui expression.

### Exceptions

*JspException*

A `JspException` may be thrown if there is an error setting the attribute.

---

## setCols(String) Method

```
public void setCols(String cols)
```

Sets the number of columns in the TextArea.

### Parameters

*cols*

– the number of columns

---

## setDisabled(String) Method

```
public void setDisabled(String disabled)
```

Set if this TextArea is disabled or not.

### Parameters

*disabled*

– true or false

---



## setOnBlur(String) Method

```
public void setOnBlur(String onblur)
```

Sets the onBlur javascript event.

### Parameters

*onblur*

– the onBlur event.

---

## setOnChange(String) Method

```
public void setOnChange(String onchange)
```

Sets the onChange javascript event.

### Parameters

*onchange*

– the onChange event.

---

## setOnClick(String) Method

```
public void setOnClick(String onclick)
```

Sets the onClick javascript event.

### Parameters

*onclick*

– the onClick event.

---

## setOnDbClick(String) Method

```
public void setOnDbClick(String ondblclick)
```

Sets the onDbClick javascript event.

### Parameters

*ondblclick*

– the onDbClick event.

---

## setOnFocus(String) Method

```
public void setOnFocus(String onblur)
```

Sets the onFocus javascript event.

### Parameters

*onblur*

– the onFocus event.

---

## setOnKeyDown(String) Method

```
public void setOnKeyDown(String onkeydown)
```

Sets the onKeyDown javascript event.

### Parameters

*onkeydown*

– the onKeyDown event.

---

## setOnKeyPress(String) Method

```
public void setOnKeyPress(String onkeypress)
```

Sets the onKeyPress javascript event.

### Parameters

*onkeypress*

– the onKeyPress event.

---

## setOnKeyUp(String) Method

```
public void setOnKeyUp(String onkeyup)
```

Sets the onKeyUp javascript event.

### Parameters

*onkeyup*

– the onKeyUp event.

---

## setOnMouseDown(String) Method

```
public void setOnMouseDown(String onmousedown)
```

Sets the onMouseDown javascript event.

### Parameters

*onmousedown*

– the onMouseDown event.

---

## setOnMouseMove(String) Method

```
public void setOnMouseMove(String onmousemove)
```

Sets the onMouseMove javascript event.

### Parameters

*onmousemove*

– the onMouseMove event.

---

## setOnMouseOut(String) Method

```
public void setOnMouseOut(String onmouseout)
```

Sets the onMouseOut javascript event.

### Parameters

*onmouseout*

– the onMouseOut event.

---

## setOnMouseOver(String) Method

```
public void setOnMouseOver(String onmouseover)
```

Sets the onMouseOver javascript event.

### Parameters

*onmouseover*

– the onMouseOver event.

---

## setOnMouseUp(String) Method

```
public void setOnMouseUp(String onmouseup)
```

Sets the onMouseUp javascript event.

### Parameters

*onmouseup*

– the onMouseUp event.

---

## setOnSelect(String) Method

```
public void setOnSelect(String onselect)
```

Sets the onSelect javascript event.

### Parameters

*onselect*

– the onSelect event.

---

## setReadOnly(boolean) Method

```
public void setReadOnly(boolean readonly)
```

Set if this TextArea is read-only.

### Parameters

*readonly*

– the read-only state

---

## setRows(String) Method

```
public void setRows(String rows)
```

Sets the number of rows in the TextArea.

### Parameters

*rows*

– the number of rows

---

## setStyle(String) Method

```
public void setStyle(String style)
```

Sets the style of the rendered html tag.

### Parameters

*style*

– the html style.

---

## setStyleClass(String) Method

```
public void setStyleClass(String styleClass)
```

Sets the style class of the rendered html tag.

### Parameters

*styleClass*

– the html style class.

---

## setTabIndex(String) Method

```
public void setTabIndex(String tabindex)
```

Sets the tabIndex of the rendered html tag.

### Parameters

*tabindex*

– the tab index.

---

## setTagId(String) Method

```
public void setTagId(String tagId)
```

Set the ID of the textarea.

### Parameters

*tagId*

– the ID.

# TextBox Class

*public class* **TextBox**

extends DefaultableDataSourceTag  
implements Formattable, IAttributeConsumer

Renders a databound TextBox with the given attributes.

Attribute Descriptions			
Attribute	Required	Runtime Expression Evaluation	Data Bindable
<b>accessKey</b>	No	No	No
	Define the keyboard navigation key for the element.		
<b>alt</b>	No	No	Read Only
	Define the alt attribute of the <Input> element.		
<b>dataSource</b>	Yes	No	Read / Write
	The name of a form bean property or an expression that contains the TextBox state.		
<b>defaultValue</b>	No	No	Read Only
	A literal value or an expression that is used to set the TextBox's initial state. The default value is applied only if the dataSource returns null.		
<b>disabled</b>	No	No	Read Only
	A boolean value which if true will cause the textBox to be disabled.		
<b>maxLength</b>	No	No	Read Only
	The maximum number of characters in the TextBox.		
<b>password</b>	No	No	No
	A boolean specifying whether the password characters that the user enters into the textBox will be disguised in the browser.		
<b>readOnly</b>	No	No	No
	A boolean specifying whether the TextBox is read-only.		
<b>size</b>	No	No	Read Only
	The visible number of characters in the TextBox.		

<b>tagId</b>	No	No	No
	The ID of this textbox used by the enclosing html tag to get the tag's name, for JavaScript and focus purposes.		

**Related Topics**

&lt;netui:textBox&gt; Sample

***Hierarchy***

```

Object
  TagSupport
    BodyTagSupport
      AbstractBaseTag
        DataSourceTag
          DefaultableDataSourceTag
            TextBox

```

***All Implemented Interfaces***

BodyTag, Formattable, IAttributeConsumer, IterationTag, Serializable, Tag

**Field Summary**

```

public
static DISABLED
final String      String

public
static MAXLENGTH
final String      String

public
static READONLY
final String      String

public
static SIZE
final String      String

public
static TYPE
final String      String

public
static VALUE
final String      String

```

**Fields from com.bea.wlw.netui.tags.AbstractBaseTag**

ATTR\_GENERAL, ATTR\_GENERAL\_EXPRESSION, ATTR\_JAVASCRIPT, ATTR\_STYLE, CLASS, ID, JAVASCRIPT\_STATUS, NAME, NETUI\_UNIQUE\_CNT, ONCLICK,

ONDBLCLICK, ONKEYDOWN, ONKEYPRESS, ONKEYUP, ONMOUSEDOWN, ONMOUSEMOVE, ONMOUSEOUT, ONMOUSEOVER, ONMOUSEUP, STYLE, TABINDEX

### Fields from `javax.servlet.jsp.tagext.BodyTagSupport`

`bodyContent`

### Fields from `javax.servlet.jsp.tagext.TagSupport`

`id`, `pageContext`

### Fields from interface `javax.servlet.jsp.tagext.BodyTag`

`EVAL_BODY_BUFFERED`, `EVAL_BODY_TAG`

### Fields from interface `javax.servlet.jsp.tagext.IterationTag`

`EVAL_BODY_AGAIN`

### Fields from interface `javax.servlet.jsp.tagext.Tag`

`EVAL_BODY_INCLUDE`, `EVAL_PAGE`, `SKIP_BODY`, `SKIP_PAGE`

## Constructor Summary

***TextBox()***

## Method Summary

```
public
    void addFormatter(FormatTag.Formatter formatter)
        Adds a FormatTag.Formatter to the TextBox's set of
        formatters

public
    int doEndTag()
        Render the TextBox.

public
    int doStartTag()
        Prepare the TextBox's formatters.

public
    String getAccessKey()
        Gets the accessKey attribute value.

public
    String getAlt()
        Gets the alt attribute on the generate <input tag.
```



```
public getDisabled()
String      Gets if this TextBox is disabled or not.

public
String getMaxLength()
           Gets the maximum length (in characters) of the TextBox.

public
String getOnBlur()
           Gets the onBlur javascript event.

public
String getOnChange()
           Gets the onChange javascript event.

public
String getOnClick()
           Gets the onClick javascript event.

public
String getOnDbClick()
           Gets the onDbClick javascript event.

public
String getOnFocus()
           Gets the onFocus javascript event.

public
String getOnKeyDown()
           Gets the onKeyDown javascript event.

public
String getOnKeyPress()
           Gets the onKeyPress javascript event.

public
String getOnKeyUp()
           Gets the onKeyUp javascript event.

public
String getOnMouseDown()
           Gets the onMouseDown javascript event.

public
String getOnMouseMove()
           Gets the onMouseMove javascript event.

public
String getOnMouseOut()
           Gets the onMouseOut javascript event.

public
String getOnMouseOver()
           Gets the onMouseOver javascript event.

public
String getOnMouseUp()
           Gets the onMouseUp javascript event.

public
String getOnSelect()
           Gets the onSelect javascript event.
```

```

    public getPassword()
boolean      Gets the password state (true means this is a password field).

    public
boolean getReadOnly()
            Gets if this TextBox is read-only.

    public
String getSize()
            Gets the size (in characters) of the TextBox.

    public
String getStyle()
            Gets the style of the rendered html tag.

    public
String getStyleClass()
            Gets the style class of the rendered html tag.

    public
String getTabIndex()
            Gets the tabIndex of the rendered html tag.

    public
String getTagId()
            Return the ID of the textbox.

    public
String getTagName()
            Return the name of the Tag.

protected
    void localRelease()
            Release any acquired resources.

    public
    void setAccessKey(String accessKey)
            Sets the accessKey attribute value.

    public
    void setAlt(String alt)
            Sets the alt attribute value.

    public
    void setAttribute(String name, String value)
            Set an attribute value.

    public
    void setDisabled(String disabled)
            Set if this TextBox is disabled or not.

    public
    void setMaxlength(String maxlength)
            Set the maximum length (in characters) of the TextBox.

    public
    void setOnBlur(String onblur)
            Sets the onBlur javascript event.

    public
    void setOnChange(String onchange)
            Sets the onChange javascript event.

```

```

public setOnClick(String onclick)
    void          Sets the onClick javascript event.
public
    void setOnDbClick(String ondblclick)
        Sets the onDbClick javascript event.
public
    void setOnFocus(String onblur)
        Sets the onFocus javascript event.
public
    void setOnKeyDown(String onkeydown)
        Sets the onKeyDown javascript event.
public
    void setOnKeyPress(String onkeypress)
        Sets the onKeyPress javascript event.
public
    void setOnKeyUp(String onkeyup)
        Sets the onKeyUp javascript event.
public
    void setOnMouseDown(String onmousedown)
        Sets the onMouseDown javascript event.
public
    void setOnMouseMove(String onmousemove)
        Sets the onMouseMove javascript event.
public
    void setOnMouseOut(String onmouseout)
        Sets the onMouseOut javascript event.
public
    void setOnMouseOver(String onmouseover)
        Sets the onMouseOver javascript event.
public
    void setOnMouseUp(String onmouseup)
        Sets the onMouseUp javascript event.
public
    void setOnSelect(String onselect)
        Sets the onSelect javascript event.
public
    void setPassword(boolean password)
        Set the password state (true means this is a password field).
public
    void setReadOnly(boolean readonly)
        Set if this TextBox is read-only.
public
    void setSize(String size)
        Set the size (in characters) of the TextBox.
public
    void setStyle(String style)
        Sets the style of the rendered html tag.

```

```

public setStyleClass(String styleClass)
    void          Sets the style class of the rendered html tag.
public
    void setTabIndex(String tabIndex)
        Sets the tabIndex of the rendered html tag.
public
    void setTagId(String tagId)
        Set the ID of the textbox.

```

**Methods from com.bea.wlw.netui.tags.DefaultableDataSourceTag**

getDefaultValue, setDefaultValue

**Methods from com.bea.wlw.netui.tags.DataSourceTag**

getDataSource, localRelease, setDataSource

**Methods from com.bea.wlw.netui.tags.AbstractBaseTag**

addTagIdMapping, evaluateAttributeToString, filter, filter,  
getJavaScriptUtils, getNearestForm, getScriptReporter, getUserLocale,  
localRelease, registerTagError, release, renderAttribute, rewriteName,  
write

**Methods from javax.servlet.jsp.tagext.BodyTagSupport**

doAfterBody, doInitBody, getBodyContent, getPreviousOut, release,  
setBodyContent

**Methods from javax.servlet.jsp.tagext.TagSupport**

doAfterBody, doEndTag, doStartTag, findAncestorWithClass, getId,  
getParent, getValue, getValues, release, removeValue, setId,  
setPageContext, setParent, setValue

**Methods from class java.lang.Object**

clone, equals, finalize, getClass, hashCode, notify, notifyAll,  
toString, wait, wait, wait

**Methods from interface javax.servlet.jsp.tagext.BodyTag**

doInitBody, setBodyContent

**Methods from interface com.bea.wlw.netui.tags.html.Formatable**

addFormatter

**Methods from interface com.bea.wlw.netui.tags.IAttributeConsumer**

setAttribute

**Methods from interface `javax.servlet.jsp.tagext.IterationTag`**

`doAfterBody`

**Methods from interface `javax.servlet.jsp.tagext.Tag`**

`doEndTag`, `doStartTag`, `getParent`, `release`, `setPageContext`, `setParent`

## Field Detail

### DISABLED

```
public static final String DISABLED
```

---

### MAXLENGTH

```
public static final String MAXLENGTH
```

---

### READONLY

```
public static final String READONLY
```

---

### SIZE

```
public static final String SIZE
```

---

### TYPE

```
public static final String TYPE
```

---

### VALUE

```
public static final String VALUE
```

## Constructor Detail

### TextBox

```
public TextBox()
```

## Method Detail

### addFormatter(FormatTag.Formatter) Method

```
public void addFormatter(FormatTag.Formatter formatter)
```

Adds a FormatTag.Formatter to the TextBox's set of formatters

#### Parameters

*formatter*

– a FormatTag.Formatter added by a child FormatTag.

