

Escenic Widget Framework
Core Widgets Tag Reference
4.0.3-1

Table of Contents

1 Introduction	5
2 wf-core Tag Library	6
2.1 wf-core:exists	6
2.2 wf-core:findFragmentItem	6
2.3 wf-core:getArticleSummariesInGroup	7
2.4 wf-core:getRssFeed	7
2.5 wf-core:mapCSS	8
2.6 wf-core:resolveConfigSections	8
2.7 wf-core:resolveWireframe	8
2.8 wf-core:stripHTML	9
2.9 wf-core:getFileTypeStyleClass	9
2.10 wf-core:getEventOccurrenceListMap	10
2.11 wf-core:getArticlesInGroup	11
2.12 wf-core:getPicturesInGroup	11
2.13 wf-core:getTaggedArticleIdList	12
2.14 wf-core:getCurtailedText	13
2.15 wf-core:getGroupByName	13
2.16 wf-core:getArticlesInList	14
2.17 wf-core:getArticleFieldsMap	14
2.18 wf-core:getPresentationPool	15
2.19 wf-core:listSkins	15
2.20 wf-core:generatePageURL	15
2.21 wf-core:iterateViews	16
2.22 wf-core:resolveView	16
2.23 wf-core:image	17
2.24 wf-core:media	18
2.25 wf-core:renderFragment	19
2.26 wf-core:evalField	19
2.27 wf-core:getCaptchaCustomHTML	20
2.28 wf-core:getConfigSectionPool	20
2.29 wf-core:getFieldValueByKey	20
2.30 wf-core:getTagCloud	21
2.31 wf-core:getVideoPreviewImageUrl	21
2.32 wf-core:getYouTubeVideoUrl	22

2.33 wf-core:include	22
2.34 wf-core:includeESI	23
2.35 wf-core:isVideoArticle	23
2.36 wf-core:notPresent	24
2.37 wf-core:present	24
2.38 wf-core:renderContext	24
2.39 wf-core:renderItemsInArea	24
2.40 wf-core:showItems	25
2.41 wf-core:view	25
2.42 wf-core:widgetAjaxMarkup	26
2.43 wf-core:pageFragment	26
2.44 wf-core:iteratorView	26
2.45 wf-core:renderGridItem	27
2.46 wf-core:generateGroupId	27
2.47 wf-core:renderGroup	28
2.48 wf-core:profileContent	28
2.49 wf-core:profilePhoto	28
3 wf-functions Tag Library	30
3.1 wf-functions:getRssSource	30
3.2 wf-functions:getCaptchaHTML	30
3.3 wf-functions:convertFahrenheitToCelsius	30
3.4 wf-functions:getRelativeDate	30
3.5 wf-functions:getDateObject	31
3.6 wf-functions:getDateString	31
3.7 wf-functions:getTermFromTagIdentifier	31
3.8 wf-functions:getImageDimension	32
3.9 wf-functions:parseFeedForImage	32
3.10 wf-functions:getConfigSectionByInheritance	32
3.11 wf-functions:isContainedInList	32
3.12 wf-functions:getImageRepresentation	33
3.13 wf-functions:getSuitableRepresentation	33
3.14 wf-functions:isFieldAccessible	33
3.15 wf-functions:getDateDifference	34
3.16 wf-functions:handleLineBreaks	34
3.17 wf-functions:isValidName	34
3.18 wf-functions:isNumeric	34
3.19 wf-functions:filterArticleList	35

3.20 wf-functions:getSubList.....	35
3.21 wf-functions:getTopicURL.....	35
3.22 wf-functions:matchFileExtension.....	35
3.23 wf-functions:getDescBreakpointsAsList.....	36
3.24 wf-functions:getDefaultAspectVariant.....	36
3.25 wf-functions:getArticleSummary.....	36
3.26 wf-functions:findGroupByName.....	36
3.27 wf-functions:generatePageURLWithoutQueryString.....	37
3.28 wf-functions:isImageContent.....	37
3.29 wf-functions:getRelativeTime.....	37
3.30 wf-functions:getClientTypeDefinition.....	38
3.31 wf-functions:getImageSizes.....	38
3.32 wf-functions:tagPageUriWithSection.....	38
3.33 wf-functions:tagPageUriWithPublication.....	38
3.34 wf-functions:taggingEntity.....	39
3.35 wf-functions:tagUriHashCode.....	39
3.36 wf-functions:createSectionIDPath.....	39
3.37 wf-functions:hasProperty.....	40
3.38 wf-functions:hasLazyFragment.....	40
3.39 wf-functions:getRelatedElements.....	40
3.40 wf-functions:isMediaContent.....	40
3.41 wf-functions:getSection.....	41
3.42 wf-functions:getMasterWidgetNestingLimit.....	41
3.43 wf-functions:isValidMasterWidgetNestingLimit.....	41
3.44 wf-functions:decreaseMasterWidgetNestingLevel.....	42
3.45 wf-functions:increaseMasterWidgetNestingLevel.....	42
3.46 wf-functions:getMediaInfoLink.....	42
3.47 wf-functions:isEmbeddedContent.....	42
3.48 wf-functions:resolveAMPLink.....	43
3.49 wf-functions:getAMPGATrackerConfig.....	43
4 wf-util Tag Library.....	44
4.1 wf-util:wrap.....	44
4.2 wf-util:link.....	44
4.3 wf-util:use.....	45
4.4 wf-util:sanitize.....	45
4.5 wf-util:embedFileContent.....	45
5 Deprecated Tags.....	47

1 Introduction

This manual contains reference information about various tag library Escenic used by Widget Framework. It contains only reference information about those tags which are used by core widgets, and assumes you have a general understanding of the Widget Framework and how it works. Specifically, you should at least have read the [Widget Framework User Guide](#). If you are going to modify those tags or create new tag , then you also need to have a good understanding of both JSP and java.

The manual is primarily intended to be used by template developers who will used these tags to develop new widgets/templates.

2 wf-core Tag Library

This library contains widget framework core tags. These tags are used in the framework and various widgets. Widgets used these tags to fetch contents, perform business logic and render html.

2.1 wf-core:exists

This tag checks whether a particular controller, view, template exist or not.

Syntax

```
<wf-core:exists  
  id="..."  
  path="..."?  
  scope="..."?  
  type="..."?  
  view="..."?  
  widgetName="..."?/>
```

Attributes

id, mandatory

Names the scripting variable created by this tag, as well as the key value used to locate the bean added to the specified scope.

type

This field indicates which type of resource is looking for. There are currently four types:

widgetName

Specify widget name for which you are looking a particular controller or view or index.jsp file exist or not

path

This attribute contains path of the file which this tag will check.

view

View name which you are looking for

scope

Specify in which scope result will be set

2.2 wf-core:findFragmentItem

This tag returns the PresentationElement that matches with given fragment token. This tag is used for lazy loading of page fragments.

Syntax

```
<wf-core:findFragmentItem  
  fragmentToken="..."  
  var="..."/>
```

Attributes

var, mandatory

Result will be returned in that variable

fragmentToken, mandatory

Fragment token that will be matched. Each element in the template pool will be searched for this token.

2.3 wf-core:getArticleSummariesInGroup

This tag reads article summaries from a group

Syntax

```
<wf-core:getArticleSummariesInGroup
  contentType="..."?
  group="..."
  max="..."?
  scope="..."?
  var="..." />
```

Attributes

var, mandatory

In the content studio we can create group under an area and in the group we can place various content items. This tag will read article summaries of content items for the given group. If there is multiple content type item you can filter item also.

group, mandatory

the group name for which article summaries will be read

contentType

Filter content item by content type

max

Define how many items you want to fetch

scope

Specify in which scope result will be set

2.4 wf-core:getRssFeed

Will fetch RSS feed from specified URL.

Syntax

```
<wf-core:getRssFeed
  id="..."
  maxArticles="..."
  scope="..."?
  sourceUrls="..." />
```

Attributes

id, mandatory

The name of the attribute that will contain the list of feed entries

sourceUrls, mandatory

URLs from which feed will be fetched.

maxArticles, mandatory

Limits how many feed will be fetched

scope

Specify in which scope fetched feed will be set

2.5 wf-core:mapCSS

This tag will read css mappings from `CssMappingResources.properties` file.