---

### doEndTag() Method

```
public int doEndTag()  
    throws JspException
```

Render the TextBox.

#### Overrides

BodyTagSupport.doEndTag()

#### Exceptions

*JspException*

if a JSP exception has occurred

---

### doStartTag() Method

```
public int doStartTag()  
    throws JspException
```

Prepare the TextBox's formatters.

#### Overrides

BodyTagSupport.doStartTag()

#### Exceptions

*JspException*

if a JSP exception has occurred

---

### getAccessKey() Method

```
public String getAccessKey()
```

Gets the accessKey attribute value.

### Returns

*the accessKey value.*

---

## getAlt() Method

```
public String getAlt()
```

Gets the alt attribute on the generate <input tag.

### Returns

*the alt value.*

---

## getDisabled() Method

```
public String getDisabled()
```

Gets if this TextBox is disabled or not.

### Returns

*the disabled state (true or false)*

---

## getMaxlength() Method

```
public String getMaxlength()
```

Gets the maximum length (in characters) of the TextBox.

### Returns

*the max length*

---

## getOnBlur() Method

```
public String getOnBlur()
```

Gets the onBlur javascript event.

### Returns

*the onBlur event.*

---

## getOnChange() Method

```
public String getOnChange()
```

Gets the onChange javascript event.

### Returns

*the onChange event.*

---

## getOnClick() Method

```
public String getOnClick()
```

Gets the onClick javascript event.

### Returns

*the onClick event.*

---

## getOnDbClick() Method

```
public String getOnDbClick()
```

Gets the onDbClick javascript event.

### Returns

*the onDbClick event.*

---

## getOnFocus() Method

```
public String getOnFocus()
```

Gets the onFocus javascript event.

### Returns

*the onFocus event.*

---

## getOnKeyDown() Method

```
public String getOnKeyDown()
```

Gets the onKeyDown javascript event.



## Returns

*the onKeyDown event.*

---

## getOnKeyPress() Method

```
public String getOnKeyPress()
```

Gets the onKeyPress javascript event.

## Returns

*the onKeyPress event.*

---

## getOnKeyUp() Method

```
public String getOnKeyUp()
```

Gets the onKeyUp javascript event.

## Returns

*the onKeyUp event.*

---

## getOnMouseDown() Method

```
public String getOnMouseDown()
```

Gets the onMouseDown javascript event.

## Returns

*the onMouseDown event.*

---

## getOnMouseMove() Method

```
public String getOnMouseMove()
```

Gets the onMouseMove javascript event.

## Returns

*the onMouseMove event.*

---

## getOnMouseOut() Method

```
public String getOnMouseOut()
```

Gets the onMouseOut javascript event.

### Returns

*the onMouseOut event.*

---

## getOnMouseOver() Method

```
public String getOnMouseOver()
```

Gets the onMouseOver javascript event.

### Returns

*the onMouseOver event.*

---

## getOnMouseUp() Method

```
public String getOnMouseUp()
```

Gets the onMouseUp javascript event.

### Returns

*the onMouseUp event.*

---

## getOnSelect() Method

```
public String getOnSelect()
```

Gets the onSelect javascript event.

### Returns

*the onSelect event.*

---

## getPassword() Method

```
public boolean getPassword()
```

Gets the password state (true means this is a password field).

## Returns

*the password state*

---

## getReadonly() Method

```
public boolean getReadonly()
```

Gets if this TextBox is read-only.

## Returns

*the read-only state*

---

## getSize() Method

```
public String getSize()
```

Gets the size (in characters) of the TextBox.

## Returns

*the size*

---

## getStyle() Method

```
public String getStyle()
```

Gets the style of the rendered html tag.

## Returns

*the style.*

---

## getStyleClass() Method

```
public String getStyleClass()
```

Gets the style class of the rendered html tag.

## Returns

*the style class.*

---

## getTabIndex() Method

```
public String getTabIndex()
```

Gets the tabIndex of the rendered html tag.

### Returns

*the tabIndex.*

---

## getTagId() Method

```
public String getTagId()
```

Return the ID of the textbox.

### Returns

*the ID.*

---

## getTagName() Method

```
public String getTagName()
```

Return the name of the Tag.

### Overrides

```
AbstractBaseTag.getTagName()
```

---

## localRelease() Method

```
protected void localRelease()
```

Release any acquired resources.

### Overrides

```
DefaultableDataSourceTag.localRelease()
```

---

## setAccessKey(String) Method

```
public void setAccessKey(String accessKey)
```

Sets the accessKey attribute value. This should key value of the keyboard navigation key. It is recommended not to use the following values because there are often used by browsers A, C, E, F, G, H, V, left arrow, and right arrow.

## Parameters

*accessKey*  
– the accessKey value.

---

## setAlt(String) Method

```
public void setAlt(String alt)
```

Sets the alt attribute value.

## Parameters

*alt*  
– the alt value.

---

## setAttribute(String, String) Method

```
public void setAttribute(String name,  
                        String value)  
    throws JspException
```

Set an attribute value. The *name* represents the name of the attribute. The *value* represents the value and may contain a netui expression. This method may result in errors being generated. This requires that the tag buffer its body and write attributes in the end tag. For the textarea tag it is not legal to set the *id* or *name* attributes with this method.

## Parameters

*name*  
The name of the attribute. This value may not be null or the empty string.

*value*  
The value of the attribute. This may contain a netui expression.

## Exceptions

*JspException*  
A JspException may be thrown if there is an error setting the attribute.

---

## setDisabled(String) Method

```
public void setDisabled(String disabled)
```

Set if this TextBox is disabled or not.

**Parameters***disabled*– true or false

---

**setMaxLength(String) Method**

```
public void setMaxLength(String maxlength)
```

Set the maximum length (in characters) of the TextBox.

**Parameters***maxlength*– the max length

---

**setOnBlur(String) Method**

```
public void setOnBlur(String onblur)
```

Sets the onBlur javascript event.

**Parameters***onblur*– the onBlur event.

---

**setOnChange(String) Method**

```
public void setOnChange(String onchange)
```

Sets the onChange javascript event.

**Parameters***onchange*– the onChange event.

---

**setOnClick(String) Method**

```
public void setOnClick(String onclick)
```

Sets the onClick javascript event.

## Parameters

*onclick*

– the onClick event.

---

## setOnDbClick(String) Method

```
public void setOnDbClick(String ondblclick)
```

Sets the onDbClick javascript event.

## Parameters

*ondblclick*

– the onDbClick event.

---

## setOnFocus(String) Method

```
public void setOnFocus(String onblur)
```

Sets the onFocus javascript event.

## Parameters

*onblur*

– the onFocus event.

---

## setOnKeyDown(String) Method

```
public void setOnKeyDown(String onkeydown)
```

Sets the onKeyDown javascript event.

## Parameters

*onkeydown*

– the onKeyDown event.

---

## setOnKeyPress(String) Method

```
public void setOnKeyPress(String onkeypress)
```

Sets the onKeyPress javascript event.

**Parameters***onkeypress*

– the onKeyPress event.

**setOnKeyUp(String) Method**

```
public void setOnKeyUp(String onkeyup)
```

Sets the onKeyUp javascript event.

**Parameters***onkeyup*

– the onKeyUp event.

**setOnMouseDown(String) Method**

```
public void setOnMouseDown(String onmousedown)
```

Sets the onMouseDown javascript event.

**Parameters***onmousedown*

– the onMouseDown event.

**setOnMouseMove(String) Method**

```
public void setOnMouseMove(String onmousemove)
```

Sets the onMouseMove javascript event.

**Parameters***onmousemove*

– the onMouseMove event.

**setOnMouseOut(String) Method**

```
public void setOnMouseOut(String onmouseout)
```

Sets the onMouseOut javascript event.



## Parameters

*onmouseout*

– the onMouseOut event.

---

## setOnMouseOver(String) Method

```
public void setOnMouseOver(String onmouseover)
```

Sets the onMouseOver javascript event.

## Parameters

*onmouseover*

– the onMouseOver event.

---

## setOnMouseUp(String) Method

```
public void setOnMouseUp(String onmouseup)
```

Sets the onMouseUp javascript event.

## Parameters

*onmouseup*

– the onMouseUp event.

---

## setOnSelect(String) Method

```
public void setOnSelect(String onselect)
```

Sets the onSelect javascript event.

## Parameters

*onselect*

– the onSelect event.

---

## setPassword(boolean) Method

```
public void setPassword(boolean password)
```

Set the password state (true means this is a password field).

## Parameters

*password*

– the password state

---

## setReadOnly(boolean) Method

```
public void setReadOnly(boolean readonly)
```

Set if this TextBox is read-only.

## Parameters

*readonly*

– the read-only state

---

## setSize(String) Method

```
public void setSize(String size)
```

Set the size (in characters) of the TextBox.

## Parameters

*size*

– the size

---

## setStyle(String) Method

```
public void setStyle(String style)
```

Sets the style of the rendered html tag.

## Parameters

*style*

– the html style.

---

## setStyleClass(String) Method

```
public void setStyleClass(String styleClass)
```

Sets the style class of the rendered html tag.

**Parameters**

*styleClass*  
– the html style class.

---

**setTabIndex(String) Method**

```
public void setTabIndex(String tabIndex)
```

Sets the tabIndex of the rendered html tag.

**Parameters**

*tabindex*  
– the tab index.

---

**setTagId(String) Method**

```
public void setTagId(String tagId)
```

Set the ID of the textbox.

**Parameters**

*tagId*  
– the ID.

# Tree Class

*public class Tree*

extends DataSourceTag

Netui tag that renders a tree control represented by a set of `TreeNode` objects.

Attribute Descriptions			
Attribute	Required	Runtime Expression Evaluation	Data Bindable
<b>action</b>	Yes	No	No
	Hyperlink to which expand/contract actions should be sent.		
<b>imageHandleDownLast</b>	No	No	No
	The image name for an open non-leaf node with no line below it.		
<b>imageHandleDownMiddle</b>	No	No	No
	The image name for an open non-leaf node with a line below it.		
<b>imageHandleRightLast</b>	No	No	No
	The image name for a closed non-leaf node with no line below it.		
<b>imageHandleRightMiddle</b>	No	No	No
	The image name for a closed non-leaf node with a line below it.		
<b>imageLineLast</b>	No	No	No
	The image name for a blank area of the tree.		
<b>imageLineMiddle</b>	No	No	No
	The image name for an area with a line through it.		
<b>imageLineVertical</b>	No	No	No
	The image name for an area with a line through it.		
<b>imageRoot</b>	No	No	No
	The name of the directory containing the images		

## Page Flow API Reference

	for our icons, relative to the page including this tag.		
<b>style</b>	No	No	No
	CSS style class to be applied to be applied to the entire rendered output of the tree control.		
<b>styleSelected</b>	No	No	No
	CSS style class to be applied to the text of any element that is currently selected.		
<b>styleUnselected</b>	No	No	No
	CSS style class to be applied to the text of any element that is not currently selected.		
<b>tree</b>	Yes	No	Read/Write
	Attribute name in the session or expression under which the root TreeNode of the tree we are rendering is stored.		

### Related Topics

[<netui:tree> Tag Sample](#)  
[Node](#)  
[TreeNode](#)

---

### *Hierarchy*

```

Object
  TagSupport
    BodyTagSupport
      AbstractBaseTag
        DataSourceTag
          Tree
  
```

### *All Implemented Interfaces*

BodyTag, IterationTag, Serializable, Tag

---

## Field Summary

protected String

### *action*

The hyperlink to be used for submitting requests to expand and contract tree nodes.

protected String

### *imageHandleDownLast*

String

protected String ***imageHandleDownMiddle***  
String

protected String ***imageHandleRightLast***  
String

protected String ***imageHandleRightMiddle***  
String

protected String ***imageLineLast***  
String

protected String ***imageLineMiddle***  
String

protected String ***imageLineVertical***  
String

protected String ***imageRoot***  
The name of the directory containing the images for our icons, relative to the page including this tag.

protected TreeNode ***rootNode***  
TreeNode

protected String ***style***  
The CSS style class to be applied to the entire tree.

protected String ***styleSelected***  
The CSS style class to be applied to the text of selected nodes.

protected String ***styleUnselected***  
The CSS style class to be applied to the text of unselected nodes.

protected String ***tree***  
The name of the attribute (in the specified scope) under which our TreeControl instance is stored.

**Fields from com.bea.wlw.netui.tags.AbstractBaseTag**

ATTR\_GENERAL, ATTR\_GENERAL\_EXPRESSION, ATTR\_JAVASCRIPT, ATTR\_STYLE, CLASS, ID, JAVASCRIPT\_STATUS, NAME, NETUI\_UNIQUE\_CNT, ONCLICK, ONDBLCLICK, ONKEYDOWN, ONKEYPRESS, ONKEYUP, ONMOUSEDOWN, ONMOUSEMOVE, ONMOUSEOUT, ONMOUSEOVER, ONMOUSEUP, STYLE, TABINDEX

**Fields from javax.servlet.jsp.tagext.BodyTagSupport**

bodyContent

**Fields from `javax.servlet.jsp.tagext.TagSupport`**`id, pageContext`**Fields from interface `javax.servlet.jsp.tagext.BodyTag`**`EVAL_BODY_BUFFERED, EVAL_BODY_TAG`**Fields from interface `javax.servlet.jsp.tagext.IterationTag`**`EVAL_BODY_AGAIN`**Fields from interface `javax.servlet.jsp.tagext.Tag`**`EVAL_BODY_INCLUDE, EVAL_PAGE, SKIP_BODY, SKIP_PAGE`

## Constructor Summary

***Tree()***

## Method Summary

```

public
    int doEndTag()
        Render this Tree control.

public
    int doStartTag()
        Prepare the Tree for rendering.

public
    String getAction()
        Gets the action used for expanding and contracting tree nodes.

public
    String getImageHandleDownLast()
        Gets the image name for an open non-leaf node with no line below it.

public
    String getImageHandleDownMiddle()
        Gets the image name for an open non-leaf node with a line below it.

public
    String getImageHandleRightLast()
        Gets the image name for a closed non-leaf node with no line below it.

public
    String getImageHandleRightMiddle()
        Gets the image name for a closed non-leaf node with a line below it.

```

```

public getImageLineLast()
String      Gets the image name for a blank area of the tree.

public
String getImageLineMiddle()
           Gets the image name for an area with a line through it.

public
String getImageLineVertical()
           Gets the image name for an area with a line through it.

public
String getImageRoot()
           Gets the name of the directory containing the images for our icons, relative to the page
           including this tag.

public
TreeNode getRootNode()
           Gets the root TreeNode of this tree.

public
String getStyle()
           Gets the CSS style class to be applied to the entire tree.

public
String getStyleSelected()
           Gets the CSS style class to be applied to the text of selected nodes.

public
String getStyleUnselected()
           Gets the CSS style class to be applied to the text of unselected nodes.

public
String getTagName()
           Return the name of the Tag.

public
String getTree()
           Gets the name of the attribute (in the session), or expression, under which the root
           TreeNode instance is stored.

protected
void localRelease()
           Release all state information set by this tag.

public
void setAction(String action)
           Sets the action used for expanding and contracting tree nodes.

public
void setImageHandleDownLast(String imageHandleDownLast)
           Sets the image name for an open non-leaf node with no line below it.

public
void setImageHandleDownMiddle(String imageHandleDownMiddle)
           Sets the image name for an open non-leaf node with a line below it.

public
void setImageHandleRightLast(String imageHandleRightLast)
           Sets the image name for a closed non-leaf node with no line below it.

public

```



## Page Flow API Reference

```
void setImageHandleRightMiddle(String imageHandleRightMiddle)
    Sets the image name for a closed non-leaf node with a line below it.

public
void setImageLineLast(String imageLineLast)
    Sets the image name for a blank area of the tree.

public
void setImageLineMiddle(String imageLineMiddle)
    Sets the image name for an area with a line through it.

public
void setImageLineVertical(String imageLineVertical)
    Sets the image name for an area with a line through it.

public
void setImageRoot(String imageRoot)
    Sets the name of the directory containing the images for our icons, relative to the page
    including this tag.

public
void setRootNode(TreeNode rootNode)
    Sets the root TreeNode of this tree.

public
void setStyle(String style)
    Sets the CSS style class to be applied to the entire tree.

public
void setStyleSelected(String styleSelected)
    Sets the CSS style class to be applied to the text of selected nodes.

public
void setStyleUnselected(String styleUnselected)
    Sets the CSS style class to be applied to the text of unselected nodes.

public
void setTree(String tree)
    Sets the name of the attribute (in the session), or expression, under which the root
    TreeNode instance is stored.
```

### Methods from **com.bea.wlw.netui.tags.DataSourceTag**

`getDataSource`, `setDataSource`

### Methods from **com.bea.wlw.netui.tags.AbstractBaseTag**

`addTagIdMapping`, `evaluateAttributeToString`, `filter`, `filter`,  
`getJavaScriptUtils`, `getNearestForm`, `getScriptReporter`, `getUserLocale`,  
`localRelease`, `registerTagError`, `release`, `renderAttribute`, `rewriteName`,  
`write`

### Methods from **javax.servlet.jsp.tagext.BodyTagSupport**

`doAfterBody`, `doInitBody`, `getBodyContent`, `getPreviousOut`, `release`,  
`setBodyContent`

### Methods from **javax.servlet.jsp.tagext.TagSupport**

`doAfterBody`, `doEndTag`, `doStartTag`, `findAncestorWithClass`, `getId`,  
`getParent`, `getValue`, `getValues`, `release`, `removeValue`, `setId`,  
`setPageContext`, `setParent`, `setValue`

### Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`,  
`toString`, `wait`, `wait`, `wait`

### Methods from interface `javax.servlet.jsp.tagext.BodyTag`

`doInitBody`, `setBodyContent`

### Methods from interface `javax.servlet.jsp.tagext.IterationTag`

`doAfterBody`

### Methods from interface `javax.servlet.jsp.tagext.Tag`

`doEndTag`, `doStartTag`, `getParent`, `release`, `setPageContext`, `setParent`

## Field Detail

### **action**

`protected String action`

The hyperlink to be used for submitting requests to expand and contract tree nodes. The placeholder `"${name}"` will be replaced by the name property of the current tree node.

---

### **imageHandleDownLast**

`protected String imageHandleDownLast`

---

### **imageHandleDownMiddle**

`protected String imageHandleDownMiddle`

---

### **imageHandleRightLast**

`protected String imageHandleRightLast`

---

### **imageHandleRightMiddle**

`protected String imageHandleRightMiddle`

## **imageLineLast**

`protected String imageLineLast`

---

## **imageLineMiddle**

`protected String imageLineMiddle`

---

## **imageLineVertical**

`protected String imageLineVertical`

---

## **imageRoot**

`protected String imageRoot`

The name of the directory containing the images for our icons, relative to the page including this tag.

---

## **rootNode**

`protected TreeNode rootNode`

---

## **style**

`protected String style`

The CSS style `class` to be applied to the entire tree.

---

## **styleSelected**

`protected String styleSelected`

The CSS style `class` to be applied to the text of selected nodes.

---

## **styleUnselected**

`protected String styleUnselected`

The CSS style `class` to be applied to the text of unselected nodes.

---

**tree**

`protected String tree`

The name of the attribute (in the specified scope) under which our `TreeControl` instance is stored.

## Constructor Detail

### Tree

`public Tree()`

## Method Detail

### doEndTag() Method

`public int doEndTag()  
throws JspException`

Render this Tree control.

*Overrides*

`BodyTagSupport.doEndTag( )`

### Exceptions

*JspException*  
if a processing error occurs

---

### doStartTag() Method

`public int doStartTag()  
throws JspException`

Prepare the Tree for rendering.

*Overrides*

`BodyTagSupport.doStartTag( )`

### Exceptions

*JspException*  
if a JSP exception has occurred

---

## getAction() Method

```
public String getAction()
```

Gets the action used for expanding and contracting tree nodes.

### Returns

*the action*

---

## getImageHandleDownLast() Method

```
public String getImageHandleDownLast()
```

Gets the image name for an open non-leaf node with no line below it.

### Returns

*the image name (including extension)*

---

## getImageHandleDownMiddle() Method

```
public String getImageHandleDownMiddle()
```

Gets the image name for an open non-leaf node with a line below it.

### Returns

*the image name (including extension)*

---

## getImageHandleRightLast() Method

```
public String getImageHandleRightLast()
```

Gets the image name for a closed non-leaf node with no line below it.

### Returns

*the image name (including extension)*

---

## getImageHandleRightMiddle() Method

```
public String getImageHandleRightMiddle()
```

Gets the image name for a closed non-leaf node with a line below it.

## Returns

*the image name (including extension)*

---

## getImageLineLast() Method

```
public String getImageLineLast()
```

Gets the image name for a blank area of the tree.

## Returns

*the image name (including extension)*

---

## getImageLineMiddle() Method

```
public String getImageLineMiddle()
```

Gets the image name for an area with a line through it.

## Returns

*the image name (including extension)*

---

## getImageLineVertical() Method

```
public String getImageLineVertical()
```

Gets the image name for an area with a line through it.

## Returns

*the image name (including extension)*

---

## getImageRoot() Method

```
public String getImageRoot()
```

Gets the name of the directory containing the images for our icons, relative to the page including this tag.

## Returns

*the directory name*

---

## getRootNode() Method

```
public TreeNode getRootNode()
```

Gets the root `TreeNode` of this tree.

### Returns

*the root `TreeNode`*

---

## getStyle() Method

```
public String getStyle()
```

Gets the CSS style `class` to be applied to the entire tree.

### Returns

*the style*

---

## getStyleSelected() Method

```
public String getStyleSelected()
```

Gets the CSS style `class` to be applied to the text of selected nodes.

### Returns

*the selected style*

---

## getStyleUnselected() Method

```
public String getStyleUnselected()
```

Gets the CSS style `class` to be applied to the text of unselected nodes.

### Returns

*the unselected style*

---

## getTagName() Method

```
public String getTagName()
```

Return the name of the Tag.

**Overrides**

```
AbstractBaseTag.getTagName()
```

---

**getTree() Method**

```
public String getTree()
```

Gets the name of the attribute (in the session), or expression, under which the root `TreeNode` instance is stored.

**Returns**

*the tree attribute name*

---

**localRelease() Method**

```
protected void localRelease()
```

Release all state information set by this tag.

**Overrides**

```
DataSourceTag.localRelease()
```

---

**setAction(String) Method**

```
public void setAction(String action)
```

Sets the action used for expanding and contracting tree nodes.

**Parameters**

*action*

– the action

---

**setImageHandleDownLast(String) Method**

```
public void setImageHandleDownLast(String imageHandleDownLast)
```

Sets the image name for an open non-leaf node with no line below it. (Defaults to "handledownlast.gif").

**Parameters**

*imageHandleDownLast*

– the image name (including extension)

---



## setImageHandleDownMiddle(String) Method

```
public void setImageHandleDownMiddle(String imageHandleDownMiddle)
```

Sets the image name for an open non-leaf node with a line below it. (Defaults to "handledownmiddle.gif").

### Parameters

*imageHandleDownMiddle*

– the image name (including extension)

---

## setImageHandleRightLast(String) Method

```
public void setImageHandleRightLast(String imageHandleRightLast)
```

Sets the image name for a closed non-leaf node with no line below it. (Defaults to "handlerightlast.gif").

### Parameters

*imageHandleRightLast*

– the image name (including extension)

---

## setImageHandleRightMiddle(String) Method

```
public void setImageHandleRightMiddle(String imageHandleRightMiddle)
```

Sets the image name for a closed non-leaf node with a line below it. (Defaults to "handlerightmiddle.gif").

### Parameters

*imageHandleRightMiddle*

– the image name (including extension)

---

## setImageLineLast(String) Method

```
public void setImageLineLast(String imageLineLast)
```

Sets the image name for a blank area of the tree. (Defaults to "linelastnode.gif").

### Parameters

*imageLineLast*

– the image name (including extension)

---

## setImageLineMiddle(String) Method

```
public void setImageLineMiddle(String imageLineMiddle)
```

Sets the image name for an area with a line through it. (Defaults to "linemiddlenode.gif").

### Parameters

*imageLineMiddle*

– the image name (including extension)

---

## setImageLineVertical(String) Method

```
public void setImageLineVertical(String imageLineVertical)
```

Sets the image name for an area with a line through it. (Defaults to "linevertical.gif").

### Parameters

*imageLineVertical*

– the image name (including extension)

---

## setImageRoot(String) Method

```
public void setImageRoot(String imageRoot)
```

Sets the name of the directory containing the images for our icons, relative to the page including this tag.

### Parameters

*imageRoot*

– the directory name

---

## setRootNode(TreeNode) Method

```
public void setRootNode(TreeNode rootNode)
```

Sets the root `TreeNode` of this tree.

### Parameters

*rootNode*

– the root `TreeNode`

---

## setStyle(String) Method

```
public void setStyle(String style)
```

Sets the CSS style `class` to be applied to the entire tree.

### Parameters

*style*

– the style

---

## setStyleSelected(String) Method

```
public void setStyleSelected(String styleSelected)
```

Sets the CSS style `class` to be applied to the text of selected nodes.

### Parameters

*styleSelected*

– the selected style

---

## setStyleUnselected(String) Method

```
public void setStyleUnselected(String styleUnselected)
```

Sets the CSS style `class` to be applied to the text of unselected nodes.

### Parameters

*styleUnselected*

– the unselected style

---

## setTree(String) Method

```
public void setTree(String tree)
```

Sets the name of the attribute (in the session), or expression, under which the root `TreeNode` instance is stored.

### Parameters

*tree*

– the tree attribute name

# TreeNode Class

*public class* **TreeNode**

extends Object  
implements Serializable

An individual node of a Tree, and rendered by that Tree instance.