Syntax

```
<wf-core:mapCSS  
  fileNamePrefix="..."  
  scope="..."?  
  var="..." />
```

Attributes

var, mandatory

Names the scripting variable created by this tag, as well as the key value used to locate the bean added to the specified scope.

fileNamePrefix, mandatory

Prefix of the properties file name from which mapping will be read

scope

Specify in which scope result will be set

2.6 wf-core:resolveConfigSections

This tag defines which template will be used for a section or article page. It traces the inheritance tree to find the correct template.

Syntax

```
<wf-core:resolveConfigSections />
```

2.7 wf-core:resolveWireframe

This tag used to identify which wireframe will be used. If nothing is specified in the parameter then **default** wireframe will be used.

Syntax

```
<wf-core:resolveWireframe
```



```
scope="..."?  
var="...">
```

Attributes

var, mandatory

scope

2.8 wf-core:stripHTML

Remove all HTML tags from given input string.

Syntax

```
<wf-core:stripHTML  
id="..."  
maxLength="..."?  
scope="..."?  
value="...">
```

Attributes

id, mandatory, no runtime expressions

The name of the attribute that will contain output string

value, mandatory

The input string that may content HTML tags

maxLength

Maximum length of the output string.

scope

Specify in which scope result will be set

2.9 wf-core:getFileTypeStyleClass

This tag gets file type style class names for related media files from their mime-types.

Syntax

```
<wf-core:getFileTypeStyleClass  
fileMimeType="..."?  
relatedObject="..."  
var="...">
```

Attributes

var, mandatory, no runtime expressions

The name of the attribute that will contain output string

relatedObject, mandatory

The object which mime-type is searched for class name.

fileMimeType

The mime-type of the related file.

2.10 wf-core:getEventOccurrenceListMap

This tag creates a list of events within given time frame.

Syntax

```
<wf-core:getEventOccurrenceListMap
  articleType="..."?
  id="..."
  max="..."?
  numberOfHours="..."?
  publicationName="..."?
  range="..."
  scheduleFieldName="..."?
  sortOrder="..."?
  sortString="..."?
  tagFieldName="..."?
  tagIdentifiers="..."?/>
```

Attributes

id, mandatory, no runtime expressions

This is the object where the result will be placed.

range, mandatory

Events within this range will be listed.

max

Maximum size of the list will be determined by this attribute.

tagIdentifiers

Events that have this tags are included in the result. If tagIdentifiers is not given, all events included.

tagFieldName

This name is used to query on solr for events.

numberOfHours

Events that will happen within this hour will be listed.

publicationName

This specifies from which publication, events have to fetch.

articleType

This attribute sets the content type of article to be listed.

scheduleFieldName

The name of the schedule field in solr index (which is derived from the name of the field in content-type definition).

sortOrder

This specifies the list sorting order whether this is ascending or descending.

sortString

This specifies on which field the results will be sorted.

2.11 wf-core:getArticlesInGroup

This tag lists all the articles in the mentioned group.

Syntax

```
<wf-core:getArticlesInGroup
  contentType="..."
  group="..."
  max="..."?
  var="..." />
```

Attributes

var, mandatory, no runtime expressions

This is the object where the results will be placed.

group, mandatory

The group name from where articles have to fetch.

contentType, mandatory

This denotes which type of articles have to fetch.

max

This denotes the maximum number of the articles to be fetched.

2.12 wf-core:getPicturesInGroup

This tag lists all the pictures in the mentioned group.

Syntax

```
<wf-core:getPicturesInGroup
  contentType="..."?
  group="..."
  includeArticleRelatedPictures="..."?
  max="..."?
  var="..." />
```

Attributes

var, mandatory, no runtime expressions

This is the object where the results will be placed.

group, mandatory

The group name from where pictures have to fetch.

contentType

This denotes which type of pictures have to fetch.

max

This denotes the maximum number of the articles to be fetched.

includeArticleRelatedPictures

This denotes whether to include related picture or not.

2.13 wf-core:getTaggedArticleIdList

This tag lists the IDs of all content items tagged with specified tags. The list can optionally be filtered and sorted in various ways.