---

## **Hierarchy**

Object  
TreeNode

## **All Implemented Interfaces**

Serializable

---

## Field Summary

protected String

### **action**

The action invoked if this node is selected by the user.

protected ArrayList

### **children**

The set of child TreeControlNodes for this node, in the order that they should be displayed.

public static

final String **EXPAND\_NODE**

String

protected boolean

### **expanded**

Is this node currently expanded?

protected String

### **href**

String

protected String

### **icon**

The pathname to the icon file displayed when this node is visible, relative to the image directory for our images.

protected String

### **label**

The label that will be displayed when this node is visible.

protected boolean

***last***  
Is this the last node in the set of children for our parent node?

protected String

***name***  
The unique (within the entire tree) name of this node.

protected TreeNode

***parent***  
The parent node of this node, or null if this is the root node.

protected boolean

***selected***  
Is this node currently selected?

public static  
final String ***SELECTED\_NODE***  
String

protected String

***target***  
The window target for the hyperlink identified by the `action` property, if this node is selected by the user.

protected int

***width***  
The display width necessary to display this item (if it is visible).

## Constructor Summary

***TreeNode***(String icon, String label, String action, String href, String target, boolean expanded)

Construct a new `TreeNode` with the specified parameters.

## Method Summary

public  
void ***addChild***(TreeNode child)  
Add a new child node to the end of the list.

public  
void ***addChild***(int offset, TreeNode child)  
Add a new child node at the specified position in the child list.

public  
void ***collapse***()  
Code that runs when the node is collapsed.

public  
void ***expand***()  
Code that runs when the node is expanded.

```

    public getAction()
        String          Gets the action invoked if this node is selected by the user.

    public
    TreeNode getChild(int index)
        Return the child node at the given zero-relative index.

    public
    TreeNode[] getChildren()
        Return the set of child nodes for this node.

    public
    String getHref()
        Gets the hyperlink to which control will be directed if this node is selected by the user.

    public
    String getIcon()
        Gets the pathname to the icon file displayed when this node is visible, relative to the
        image directory for the images.

    public
    String getLabel()
        Gets the label that will be displayed when this node is visible.

    public
    TreeNode getParent()
        Gets the parent node of this node, or null if this is the root node.

    public
    TreeNode getRoot(TreeNode node)
        Gets the root node of this tree.

    public
    String getTarget()
        Gets the window target for the hyperlink identified by the action property, if this node
        is selected.

    public
    int getWidth()
        Gets the display width necessary to display this item (if it is visible).

    public
    boolean isExpanded()
        Gets if this node is currently expanded.

    public
    boolean isLast()
        Gets whether or not this is the last node in the set of children for the parent node.

    public
    boolean isLeaf()
        Gets whether or not this a "leaf" node (i.e. one with no children)

    public
    boolean isSelected()
        Gets whether this node currently selected.

    public
    void removeChild(int offset)
        Remove the child node (and all children of that child) at the specified position in the
        child list.

```

```
public removeChild(TreeNode child)
    void          Remove the specified child node.
public
    void setExpanded(boolean expanded)
        Sets if this node is currently expanded.
public
    void setParent(TreeNode parent)
        Sets the parent node of this node.
public
    void setSelected(boolean selected)
        Sets whether this node currently selected.
```

#### Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

## Field Detail

### action

`protected String action`

The action invoked if this node is selected by the user.

---

### children

`protected ArrayList children`

The set of child `TreeControlNodes` for this node, in the order that they should be displayed.

---

### EXPAND\_NODE

`public static final String EXPAND_NODE`

---

### expanded

`protected boolean expanded`

Is this node currently expanded?

---

### href

`protected String href`

---

## icon

`protected String icon`

The pathname to the icon file displayed when this node is visible, relative to the image directory for our images.

---

## label

`protected String label`

The label that will be displayed when this node is visible.

---

## last

`protected boolean last`

Is this the last node in the set of children for our parent node?

---

## name

`protected String name`

The unique (within the entire tree) name of this node.

---

## parent

`protected TreeNode parent`

The parent node of this node, or `null` if this is the root node.

---

## selected

`protected boolean selected`

Is this node currently selected?

---

## SELECTED\_NODE

`public static final String SELECTED_NODE`

---



## target

`protected String target`

The window target for the hyperlink identified by the `action` property, if this node is selected by the user.

---

## width

`protected int width`

The display width necessary to display this item (if it is visible). If this item is not visible, the calculated width will be that of our most immediately visible parent.

# Constructor Detail

## TreeNode

```
public TreeNode(String icon,
                String label,
                String action,
                String href,
                String target,
                boolean expanded)
```

Construct a new `TreeNode` with the specified parameters.

# Method Detail

## addChild(TreeNode) Method

```
public void addChild(TreeNode child)
    throws IllegalArgumentException
```

Add a new child node to the end of the list.

### Parameters

*child*

The new child node

### Exceptions

*IllegalArgumentException*

if the name of the new child node is not unique

## **addChild(int, TreeNode) Method**

```
public void addChild(int offset,  
                    TreeNode child)  
    throws IllegalArgumentException
```

Add a new child node at the specified position in the child list.

### **Parameters**

*offset*

Zero-relative offset at which the new node should be inserted

*child*

The new child node

### **Exceptions**

*IllegalArgumentException*

if the name of the new child node is not unique

---

## **collapse() Method**

```
public void collapse()
```

Code that runs when the node is collapsed.

---

## **expand() Method**

```
public void expand()
```

Code that runs when the node is expanded.

---

## **getAction() Method**

```
public String getAction()
```

Gets the action invoked if this node is selected by the user.

### **Returns**

*the action*

---

## getChild(int) Method

```
public TreeNode getChild(int index)
```

Return the child node at the given zero–relative index.

### Parameters

*index*

The child node index

### Returns

*the child node*

---

## getChildren() Method

```
public TreeNode[] getChildren()
```

Return the set of child nodes for this node.

### Returns

*the child node array*

---

## getHref() Method

```
public String getHref()
```

Gets the hyperlink to which control will be directed if this node is selected by the user.

### Returns

*the action*

---

## getIcon() Method

```
public String getIcon()
```

Gets the pathname to the icon file displayed when this node is visible, relative to the image directory for the images.

### Returns

*the icon pathname*

---

## getLabel() Method

```
public String getLabel()
```

Gets the label that will be displayed when this node is visible.

### Returns

*the label*

---

## getParent() Method

```
public TreeNode getParent()
```

Gets the parent node of this node, or `null` if this is the root node.

### Returns

*the `TreeNode` parent*

---

## getRoot(TreeNode) Method

```
public TreeNode getRoot(TreeNode node)
```

Gets the root node of this tree.

### Parameters

*node*

The `TreeNode` to start from

### Returns

*The root node*

---

## getTarget() Method

```
public String getTarget()
```

Gets the window target for the hyperlink identified by the `action` property, if this node is selected.

### Returns

*the window target*

---

## getWidth() Method

```
public int getWidth()
```

Gets the display width necessary to display this item (if it is visible). If this item is not visible, the calculated width will be that of our most immediately visible parent.

### Returns

*the display width*

---

## isExpanded() Method

```
public boolean isExpanded()
```

Gets if this node is currently expanded.

### Returns

*the expanded state*

---

## isLast() Method

```
public boolean isLast()
```

Gets whether or not this is the last node in the set of children for the parent node.

### Returns

*if this is the last or not*

---

## isLeaf() Method

```
public boolean isLeaf()
```

Gets whether or not this a "leaf" node (i.e. one with no children)

### Returns

*if this is the last or not*

---

## isSelected() Method

```
public boolean isSelected()
```

Gets whether this node currently selected.

**Returns**

*the selected state*

---

**removeChild(int) Method**

```
public void removeChild(int offset)
```

Remove the child node (and all children of that child) at the specified position in the child list.

**Parameters**

*offset*

Zero–relative offset at which the existing node should be removed

---

**removeChild(TreeNode) Method**

```
public void removeChild(TreeNode child)
```

Remove the specified child node. All of the children of this child node will also be removed.

**Parameters**

*child*

Child node to be removed

---

**setExpanded(boolean) Method**

```
public void setExpanded(boolean expanded)
```

Sets if this node is currently expanded.

**Parameters**

*expanded*

– the expanded state

---

**setParent(TreeNode) Method**

```
public void setParent(TreeNode parent)
```

Sets the parent node of this node.

## Parameters

*parent*

– the TreeNode parent

---

## setSelected(boolean) Method

```
public void setSelected(boolean selected)
```

Sets whether this node currently selected.

## Parameters

*selected*

– the selected state

---

# URLParams Interface

*public interface URLParams*

---

## *All Known Implementing Classes*

Anchor, Image, Form, Button, AnchorColumn

---

## Method

### Summary

```
public
    void addParameter(String name, Object value)
```

### Method Detail

#### addParameter(String, Object) Method

```
public void addParameter(String name,
                          Object value)
    throws JspException
```

#### Exceptions

*JspException*



# com.bea.wlw.netui.tags.template Package

This package defines a set of JSP tag extensions providing a simple templating mechanism. A template page is a JSP page defining the overall look and feel of a set of pages. It does this by providing an overall layout structure, style and design of the page and defines placeholders for content. These placeholders are called sections. A content page is a JSP page containing the content. The content page provides the sections of content plugged into the template. Combining the content page and the template page creates the overall look and feel of the content. The big advantage of templates is the ability to change the look and feel of a set of pages without being forced to modify each page.

The NetUI Template tags define two sets of tag extensions. The first set is used by within a JSP template page to define placeholder where content page provides content. The second set of tags is used within a content page to indicate the template to use and to define the content that will be presented within the template.

The tags used by a template page are:

- **IncludeSection** — This tag defines a placeholder for content.
- **Attribute** — This tag defines an attribute that may be set within the content page.

The tags used by a content page are:

- **Section** — This tag contains the content included within a section defined in the template.
- **SetAttribute** — This tag defines the value of an attribute used by a template.
- **Template** — This tag specifies which template page is being used for the content.

One additional tag is included in the package. The **Visible** tag defines a section that may be visible or not depending upon a boolean value. The tag may be used either within content or within a template.

## Interface Summary

TemplateConstants Define the constants used by the template tags.

## Class Summary

Attribute	The Attribute tag defines an attribute within a template that may be set from a content page.
IncludeSection	Used within a template JSP page to define a placeholder for section content.
Section	Used within a content page to provide content for a placeholder section defined within a template.
SetAttribute	Set an Attribute value defined in a template.
Template	This tags defines the template to use within a content page.
Template.TemplateContext	
Visible	Tag that turns on or off display of the body content based upon the visible state of the tag.

# Attribute Class

*public class Attribute*

extends TagSupport  
implements TemplateConstants

The Attribute tag defines an attribute within a template that may be set from a content page. For example, the page's title may be defined as an attribute in the template and then provided by each content page using the template. The attribute has a name and default value. If the content page specifies a value for the attribute it will be used, otherwise the default value is used.

Attribute Descriptions			
Attribute	Required	Runtime Expression Evaluation	Data Bindable
defaultValue	No	No	No
	A default value used for the Attribute if the content page does not provide a value.		
name	Yes	No	No
	The name identifying the Attribute. Any named attribute may be used more than once within a template page.		

## Related Topics

<netui-template:...> Tags Sample  
IncludeSection  
Section  
SetAttribute  
Template  
Visible

---

## Hierarchy

```

Object
  TagSupport
    Attribute
  
```

## All Implemented Interfaces

IterationTag, Serializable, Tag, TemplateConstants

---

## Field Summary

### Fields from `javax.servlet.jsp.tagext.TagSupport`

`id`, `pageContext`

### Fields from interface `javax.servlet.jsp.tagext.IterationTag`

`EVAL_BODY_AGAIN`

### Fields from interface `javax.servlet.jsp.tagext.Tag`

`EVAL_BODY_INCLUDE`, `EVAL_PAGE`, `SKIP_BODY`, `SKIP_PAGE`

### Fields from interface `com.bea.wlw.netui.tags.template.TemplateConstants`

`TEMPLATE_ATTRIBUTES`, `TEMPLATE_SECTIONS`

## Constructor Summary

*Attribute()*

## Method Summary

```

public
    int doStartTag()
        Renders the content of the attribute.

public
    String getDefaultValue()
        Returns the defaultValue of the
        Attribute.

public
    String getName()
        Returns the name of the Attribute.

protected
    void localRelease()
        Resets all of the fields of the tag.

public
    void setDefaultValue(String defaultValue)
        Sets the defaultValue for the Attribute.

public
    void setName(String name)
        Sets the name for the Attribute.

```

### Methods from `javax.servlet.jsp.tagext.TagSupport`

`doAfterBody`, `doEndTag`, `findAncestorWithClass`, `getId`, `getParent`,  
`getValue`, `getValues`, `release`, `removeValue`, `setId`, `setPageContext`,  
`setParent`, `setValue`

### Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`,  
`toString`, `wait`, `wait`, `wait`

### Methods from interface `javax.servlet.jsp.tagext.IterationTag`

`doAfterBody`

### Methods from interface `javax.servlet.jsp.tagext.Tag`

`doEndTag`, `doStartTag`, `getParent`, `release`, `setPageContext`, `setParent`

## Constructor Detail

### Attribute

```
public Attribute()
```

## Method Detail

### `doStartTag()` Method

```
public int doStartTag()  
    throws JspException
```

Renders the content of the attribute.

#### *Overrides*

```
TagSupport.doStartTag()
```

### Returns

*EVAL\_PAGE* to continue evaluation of the page.

### Exceptions

*JspException*

If there is any failure in the tag.

## getDefaultValue() Method

```
public String getDefaultValue()
```

Returns the `defaultValue` of the `Attribute`.

### Returns

*a String value representing the default value.*

---

## getName() Method

```
public String getName()
```

Returns the name of the `Attribute`.

### Returns

*the name of the Attribute*

---

## localRelease() Method

```
protected void localRelease()
```

Resets all of the fields of the tag.

---

## setDefaultValue(String) Method

```
public void setDefaultValue(String defaultValue)
```

Sets the `defaultValue` for the `Attribute`. If the content page does not define a value for this attribute through the `SetAttribute` tag, then the `defaultValue` will be used. If neither a value nor `defaultValue` is set, then the empty String "" will be output.

### Parameters

*defaultValue*

The value to set the `defaultValue` property.

---

## setName(String) Method

```
public void setName(String name)
```

Sets the name for the `Attribute`. An attribute may be used more than once in a template page.

## Parameters

*name*

The name of the attribute. The name does not need to be unique because it may be used more than once on the page.

# IncludeSection Class

*public class IncludeSection*

extends AbstractBaseTag  
implements TemplateConstants

Used within a template JSP page to define a placeholder for section content. Within a template, one or more sections are defined within the overall structure of the page. Each section has a unique name identifying it. The content page, through the `Section` tag, provides content that is included into the defined sections.

All content found within the body of the `IncludeSection` is ignored.

Attribute Descriptions			
Attribute	Required	Runtime Expression Evaluation	Data Bindable
name	Yes	No	No
	The name of the section placeholder. The name must be unique within a template.		
defaultPage	No	No	No
	A URL identifying a default JSP or HTML page to provide content if the content page does not.		

## Related Topics

<netui-template:...> Tags Sample  
Attribute  
Section  
SetAttribute  
Template  
Visible

---

## Hierarchy

```

Object
  TagSupport
    BodyTagSupport
      AbstractBaseTag
        IncludeSection
  
```

## All Implemented Interfaces

BodyTag, IterationTag, Serializable, Tag, TemplateConstants

---

## Field Summary

### Fields from `com.bea.wlw.netui.tags.AbstractBaseTag`

ATTR\_GENERAL, ATTR\_GENERAL\_EXPRESSION, ATTR\_JAVASCRIPT, ATTR\_STYLE, CLASS, ID, JAVASCRIPT\_STATUS, NAME, NETUI\_UNIQUE\_CNT, ONCLICK, ONDBLCLICK, ONKEYDOWN, ONKEYPRESS, ONKEYUP, ONMOUSEDOWN, ONMOUSEMOVE, ONMOUSEOUT, ONMOUSEOVER, ONMOUSEUP, STYLE, TABINDEX

### Fields from `javax.servlet.jsp.tagext.BodyTagSupport`

bodyContent

### Fields from `javax.servlet.jsp.tagext.TagSupport`

id, pageContext

### Fields from interface `javax.servlet.jsp.tagext.BodyTag`

EVAL\_BODY\_BUFFERED, EVAL\_BODY\_TAG

### Fields from interface `javax.servlet.jsp.tagext.IterationTag`

EVAL\_BODY\_AGAIN

### Fields from interface `javax.servlet.jsp.tagext.Tag`

EVAL\_BODY\_INCLUDE, EVAL\_PAGE, SKIP\_BODY, SKIP\_PAGE

### Fields from interface `com.bea.wlw.netui.tags.template.TemplateConstants`

TEMPLATE\_ATTRIBUTES, TEMPLATE\_SECTIONS

## Constructor Summary

*IncludeSection()*

## Method Summary

```
public
    int doStartTag()
        Renders the content of the section into the template.

public
    String getDefaultPage()
```



## Page Flow API Reference

Returns the URL of the JSP or HTML page providing default content to the defined section.

```
public  
String getName()
```

Returns the name of the defined section within the template file.

```
public  
String getTagName()
```

Returns the name of the Tag.

```
protected  
void localRelease()
```

Resets all of the fields of the tag.

```
public  
void setDefaultPage(String defaultPage)
```

Sets a default JSP page to provide content for the section if the content page does not define the content.

```
public  
void setName(String name)  
Sets the name of the section.
```

### Methods from `com.bea.wlw.netui.tags.AbstractBaseTag`

`addTagIdMapping`, `evaluateAttributeToString`, `filter`, `filter`,  
`getJavaScriptUtils`, `getNearestForm`, `getScriptReporter`, `getUserLocale`,  
`registerTagError`, `release`, `renderAttribute`, `rewriteName`, `write`

### Methods from `javax.servlet.jsp.tagext.BodyTagSupport`

`doAfterBody`, `doEndTag`, `doInitBody`, `getBodyContent`, `getPreviousOut`,  
`release`, `setBodyContent`

### Methods from `javax.servlet.jsp.tagext.TagSupport`

`doAfterBody`, `doEndTag`, `doStartTag`, `findAncestorWithClass`, `getId`,  
`getParent`, `getValue`, `getValues`, `release`, `removeValue`, `setId`,  
`setPageContext`, `setParent`, `setValue`

### Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`,  
`toString`, `wait`, `wait`, `wait`

### Methods from interface `javax.servlet.jsp.tagext.BodyTag`

`doInitBody`, `setBodyContent`

### Methods from interface `javax.servlet.jsp.tagext.IterationTag`

`doAfterBody`

### Methods from interface `javax.servlet.jsp.tagext.Tag`

doEndTag, doStartTag, getParent, release, setPageContext, setParent

## Constructor Detail

### IncludeSection

```
public IncludeSection()
```

## Method Detail

### doStartTag() Method

```
public int doStartTag()
    throws JspException
```

Renders the content of the section into the template. Errors are reported inline within the template in development mode. If no sections are defined an error is reported. If a section is not defined and no default URL is provided an error is reported.

#### *Overrides*

```
BodyTagSupport.doStartTag()
```

#### Returns

*SKIP\_BODY* to skip any content found in the tag.

#### Exceptions

*JspException*  
on Errors.

---

### getDefaultPage() Method

```
public String getDefaultPage()
```

Returns the URL of the JSP or HTML page providing default content to the defined section.

#### Returns

*the URL of the default content.*

---

## getName() Method

```
public String getName()
```

Returns the name of the defined section within the template file.

### Returns

*the name of the IncludeSection being defined.*

---

## getTagName() Method

```
public String getTagName()
```

Returns the name of the Tag. This is used to identify the type of tag reporting errors.

### Overrides

```
AbstractBaseTag.getTagName()
```

---

## localRelease() Method

```
protected void localRelease()
```

Resets all of the fields of the tag.

### Overrides

```
AbstractBaseTag.localRelease()
```

---

## setDefaultPage(String) Method

```
public void setDefaultPage(String defaultPage)
```

Sets a default JSP page to provide content for the section if the content page does not define the content.

### Parameters

*defaultPage*

a URL identifying a JSP or HTML page providing default content to the defined section.

---

## setName(String) Method

```
public void setName(String name)
```

Sets the name of the section. This name must be unique within the template page.

## Parameters

*name*

The name of the defined section within the template. This name must be unique within the template.

## Section Class

*public class Section*

extends AbstractBaseTag  
implements TemplateConstants

Used within a content page to provide content for a placeholder section defined within a template. The body content of the tag is passed to the IncludeSection tag in the template providing the content for that section. If the name attribute matches a name attribute on a IncludeSection tag in the template, the body content of this tag will be rendered.

Attribute Descriptions			
Attribute	Required	Runtime Expression Evaluation	Data Bindable
name	Yes	No	No
	The name of the IncludeSection tag in the template that the tags provides content for.		
visible	No	No	No
	A boolean property indicating if the content is visible or not.		
visibility	No	No	Read Only
	Sets the visible property through an expression.		

### Related Topics

<netui-template:...> Tags Sample  
Attribute  
IncludeSection  
SetAttribute  
Template  
Visible

---

### Hierarchy

```

Object
  TagSupport
    BodyTagSupport
      AbstractBaseTag
        Section
  
```

### All Implemented Interfaces

BodyTag, IterationTag, Serializable, Tag, TemplateConstants

## Field Summary

### Fields from `com.bea.wlw.netui.tags.AbstractBaseTag`

ATTR\_GENERAL, ATTR\_GENERAL\_EXPRESSION, ATTR\_JAVASCRIPT, ATTR\_STYLE, CLASS, ID, JAVASCRIPT\_STATUS, NAME, NETUI\_UNIQUE\_CNT, ONCLICK, ONDBLCLICK, ONKEYDOWN, ONKEYPRESS, ONKEYUP, ONMOUSEDOWN, ONMOUSEMOVE, ONMOUSEOUT, ONMOUSEOVER, ONMOUSEUP, STYLE, TABINDEX

### Fields from `javax.servlet.jsp.tagext.BodyTagSupport`

bodyContent

### Fields from `javax.servlet.jsp.tagext.TagSupport`

id, pageContext

### Fields from interface `javax.servlet.jsp.tagext.BodyTag`

EVAL\_BODY\_BUFFERED, EVAL\_BODY\_TAG

### Fields from interface `javax.servlet.jsp.tagext.IterationTag`

EVAL\_BODY\_AGAIN

### Fields from interface `javax.servlet.jsp.tagext.Tag`

EVAL\_BODY\_INCLUDE, EVAL\_PAGE, SKIP\_BODY, SKIP\_PAGE

### Fields from interface `com.bea.wlw.netui.tags.template.TemplateConstants`

TEMPLATE\_ATTRIBUTES, TEMPLATE\_SECTIONS

## Constructor Summary

*Section()*

## Method Summary

```
public
    int doEndTag()
        Stores the buffered body content into the TEMPLATE_SECTIONS HashMap.
```

```

public doStartTag()
    int          Causes the content of the section to be rendered into a buffer.

public
String getName()
    Get the name of the IncludeSection.

public
String getTagName()
    Returns the name of the Tag.

public
String getVisibility()
    Returns the expression used to set the visibility state.

public
boolean isVisible()
    Returns the visible state of the tag.

protected
void localRelease()
    Resets all of the fields of the tag.

public
void setName(String name)
    Sets the name of the placeholder section defined in the template that this tag is
    providing content for.

public
void setVisibility(String expr)
    Sets the visible state of the tag through an expression.

public
void setVisible(boolean visible)
    Sets the visible state of the tag.

```

#### Methods from `com.bea.wlw.netui.tags.AbstractBaseTag`

```

addTagIdMapping, evaluateAttributeToString, filter, filter,
getJavaScriptUtils, getNearestForm, getScriptReporter, getUserLocale,
registerTagError, release, renderAttribute, rewriteName, write

```

#### Methods from `javax.servlet.jsp.tagext.BodyTagSupport`

```

doAfterBody, doInitBody, getBodyContent, getPreviousOut, release,
setBodyContent

```

#### Methods from `javax.servlet.jsp.tagext.TagSupport`

```

doAfterBody, doEndTag, doStartTag, findAncestorWithClass, getId,
getParent, getValue, getValues, release, removeValue, setId,
setPageContext, setParent, setValue

```

#### Methods from class `java.lang.Object`

```

clone, equals, finalize, getClass, hashCode, notify, notifyAll,
toString, wait, wait, wait

```

**Methods from interface javax.servlet.jsp.tagext.BodyTag**

doInitBody, setBodyContent

**Methods from interface javax.servlet.jsp.tagext.IterationTag**

doAfterBody

**Methods from interface javax.servlet.jsp.tagext.Tag**

doEndTag, doStartTag, getParent, release, setPageContext, setParent

## Constructor Detail

### Section

```
public Section()
```

## Method Detail

### doEndTag() Method

```
public int doEndTag()
    throws JspException
```

Stores the buffered body content into the `TEMPLATE_SECTIONS` `HashMap`. The buffered body is accessed by the template page to obtain the content for `IncludeSection` tags.

**Overrides**

```
BodyTagSupport.doEndTag( )
```

**Returns**

*EVAL\_PAGE* to continue evaluating the page.

**Exceptions**

*JspException*  
on error.

---

### doStartTag() Method

```
public int doStartTag()
    throws JspException
```

Causes the content of the section to be rendered into a buffer.



**Overrides**

`BodyTagSupport.doStartTag()`

**Returns**

*SKIP\_BODY* if the visible state is *false*, otherwise *EVAL\_BODY\_BUFFERED* to cause the body content to be buffered.

**Exceptions**

*JspException*

if there are errors.

---

**getName() Method**

`public String getName()`

Get the name of the IncludeSection.

**Returns**

*the IncludeSection name.*

---

**getTagName() Method**

`public String getTagName()`

Returns the name of the Tag. This is used to identify the type of tag reporting errors.

**Overrides**

`AbstractBaseTag.getTagName()`

**Returns**

*a constant string representing the name of the tag.*

---

**getVisibility() Method**

`public String getVisibility()`

Returns the expression used to set the visibility state.

**Returns**

*a String expression*

---

## isVisible() Method

```
public boolean isVisible()
```

Returns the visible state of the tag.

### Returns

*true if the section is visible, false otherwise.*

---

## localRelease() Method

```
protected void localRelease()
```

Resets all of the fields of the tag.

### Overrides

```
AbstractBaseTag.localRelease()
```

---

## setName(String) Method

```
public void setName(String name)
```

Sets the name of the placeholder section defined in the template that this tag is providing content for. This name is matched against the IncludeSection name. If the names match, the content of this tag will be rendered within the template's section.

### Parameters

*name*

The name of an IncludeSection the this tag is providing content for.