Syntax

```
<wf-core:getTaggedArticleIdList
  articleTypes="..."?
  id="..."
  max="..."
  publicationName="..."?
  sectionId="..."?
  sortOrder="..."?
  sortString="..."?
  startIndex="..."
  tagFieldName="..."
  tagIdentifiers="..." />
```

Attributes

id, mandatory, no runtime expressions

Names the scripting variable created by this tag, as well as the key value used to locate the bean added to the specified scope.

tagIdentifiers, mandatory

The name of a List containing the identifiers of the tags to be used as search criteria.

tagFieldName, mandatory

The name of the tag scheme containing the tags to be used as search criteria. The suffix **_facet** must be appended to the tag scheme name as follows:

```
| tag-scheme-name_facet
```

max, mandatory

The maximum number of content item IDs to list.

startIndex, mandatory

The position from which returned content items are to be listed. The default value is **0**, meaning that the content items will be listed from the start of the sorted result set. A value of **2** would cause the first two content items in the result set to be skipped.

sortString

The name of the content item date property by which the results are to be sorted. The options are:

sectionId

The id of a section. If specified, only content items belonging to this section will be listed.

publicationName

The name of a publication. If specified, only content items belonging to this publication will be listed.

articleTypes

A comma-separated list of content type names. If specified, only content items of these types will be listed.

sortOrder

The numerical order in which the returned content items are to be sorted. The content items are sorted by the property specified with the **sortString** attribute. The options are:

2.14 wf-core:getCurtailedText

This tag is used to crop a input text to a given length.

Syntax

```
<wf-core:getCurtailedText  
  ellipsis="..."?  
  inputText="..."  
  maxLength="..."  
  var="..."/>
```

Attributes

var, mandatory, no runtime expressions

This is the object where the result will be placed.

inputText, mandatory

Input text that has to be crop to a certain length.

maxLength, mandatory

The length of the output text.

ellipsis

The tailing text of the output text.

2.15 wf-core:getGroupName

This tag gets the group as a PresentationElement object from the group name.

Syntax

```
<wf-core:getGroupName  
  areaName="..."  
  groupName="..."  
  pool="..."?  
  var="..."/>
```

Attributes

var, mandatory, no runtime expressions

This is the object where the result will be placed.

groupName, mandatory

Name of the group which will be searched.

areaName, mandatory

The name of the area where the group is located.

pool

The active PresentationPool object.

2.16 wf-core:getArticlesInList

This tag reads articles from a list of a section.

Syntax

```
<wf-core:getArticlesInList
  contentType="..."
  listName="..."
  max="..."?
  var="..."/>
```

Attributes

var, mandatory, no runtime expressions

This is the object where the result will be placed.

listName, mandatory

This is the list name from where the articles have to be fetched.

contentType, mandatory

This attribute tells which type of contents to be fetched.

max

This attribute tells maximum number of articles to be fetched.

2.17 wf-core:getArticleFieldsMap

This tag is used to make a map with the widget fields of all panels.

Syntax

```
<wf-core:getArticleFieldsMap
  articleTypeName="..."?
  content="..."
  publicationId="..."
  var="..."/>
```

Attributes

articleTypeName

Deprecated. Usage of this attribute should be removed.

publicationId, mandatory

This attribute is used to specify the publication.

content, mandatory

This sets the article from which the value have to be fetched.

var, mandatory, no runtime expressions

This is the object where the results will be placed.

2.18 wf-core:getPresentationPool

This tag gets the current presentation pool object.

Syntax

```
<wf-core:getPresentationPool
  section="..."?
  var="..."/>
```

Attributes

var, mandatory, no runtime expressions

This is the object where the result will be placed.

section

The current section object.

2.19 wf-core:listSkins

This tag will return available skins. Basically this tag will check available top level directories under the skin directory of publication. It will treat those top level directories as skin.

Syntax

```
<wf-core:listSkins
  id="..."/>
```

Attributes

id, mandatory

Names the scripting variable created by this tag, as well as the key value used to locate the bean added to the specified scope.

2.20 wf-core:generatePageURL

This tag will generate page URL from the existing page URL and parameters supplied as attribute to the tag.

Syntax

```
<wf-core:generatePageURL
  parameters="..."?
  var="..."/>
```

Attributes

var, mandatory, no runtime expressions

The variable where the generated page URL value will be stored.

parameters

A map of parameters to be added as query string to the generated URL.