---

## setVisibility(String) Method

```
public void setVisibility(String expr)
```

Sets the visible state of the tag through an expression.

### Parameters

*expr*

an expression will be checked to see if the section is visible or not.

---

## **setVisible(boolean) Method**

```
public void setVisible(boolean visible)
```

Sets the visible state of the tag.

### **Parameters**

*visible*

Boolean value representing the visible state.

# SetAttribute Class

*public class SetAttribute*

extends AbstractBaseTag  
implements TemplateConstants

Set an `Attribute` value defined in a template. This tag is used in content pages to set the value of attributes defined in a template. The attribute value will override any default value defined on the `Attribute`.

Attribute Descriptions			
Attribute	Required	Runtime Expression Evaluation	Data Bindable
name	Yes	No	No
	The name of the Attribute in the template for which this tag sets a value for.		
value	Yes	No	Read Only
	The value of the Attribute. This may be either a String value or an expression.		

## Related Topics

<netui-template:...> Tags Sample  
Attribute  
IncludeSection  
Section  
Template  
Visible

---

## Hierarchy

```

Object
  TagSupport
    BodyTagSupport
      AbstractBaseTag
        SetAttribute
  
```

## All Implemented Interfaces

BodyTag, IterationTag, Serializable, Tag, TemplateConstants

---

## Field Summary

### Fields from `com.bea.wlw.netui.tags.AbstractBaseTag`

ATTR\_GENERAL, ATTR\_GENERAL\_EXPRESSION, ATTR\_JAVASCRIPT, ATTR\_STYLE, CLASS, ID, JAVASCRIPT\_STATUS, NAME, NETUI\_UNIQUE\_CNT, ONCLICK, ONDBLCLICK, ONKEYDOWN, ONKEYPRESS, ONKEYUP, ONMOUSEDOWN, ONMOUSEMOVE, ONMOUSEOUT, ONMOUSEOVER, ONMOUSEUP, STYLE, TABINDEX

### Fields from `javax.servlet.jsp.tagext.BodyTagSupport`

bodyContent

### Fields from `javax.servlet.jsp.tagext.TagSupport`

id, pageContext

### Fields from interface `javax.servlet.jsp.tagext.BodyTag`

EVAL\_BODY\_BUFFERED, EVAL\_BODY\_TAG

### Fields from interface `javax.servlet.jsp.tagext.IterationTag`

EVAL\_BODY\_AGAIN

### Fields from interface `javax.servlet.jsp.tagext.Tag`

EVAL\_BODY\_INCLUDE, EVAL\_PAGE, SKIP\_BODY, SKIP\_PAGE

### Fields from interface `com.bea.wlw.netui.tags.template.TemplateConstants`

TEMPLATE\_ATTRIBUTES, TEMPLATE\_SECTIONS

## Constructor Summary

*SetAttribute()*

## Method Summary

```
public
    int doStartTag()
        Tag Lifecycle method called when the tag is first seen.

public
    String getName()
```

## Page Flow API Reference

Return the name of the `Attribute` we are setting an value for.

```
public  
String getTagName()
```

Return the name of the tag.

```
public  
String getValue()
```

Return the value value.

```
protected  
void localRelease()
```

Reset all of the fields of the tag.

```
public  
void setName(String name)
```

Set the name of the attribute.

```
public  
void setValue(String value)
```

Set the value of the `Attribute`.

### Methods from `com.bea.wlw.netui.tags.AbstractBaseTag`

`addTagIdMapping`, `evaluateAttributeToString`, `filter`, `filter`,  
`getJavaScriptUtils`, `getNearestForm`, `getScriptReporter`, `getUserLocale`,  
`registerTagError`, `release`, `renderAttribute`, `rewriteName`, `write`

### Methods from `javax.servlet.jsp.tagext.BodyTagSupport`

`doAfterBody`, `doEndTag`, `doInitBody`, `getBodyContent`, `getPreviousOut`,  
`release`, `setBodyContent`

### Methods from `javax.servlet.jsp.tagext.TagSupport`

`doAfterBody`, `doEndTag`, `doStartTag`, `findAncestorWithClass`, `getId`,  
`getParent`, `getValue`, `getValues`, `release`, `removeValue`, `setId`,  
`setPageContext`, `setParent`, `setValue`

### Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`,  
`toString`, `wait`, `wait`, `wait`

### Methods from interface `javax.servlet.jsp.tagext.BodyTag`

`doInitBody`, `setBodyContent`

### Methods from interface `javax.servlet.jsp.tagext.IterationTag`

`doAfterBody`

### Methods from interface `javax.servlet.jsp.tagext.Tag`

`doEndTag`, `doStartTag`, `getParent`, `release`, `setPageContext`, `setParent`

## Constructor Detail

### SetAttribute

```
public SetAttribute()
```

## Method Detail

### doStartTag() Method

```
public int doStartTag()  
    throws JspException
```

Tag Lifecycle method called when the tag is first seen. This method will add the `Attribute` value to a `HashMap` stored in the request allowing the template to access the value. If there are errors, the error text will be placed into the attribute value. Nothing is written into the `ServletResponse`

#### *Overrides*

```
BodyTagSupport.doStartTag()
```

#### Returns

*EVAL\_PAGE* to continue processing the page.

#### Exceptions

*JspException*  
on error

---

### getName() Method

```
public String getName()
```

Return the name of the `Attribute` we are setting an value for.

#### Returns

*A String value representing the name of the Attribute.*

---

## getTagName() Method

```
public String getTagName()
```

Return the name of the tag. This is used by error reporting in the base class `AbstractBaseTag`.

### *Overrides*

```
AbstractBaseTag.getTagName()
```

---

## getValue() Method

```
public String getValue()
```

Return the value value. If the value is an expression, the expression will be returned, not the value referred to by the expression.

### **Returns**

*a String value representing the Attribute value.*

---

## localRelease() Method

```
protected void localRelease()
```

Reset all of the fields of the tag.

### *Overrides*

```
AbstractBaseTag.localRelease()
```

---

## setName(String) Method

```
public void setName(String name)
```

Set the name of the attribute.

### **Parameters**

*name*

The name of the `Attribute` in the template for which this tags sets the value.

---

## setValue(String) Method

```
public void setValue(String value)
```

Set the value of the `Attribute`. This attribute may be assigned a read only expression.



## Parameters

*value*

The value to use for the `Attribute` in the template.

# Template Class

*public class **Template***

extends AbstractBaseTag  
implements ErrorReporter, TemplateConstants

This tag defines the template to use within a content page. The content page interacts with the template page through children tags of the `Template` tag. The legal children are as follows:

- `setAttribute` -- A Tag that will set an Attribute on the template.
- `section` -- A tag that defines the content of a section defined in the template.

The URL of the template file is set as the `templatePage` attribute on the `Template` tag. The `Template` file is included from the `Template` tag and will include sections defined in the content page. The content is contained in one or more `Section` tags, which are children of the `Template` tag. In addition, the content page can set attributes of the template page.

Attribute Descriptions			
Attribute	Required	Runtime Expression Evaluation	Data Bindable
<b>templatePage</b>	Yes	No	No
	The URL of the template JSP. This is a URL that locates the template JSP page.		
<b>reportErrors</b>	No	No	No
	Does the template contain a <code>&lt;netui:html&gt;</code> tag that should report tag errors?		

## Related Topics

`<netui-template:...>` Tags Sample  
Attribute  
IncludeSection  
Section  
SetAttribute  
Visible

---

## Hierarchy

```

Object
  TagSupport
    BodyTagSupport
      AbstractBaseTag
        Template
  
```

*All Implemented Interfaces*BodyTag, ErrorReporter, IterationTag, Serializable, Tag, TemplateConstants

---

## Nested Class Summary

```

    public
    static class Template.TemplateContext

```

## Field Summary

### Fields from `com.bea.wlw.netui.tags.AbstractBaseTag`

```

ATTR_GENERAL, ATTR_GENERAL_EXPRESSION, ATTR_JAVASCRIPT, ATTR_STYLE,
CLASS, ID, JAVASCRIPT_STATUS, NAME, NETUI_UNIQUE_CNT, ONCLICK,
ONDBLCLICK, ONKEYDOWN, ONKEYPRESS, ONKEYUP, ONMOUSEDOWN, ONMOUSEMOVE,
ONMOUSEOUT, ONMOUSEOVER, ONMOUSEUP, STYLE, TABINDEX

```

### Fields from `javax.servlet.jsp.tagext.BodyTagSupport`

```

bodyContent

```

### Fields from `javax.servlet.jsp.tagext.TagSupport`

```

id, pageContext

```

### Fields from interface `javax.servlet.jsp.tagext.BodyTag`

```

EVAL_BODY_BUFFERED, EVAL_BODY_TAG

```

### Fields from interface `com.bea.wlw.netui.tags.ErrorReporter`

```

CONTAINER_ERRORS

```

### Fields from interface `javax.servlet.jsp.tagext.IterationTag`

```

EVAL_BODY_AGAIN

```

### Fields from interface `javax.servlet.jsp.tagext.Tag`

```

EVAL_BODY_INCLUDE, EVAL_PAGE, SKIP_BODY, SKIP_PAGE

```

### Fields from interface `com.bea.wlw.netui.tags.template.TemplateConstants`

```

TEMPLATE_ATTRIBUTES, TEMPLATE_SECTIONS

```

## Constructor

## Summary

*Template()*

## Method

## Summary

```

public
    void addError(AbstractPageError ape)
        Add an error to the errors being reported by this tag.

public
    int doEndTag()
        The tag extension lifecycle method called after the tag has processed the body.

public
    int doStartTag()
        the tag extension lifecycle method called when the tag is first encountered.

public
boolean getReportErrors()
    Return a boolean indicating that the template expects the template file to report
    errors

public
String getTagName()
    Returns the name of the Tag.

public
String getTemplatePage()
    Return the URL of the template JSP page.

public
boolean isReporting()
    This boolean indicates if an ErrorReporter is reporting errors or not.

protected
    void localRelease()
        Reset all of the fields of the tag.

public
ArrayList returnErrors()
    Return an ArrayList of the errors

public
    void setReportErrors(boolean reportErrors)
        Set the boolean indicating that the template should report errors.

public
    void setTemplatePage(String templatePage)
        Set the URL of the template to use.

```

Methods from `com.bea.wlw.netui.tags.AbstractBaseTag`

`addTagIdMapping, evaluateAttributeToString, filter, filter,  
getJavaScriptUtils, getNearestForm, getScriptReporter, getUserLocale,  
registerTagError, release, renderAttribute, rewriteName, write`

### Methods from `javax.servlet.jsp.tagext.BodyTagSupport`

`doAfterBody, doInitBody, getBodyContent, getPreviousOut, release,  
setBodyContent`

### Methods from `javax.servlet.jsp.tagext.TagSupport`

`doAfterBody, doEndTag, doStartTag, findAncestorWithClass, getId,  
getParent, getValue, getValues, release, removeValue, setId,  
setPageContext, setParent, setValue`

### Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll,  
toString, wait, wait, wait`

### Methods from interface `javax.servlet.jsp.tagext.BodyTag`

`doInitBody, setBodyContent`

### Methods from interface `com.bea.wlw.netui.tags.ErrorReporter`

`addError, isReporting, returnErrors`

### Methods from interface `javax.servlet.jsp.tagext.IterationTag`

`doAfterBody`

### Methods from interface `javax.servlet.jsp.tagext.Tag`

`doEndTag, doStartTag, getParent, release, setPageContext, setParent`

## Constructor Detail

### Template

```
public Template()
```

## Method Detail

## addError(AbstractPageError) Method

```
public void addError(AbstractPageError ape)
```

Add an error to the errors being reported by this tag.

### Parameters

*ape*

– The AbstractPageError to add

---

## doEndTag() Method

```
public int doEndTag()
    throws JspException
```

The tag extension lifecycle method called after the tag has processed the body. This method will include the template JSP page specified by the `templatePage` attribute. The contents of the template are made available to the template page.

### Overrides

`BodyTagSupport.doEndTag()`

### Returns

*SKIP\_PAGE* to skip all processing after the template.

### Exceptions

*JspException*

on all errors. The most common error is an error indicating that the JSP page representing the Template isn't found.

---

## doStartTag() Method

```
public int doStartTag()
    throws JspException
```

the tag extension lifecycle method called when the tag is first encountered. This will cause the body of the tag to be evaluated.

### Overrides

`BodyTagSupport.doStartTag()`

### Returns

*int* indicating that the body should be evaluated.

## Exceptions

*JspException*  
on errors.

---

## getReportErrors() Method

```
public boolean getReportErrors()
```

Return a boolean indicating that the template expects the template file to report errors

---

## getTagName() Method

```
public String getTagName()
```

Returns the name of the Tag. This is used to identify the type of tag reporting errors.

### Overrides

```
AbstractBaseTag.getTagName()
```

---

## getTemplatePage() Method

```
public String getTemplatePage()
```

Return the URL of the template JSP page.

### Returns

*a String representing the URL of the JSP template.*

---

## isReporting() Method

```
public boolean isReporting()
```

This boolean indicates if an ErrorReporter is reporting errors or not. The caller should check this before calling addError because the ErrorReporter may be off for some reason.

### Returns

*a boolean indicating if the tag is reporting errors or not.*

---

## localRelease() Method

```
protected void localRelease()
```

Reset all of the fields of the tag.

**Overrides**

`AbstractBaseTag.localRelease()`

---

**returnErrors() Method**

`public ArrayList returnErrors()`

Return an ArrayList of the errors

**Returns**

*an ArrayList of all errors.*

---

**setReportErrors(boolean) Method**

`public void setReportErrors(boolean reportErrors)`

Set the boolean indicating that the template should report errors. The errors will be reported through a HTML tag that must be found in the template JSP.

**Parameters**

*reportErrors*

boolean indicating that errors should be reported

---

**setTemplatePage(String) Method**

`public void setTemplatePage(String templatePage)`

Set the URL of the template to use. The `templatePage` is an URL which identifies the JSP template page.

**Parameters**

*templatePage*

– a URL pointing to a JSP page that represents the template.



# Template.TemplateContext Class

*public static class Template.TemplateContext*

extends Object

---

## Hierarchy

Object  
Template.TemplateContext

## Enclosing class

Template

---

## Constructor Summary

*Template.TemplateContext()*

## Method Summary

Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`,  
`toString`, `wait`, `wait`, `wait`

## Constructor Detail

### Template.TemplateContext

```
public Template.TemplateContext()
```

# TemplateConstants Interface

*public interface **TemplateConstants***

Define the constants used by the template tags. Mostly these are the names used to store objects into the `ServletRequest`. These objects are used during the process of generating a page with a template.

---

## *All Known Implementing Classes*

Visible, Template, SetAttribute, Section, IncludeSection, Attribute

---

## Field Summary

```
public
static TEMPLATE_ATTRIBUTES
final String          This is the name of a HashMap containing all of the attributes.

public
static TEMPLATE_SECTIONS
final String          This is the name of a HashMap containing all of the sections.
```

## Field Detail

### TEMPLATE\_ATTRIBUTES

```
public static final String TEMPLATE_ATTRIBUTES
```

This is the name of a HashMap containing all of the attributes. The HashMap will be stored in the `ServletRequest`.

---

### TEMPLATE\_SECTIONS

```
public static final String TEMPLATE_SECTIONS
```

This is the name of a HashMap containing all of the sections. The HashMap will be stored in the `ServletRequest`.

# Visible Class

*public class Visible*

extends AbstractBaseTag  
implements TemplateConstants

Tag that turns on or off display of the body content based upon the visible state of the tag.

Attribute Descriptions			
Attribute	Required	Runtime Expression Evaluation	Data Bindable
visible	No	No	No
	A boolean property indicating if the content is visible or not.		
visibility	No	No	Read Only
	Sets the visible property through an expression.		
negate	No	No	No
	Sets the negate property which will negate the visible state.		

## Related Topics

<netui-template:...> Tags Sample  
Attribute  
IncludeSection  
Section  
SetAttribute  
Template

---

## Hierarchy

```

Object
  TagSupport
    BodyTagSupport
      AbstractBaseTag
        Visible
  
```

## All Implemented Interfaces

BodyTag, IterationTag, Serializable, Tag, TemplateConstants

---

## Field Summary

### Fields from `com.bea.wlw.netui.tags.AbstractBaseTag`

ATTR\_GENERAL, ATTR\_GENERAL\_EXPRESSION, ATTR\_JAVASCRIPT, ATTR\_STYLE, CLASS, ID, JAVASCRIPT\_STATUS, NAME, NETUI\_UNIQUE\_CNT, ONCLICK, ONDBLCLICK, ONKEYDOWN, ONKEYPRESS, ONKEYUP, ONMOUSEDOWN, ONMOUSEMOVE, ONMOUSEOUT, ONMOUSEOVER, ONMOUSEUP, STYLE, TABINDEX

### Fields from `javax.servlet.jsp.tagext.BodyTagSupport`

bodyContent

### Fields from `javax.servlet.jsp.tagext.TagSupport`

id, pageContext

### Fields from interface `javax.servlet.jsp.tagext.BodyTag`

EVAL\_BODY\_BUFFERED, EVAL\_BODY\_TAG

### Fields from interface `javax.servlet.jsp.tagext.IterationTag`

EVAL\_BODY\_AGAIN

### Fields from interface `javax.servlet.jsp.tagext.Tag`

EVAL\_BODY\_INCLUDE, EVAL\_PAGE, SKIP\_BODY, SKIP\_PAGE

### Fields from interface `com.bea.wlw.netui.tags.template.TemplateConstants`

TEMPLATE\_ATTRIBUTES, TEMPLATE\_SECTIONS

## Constructor Summary

*Visible()*

## Method Summary

```
public
    int doStartTag()
        Renders the body content if the visible state is
        true.
```

```

public getTagName()
String      Returns the name of the Tag.

public
String getVisibility()
           Returns the expression used to set the visibility state.

public
boolean isNegate()
           Returns the negate state of the tag.

public
boolean isVisible()
           Returns the visible state of the tag.

protected
void localRelease()
           Resets all of the fields of the tag.

public
void setNegate(boolean negate)
           Sets the negate state of the tag.

public
void setVisibility(String expr)
           Sets the visible state of the tag through an
           expression.

public
void setVisible(boolean visible)
           Sets the visible state of the tag.

```

#### Methods from **com.bea.wlw.netui.tags.AbstractBaseTag**

```

addTagIdMapping, evaluateAttributeToString, filter, filter,
getJavaScriptUtils, getNearestForm, getScriptReporter, getUserLocale,
registerTagError, release, renderAttribute, rewriteName, write

```

#### Methods from **javax.servlet.jsp.tagext.BodyTagSupport**

```

doAfterBody, doEndTag, doInitBody, getBodyContent, getPreviousOut,
release, setBodyContent

```

#### Methods from **javax.servlet.jsp.tagext.TagSupport**

```

doAfterBody, doEndTag, doStartTag, findAncestorWithClass, getId,
getParent, getValue, getValues, release, removeValue, setId,
setPageContext, setParent, setValue

```

#### Methods from class **java.lang.Object**

```

clone, equals, finalize, getClass, hashCode, notify, notifyAll,
toString, wait, wait, wait

```

#### Methods from interface **javax.servlet.jsp.tagext.BodyTag**

```

doInitBody, setBodyContent

```

#### Methods from interface `javax.servlet.jsp.tagext.IterationTag`

`doAfterBody`

#### Methods from interface `javax.servlet.jsp.tagext.Tag`

`doEndTag`, `doStartTag`, `getParent`, `release`, `setPageContext`, `setParent`

## Constructor Detail

### Visible

```
public Visible()
```

## Method Detail

### `doStartTag()` Method

```
public int doStartTag()  
    throws JspException
```

Renders the body content if the visible state is `true`. Otherwise nothing is rendered.

#### *Overrides*

```
BodyTagSupport.doStartTag()
```

#### Returns

*SKIP\_BODY* if the visible state is *false*, otherwise *EVAL\_BODY\_INCLUDE* to evaluate the body content.

#### Exceptions

*JspException*  
if there are errors.

---

### `getTagName()` Method

```
public String getTagName()
```

Returns the name of the Tag. This is used to identify the type of tag reporting errors.

#### *Overrides*

```
AbstractBaseTag.getTagName()
```

**Returns**

*a constant string representing the name of the tag.*

---

**getVisibility() Method**

```
public String getVisibility()
```

Returns the expression used to set the visibility state.

**Returns**

*a String expression*

---

**isNegate() Method**

```
public boolean isNegate()
```

Returns the negate state of the tag.

**Returns**

*true if the isVisible should be negated, false otherwise.*

---

**isVisible() Method**

```
public boolean isVisible()
```

Returns the visible state of the tag.

**Returns**

*true if the body content is visible, false otherwise.*

---

**localRelease() Method**

```
protected void localRelease()
```

Resets all of the fields of the tag.

**Overrides**

```
AbstractBaseTag.localRelease()
```

---

## setNegate(boolean) Method

```
public void setNegate(boolean negate)
```

Sets the negate state of the tag. If negate is `true` this will negate the meaning of the visible state.

### Parameters

*negate*

Boolean value representing the negate state.

---

## setVisibility(String) Method

```
public void setVisibility(String expr)
    throws JspException
```

Sets the visible state of the tag through an expression.

### Parameters

*expr*

an expression will be checked to see if the body content is visible or not.

### Exceptions

*JspException*

---

## setVisible(boolean) Method

```
public void setVisible(boolean visible)
```

Sets the visible state of the tag.

### Parameters

*visible*

Boolean value representing the visible state.



# **com.bea.wlw.netui.util Package**

## **Class Summary**

Bundle	Convenience class for dealing with resource bundles.
HtmlExceptionFormatter	Format a Throwable object so that it displays well in HTML.
PopulateContextHelper	
TemplateHelper	

# Bundle Class

*public class **Bundle***

extends `Object`

Convenience class for dealing with resource bundles.

---

## *Hierarchy*

`Object`  
`Bundle`

---

## Method Summary

```
public
static String getErrorString(String aKey, Object[] args)
public
static String getErrorString(String aKey)
public
static String getString(String aKey)
String      Returns the string specified by aKey from the errors.properties bundle.
public
static String getString(String aKey, Object arg)
String      Returns the string specified by aKey from the errors.properties bundle.
public
static String getString(String aKey, Object[] args)
String      Returns the string specified by aKey from the errors.properties bundle.
```

## Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`,  
`toString`, `wait`, `wait`, `wait`

## Method Detail

### **getErrorString(String, Object[]) Method**

```
public static String getErrorString(String aKey,
                                     Object[] args)
```

---

## getErrorString(String) Method

```
public static String getErrorString(String aKey)
```

---

## getString(String) Method

```
public static String getString(String aKey)
```

Returns the string specified by aKey from the errors.properties bundle.

---

## getString(String, Object) Method

```
public static String getString(String aKey,  
                               Object arg)
```

Returns the string specified by aKey from the errors.properties bundle.

### Parameters

*aKey*  
The key for the message pattern in the bundle.

*arg*  
The arg to use in the message format.

---

## getString(String, Object[]) Method

```
public static String getString(String aKey,  
                               Object[] args)
```

Returns the string specified by aKey from the errors.properties bundle.

### Parameters

*aKey*  
The key for the message pattern in the bundle.

*args*  
The args to use in the message format.

# HtmlExceptionFormatter Class

*public class **HtmlExceptionFormatter***

extends `Object`

Format a Throwable object so that it displays well in HTML.

---

## *Hierarchy*

`Object`

`HtmlExceptionFormatter`

---

## Constructor Summary

***HtmlExceptionFormatter()***

## Method Summary

`public`  
`static format(String message, Throwable t, boolean stackTrace)`  
`String`      Format an exception into XHTML.

Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`,  
`toString`, `wait`, `wait`, `wait`

## Constructor Detail

### HtmlExceptionFormatter

`public HtmlExceptionFormatter()`

## Method Detail

## **format(String, Throwable, boolean) Method**

```
public static String format(String message,  
                           Throwable t,  
                           boolean stackTrace)
```

Format an exception into XHTML. Optionally include a message and the stack trace.

# PopulateContextHelper Class

*public final class **PopulateContextHelper***

extends Object

---

## **Hierarchy**

Object  
PopulateContextHelper

---

## Constructor Summary

*PopulateContextHelper()*

## Method Summary

```
public
static
void populate(Map contexts, PageContext pageContext, Tag currentTag)

public
static populate(Map contexts, ServletRequest request, ServletResponse response,
void ActionForm actionForm)
```

## Methods from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructor Detail

### PopulateContextHelper

```
public PopulateContextHelper()
```

## Method Detail

### populate(Map, PageContext, Tag) Method

```
public static void populate(Map contexts,
                           PageContext pageContext,
                           Tag currentTag)
```

---

### populate(Map, ServletRequest, ServletResponse, ActionForm) Method

```
public static void populate(Map contexts,
                           ServletRequest request,
                           ServletResponse response,
                           ActionForm actionForm)
```

**DEPRECATED** Use `com.bea.wlw.netui.pageflow.util.TemplateHelper` instead.

# TemplateHelper Class

*public class TemplateHelper*

extends Object  
implements Cloneable

---

## **Hierarchy**

Object  
TemplateHelper

## **All Implemented Interfaces**

Cloneable

---

## **Constructor Summary**

*TemplateHelper()*

## **Method Summary**

Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`,  
`toString`, `wait`, `wait`, `wait`

## **Constructor Detail**

### **TemplateHelper**

`public TemplateHelper()`



# com.bea.wlw.netui.util.iterator Package

## Class Summary

ArrayIterator	ArrayIterator provides an Iterator over a Java array.
AtomicObjectIterator	
EmptyIterator	
EnumerationIterator	An Iterator implementation for rendering data from an Enumeration.
IteratorFactory	IteratorFactory provides a way to create an Iterator for different kinds of objects.
MapIterator	MapIterator provides an Iterator over map values.

## Exception Summary

IteratorFactoryException	An exception that is thrown when the IteratorFactory can not create an iterator for the given type.
--------------------------	---

# ArrayIterator Class

*public class ArrayIterator*

extends Object  
implements Iterator

ArrayIterator provides an Iterator over a Java array. ArrayIterator will return each element in the array in the order they are stored in the array. Multidimensional arrays are handled by returning a sequence of sub-arrays. So a three dimensional array of integers will return a sequence of two dimensional arrays of integers. This Iterator does not support the `remove()` method.

---

## Hierarchy

Object  
  ArrayIterator

## All Implemented Interfaces

Iterator

---

## Constructor Summary

*ArrayIterator*(Object array)

## Method Summary

```
public  
boolean hasNext()  
public  
Object next()  
public  
void remove()
```

### Methods from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Methods from interface java.util.Iterator

hasNext, next, remove

## Constructor Detail

### ArrayIterator

```
public ArrayIterator(Object array)
```

## Method Detail

### hasNext() Method

```
public boolean hasNext()
```

---

### next() Method

```
public Object next()
```

---

### remove() Method

```
public void remove()
```

# AtomicObjectIterator Class

*public class AtomicObjectIterator*

extends Object  
implements Iterator

---

## *Hierarchy*

Object  
AtomicObjectIterator

## *All Implemented Interfaces*

Iterator

---

## Method Summary

```
public  
boolean hasNext()  
public  
Object next()  
public  
void remove()
```

### Methods from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Methods from interface java.util.Iterator

hasNext, next, remove

## Method Detail

### hasNext() Method

```
public boolean hasNext()
```

---

## **next() Method**

```
public Object next()
```

---

## **remove() Method**

```
public void remove()
```

# EmptyIterator Class

*public class **EmptyIterator***

extends Object  
implements Iterator

---

## *Hierarchy*

Object  
EmptyIterator

## *All Implemented Interfaces*

Iterator

---

## Method Summary

```
public  
boolean hasNext()  
public  
Object next()  
public  
void remove()
```

### Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`,  
`toString`, `wait`, `wait`, `wait`

### Methods from interface `java.util.Iterator`

`hasNext`, `next`, `remove`

## Method Detail

### **hasNext()** Method

```
public boolean hasNext()
```

---

## **next() Method**

```
public Object next()
```

---

## **remove() Method**

```
public void remove()
```

# EnumerationIterator Class

*public class EnumerationIterator*

extends Object  
implements Iterator

An Iterator implementation for rendering data from an Enumeration.

---

## *Hierarchy*

Object  
EnumerationIterator

## *All Implemented Interfaces*

Iterator

---

## Constructor Summary

*EnumerationIterator*(Enumeration enum)

## Method Summary

public  
boolean **hasNext**()  
public  
Object **next**()  
public  
void **remove**()

### Methods from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Methods from interface java.util.Iterator

hasNext, next, remove



## Constructor Detail

### EnumerationIterator

```
public EnumerationIterator(Enumeration enum)
```

## Method Detail

### hasNext() Method

```
public boolean hasNext()
```

---

### next() Method

```
public Object next()
```

---

### remove() Method

```
public void remove()
```

# IteratorFactory Class

*public class **IteratorFactory***

extends Object

IteratorFactory provides a way to create an Iterator for different kinds of objects. The supported types are:

- java.util.Iterator
- java.util.Collection
- java.util.Map
- javax.sql.RowSet
- java.util.Enumeration
- XMLRuntime.XMLList
- XMLRuntime.XML
- Any Java Object array

If a java.util.Iterator is supplied, the Iterator will simply be returned to the caller.

If an object type not listed above is supplied the object will be wrapped in an iterator that has a single item, the provided object.

---

## *Hierarchy*

Object  
IteratorFactory

---

## Field Summary

public  
static **EMPTY\_ITERATOR**  
final Iterator Iterator

## Constructor Summary

*IteratorFactory()*

## Method Summary

```
public
  static makeIterator(Object object)
Iterator      Create a new Iterator for the supplied object.
```

### Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

## Field Detail

### EMPTY\_ITERATOR

```
public static final Iterator EMPTY_ITERATOR
```

## Constructor Detail

### IteratorFactory

```
public IteratorFactory()
```

## Method Detail

### makeIterator(Object) Method

```
public static Iterator makeIterator(Object object)
    throws IteratorFactoryException
```

Create a new `Iterator` for the supplied object. If the passed object is null, this method will return null;

#### Parameters

*object*  
the object to build an iterator from

#### Returns

*an iterator for the supplied object or null*

## Exceptions

*IteratorFactoryException*

# IteratorFactoryException Class

*public class **IteratorFactoryException***

extends `Exception`

An exception that is thrown when the `IteratorFactory` can not create an iterator for the given type. The illegal type will be available from the `getType` call.

---

## **Hierarchy**

```
Object
  Throwable
    Exception
      IteratorFactoryException
```

## **All Implemented Interfaces**

`Serializable`

---

## **Constructor Summary**

***IteratorFactoryException**(String message, Class type)*

***IteratorFactoryException**(String message)*

## **Method Summary**

```
public
  Class getType()
public
  String getTypeName()
```

### **Methods from `java.lang.Throwable`**

```
fillInStackTrace, getCause, getLocalizedMessage, getMessage,
getStackTrace, initCause, printStackTrace, printStackTrace,
printStackTrace, setStackTrace, toString
```

### **Methods from class `java.lang.Object`**

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

## Constructor Detail

### IteratorFactoryException

```
public IteratorFactoryException(String message,  
                                Class type)
```

---

### IteratorFactoryException

```
public IteratorFactoryException(String message)
```

## Method Detail

### getType() Method

```
public Class getType()
```

---

### getTypeName() Method

```
public String getTypeName()
```

# MapIterator Class

*public class MapIterator*

extends Object  
implements Iterator

MapIterator provides an Iterator over map values. MapIterator iterates over the sequence of values stored in the map.

---

## **Hierarchy**

Object  
MapIterator

## **All Implemented Interfaces**

Iterator

---

## Constructor Summary

*MapIterator*(Map map)

## Method Summary

```
public  
boolean hasNext()  
public  
Object next()  
public  
void remove()
```

### Methods from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Methods from interface java.util.Iterator

hasNext, next, remove

## Constructor Detail

### MapIterator

```
public MapIterator(Map map)
```

## Method Detail

### hasNext() Method

```
public boolean hasNext()
```

---

### next() Method

```
public Object next()
```

---

### remove() Method

```
public void remove()
```



# **com.bea.wlw.netui.util.tags Package**

## **Class Summary**

**GroupOption** GroupOption is a simple JavaBean that can be used to fully specify the name, value, alt text and accesskey of either a `CheckBoxGroup` or a `RadioButtonGroup`.

# GroupOption Class

*public class GroupOption*

extends Object  
implements Serializable

GroupOption is a simple JavaBean that can be used to fully specify the name, value, alt text and accesskey of either a `CheckBoxGroup` or a `RadioButtonGroup`. The name will appear in the HTML after the checkbox or radio button. The name attribute is required for output by the tags. The value can either be set separately or will default to the name. Optionally an alt text attribute and accesskey value can be provided.

---

## *Hierarchy*

Object  
GroupOption

## *All Implemented Interfaces*

Serializable

---

## Constructor Summary

### *GroupOption()*

Default Constructor.

### *GroupOption(String name, String value, String alt, String accessKey)*

Construct a GroupOption setting all the values.

### *GroupOption(String name, String value)*

Construct a GroupOption setting the name and value.

### *GroupOption(String name)*

Construct a GroupOption setting the name.

## Method Summary

```

public
String getAccessKey()
    Get the accesskey of the option.

public
String getAlt()
    Get the alt text of the option.

public
String getName()
    Get the name of the option.

public
String getValue()
    Get the value of the option.

public
    void setAccessKey(String accessKey)
        Set the alt text of the option.

public
    void setAlt(String alt)
        Set the alt text of the option.

public
    void setName(String name)
        Set the name of the option which will appear next to the option.

public
    void setValue(String value)
        Set the value of the option.

```

### Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

## Constructor Detail

### GroupOption

```

public GroupOption()

    Default Constructor.

```

---

### GroupOption

```

public GroupOption(String name,
                   String value,
                   String alt,
                   String accessKey)

```

Construct a GroupOption setting all the values.

---

## GroupOption

```
public GroupOption(String name,  
                   String value)
```

Construct a GroupOption setting the name and value.

---

## GroupOption

```
public GroupOption(String name)
```

Construct a GroupOption setting the name.

## Method Detail

### getAccessKey() Method

```
public String getAccessKey()
```

Get the accesskey of the option.

#### Returns

*The accessKey of the option.*

---

### getAlt() Method

```
public String getAlt()
```

Get the alt text of the option.

#### Returns

*The alt text of the option.*

---

### getName() Method

```
public String getName()
```

Get the name of the option.

## Returns

*The name of the option that was set.*

---

## getValue() Method

```
public String getValue()
```

Get the value of the option. If no value has been set this method will return value assigned to the name property.

## Returns

*The value of the option.*

---

## setAccessKey(String) Method

```
public void setAccessKey(String accessKey)
```

Set the alt text of the option.

## Parameters

*accessKey*

The value that will be set for the accesskey attribute.

---

## setAlt(String) Method

```
public void setAlt(String alt)
```

Set the alt text of the option.

## Parameters

*alt*

The text that will be set for the alt attribute.

---

## setName(String) Method

```
public void setName(String name)
```

Set the name of the option which will appear next to the option.

## Parameters

*name*

The name of the created option.

---

## setValue(String) Method

```
public void setValue(String value)
```

Set the value of the option. This value will be written out as the `value` attribute.

## Parameters

*value*

The name of the created option.

# com.bea.wlw.netui.util.type Package

## Interface Summary

TypeConverter

## Class Summary

TypeUtils

## Exception Summary

TypeConversionException

A TypeConversionException is thrown when an error has occurred performing a type conversion in the `TypeUtils` class.

TypeConverterNotFoundException

# TypeConversionException Class

*public class TypeConversionException*

extends RuntimeException

A TypeConversionException is thrown when an error has occurred performing a type conversion in the TypeUtils class.

---

## **Hierarchy**

Object  
  Throwable  
    Exception  
      RuntimeException  
        TypeConversionException

## **All Implemented Interfaces**

Serializable

---

## **Constructor Summary**

### ***TypeConversionException()***

Construct a TypeConversionException.

### ***TypeConversionException(String message, Throwable cause)***

Construct a TypeConversionException.

### ***TypeConversionException(String message)***

Construct a TypeConversionException.

### ***TypeConversionException(Throwable cause)***

Construct a TypeConversionException.



## Method Summary

```
public
String getLocalizedMessage()
    Get a localized message describing the cause of this
    exception.

public
    void setLocalizedMessage(String localizedMessage)
        Set a localized message describing the cause of this exception.
```

### Methods from `java.lang.Throwable`

```
fillInStackTrace, getCause, getMessage, getStackTrace, initCause,
printStackTrace, printStackTrace, printStackTrace, setStackTrace,
toString
```

### Methods from class `java.lang.Object`

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll,
toString, wait, wait, wait
```

## Constructor Detail

### TypeConversionException

```
public TypeConversionException()

    Construct a TypeConversionException.
```

---

### TypeConversionException

```
public TypeConversionException(String message,
                               Throwable cause)

    Construct a TypeConversionException.
```

---

### TypeConversionException

```
public TypeConversionException(String message)

    Construct a TypeConversionException.
```

---

## TypeConversionException

```
public TypeConversionException(Throwable cause)
```

Construct a TypeConversionException.

## Method Detail

### getLocalizedMessage() Method

```
public String getLocalizedMessage()
```

Get a localized message describing the cause of this exception.

#### *Overrides*

```
Throwable.getLocalizedMessage()
```

#### Returns

*a localized message string*

---

### setLocalizedMessage(String) Method

```
public void setLocalizedMessage(String localizedMessage)
```

Set a localized message describing the cause of this exception.

#### Parameters

*localizedMessage*  
a localized message

com.bea.wlw.netui.util.type

# TypeConverter Interface

*public interface TypeConverter*

## Method

### Summary

```
public  
Object convertToObject(String value)
```

### Method Detail

#### convertToObject(String) Method

```
public Object convertToObject(String value)
```

# TypeConverterNotFoundException Class

*public class TypeConverterNotFoundException*

extends RuntimeException

---

## Hierarchy

Object  
  Throwable  
    Exception  
      RuntimeException  
        TypeConverterNotFoundException

## All Implemented Interfaces

Serializable

---

## Constructor Summary

*TypeConverterNotFoundException()*

*TypeConverterNotFoundException(String message, Throwable t)*

*TypeConverterNotFoundException(String message)*

*TypeConverterNotFoundException(Throwable t)*

## Method Summary

public  
String **getLocalizedMessage()**  
public  
void **setLocalizedMessage(String localizedMessage)**

Methods from `java.lang.Throwable`

`fillInStackTrace, getCause, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString`

### Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

## Constructor Detail

### TypeConverterNotFoundException

```
public TypeConverterNotFoundException()
```

---

### TypeConverterNotFoundException

```
public TypeConverterNotFoundException(String message,
                                     Throwable t)
```

---

### TypeConverterNotFoundException

```
public TypeConverterNotFoundException(String message)
```

---

### TypeConverterNotFoundException

```
public TypeConverterNotFoundException(Throwable t)
```

## Method Detail

### getLocalizedMessage() Method

```
public String getLocalizedMessage()
```

#### *Overrides*

```
Throwable.getLocalizedMessage()
```

---

### setLocalizedMessage(String) Method

```
public void setLocalizedMessage(String localizedMessage)
```

# TypeUtils Class

*public final class **TypeUtils***

extends Object

---

## Hierarchy

Object  
  TypeUtils

---

## Method

### Summary

```
public
static
  final convertToObject(String value, Class type)
Object
```

### Methods from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Method Detail

### convertToObject(String, Class) Method

```
public static final Object convertToObject(String value,
                                           Class type)
```

### Returns

*the Object result of converting the String to the type.*