2.21 wf-core:iterateViews

This tag iterates through a map of views to select a particular one for each item in a list of articles.

Syntax

```
<wf-core:iterateViews
  items="..."
  selectableViewMap="..."
  var="..."
  varIndex="..."?>
  ...
</wf-core:iterateViews>
```

Attributes

var, mandatory, no runtime expressions

The scripting variable through which the selected view is exposed in **page** scope.

items, mandatory

A list of teaser items, a view is to be selected for each item in the list.

selectableViewMap, mandatory

The map of selectable views.

varIndex, no runtime expressions

A scripting variable that if set, can be used to access the index of the current item to be rendered by the selected view.

2.22 wf-core:resolveView

This tag iterates through a map of views to select a particular one for each item in a list of articles.

Syntax

```
<wf-core:resolveView
  item="..."
  selectableViewMap="..."
  var="..."
  varIndex="..."?>
  ...
</wf-core:resolveView>
```

Attributes

var, mandatory, no runtime expressions

The scripting variable through which the selected view is exposed in **page** scope.

item, mandatory

A ResultItem object

selectableViewMap, mandatory

The map of selectable views.

varIndex, no runtime expressions

A scripting variable that if set, can be used to access the index of the current item to be rendered by the selected view.

2.23 wf-core:image

This tag will create an `img` element with appropriate image representation.

Syntax

```
<wf-core:image
  alt="..."?
  article="..."?
  articleId="..."?
  articleTypeName="..."?
  aspectVariant="..."?
  imagePolicy="..."?
  onClick="..."?
  preload="..."?
  scale="..."?
  styleClass="..."?
  title="..."?
  width="..."?/>
```

Attributes

articleId

An article id identifying the image content item to be displayed. The id can be supplied as an `int`, `Integer` or `String`.

article

`Aneo.xredsys.presentation.PresentationArticle`: the content item representing the image to be displayed.

width

The width value to be used if `imagePolicy` is set to `grid`. If you specify both `width` and `scale`, then `scale` takes priority.

alt

If specified, sets the `alt` attribute of the generated `img` element. By default the `alt` attribute is taken from the source content item.

title

If specified, sets the `title` attribute of the generated `img` element. By default the `title` attribute is taken from the source content item.

scale

The image width specified as a percentage of the available space (that is, the width of the image's enclosing element).

articleTypeName

Deprecated. The type of the article.

styleClass

The name of a CSS style (HTML `class` attribute) to be assigned to the generated HTML `img` tag.

onClick

Deprecated. The onclick handler of the image.

imagePolicy

Specify **adaptive** or **picturefill** to enable responsive image selection or **grid** to disable it. If this attribute is not specified, the value is taken from **image.policy** content profile section parameter.

aspectVariant

Specifies the required image aspect. The allowed options are **PORTRAIT**, **LANDSCAPE**, **SQUARE** or **FREE**. If no value is specified, then a default of **LANDSCAPE** is used. (This default can be changed by setting the **defaultAspectVariant** property in **com/escenic/framework/cache/ImageRepresentationCacheManager.properties**.)

preload

Specify **placeholder** or **proxy**, if you want the image tag to load some placeholder or proxy images initially. Otherwise set **none** to disable preloading. If this attribute is not specified, the value is taken from **image.preload** content profile section parameter.

2.24 wf-core:media

This tag will render video in widget framework.

Syntax

```
<wf-core:media
  article="..."?
  articleID="..."?
  audioPlayer="..."?
  autoAdvance="..."?
  autoPlay="..."?
  keyframeAspectVariant="..."?
  playMode="..."?
  scale="..."?
  styleClass="..."?
  videoPlayer="..."?/>
```

Attributes

article

Aneo.xredsys.presentation.PresentationArticle: the content item representing the video to be played.

articleID

An article id identifying the media content item to be displayed.

styleClass

The name of a CSS style (HTML **class** attribute) to be assigned to the generated HTML **media** tag.

scale

The media width specified as a percentage of the available space (that is, the width of the media's enclosing element).

playMode

Specifies the play mode how audio and video content should be played. The allowed options are **content**, **embedded** or **popup**.

autoPlay

Specifies if the audio or video will be played automatically. The default value is **false**.

keyframeAspectVariant

Specifies the required media keyframe aspect. The allowed options are **PORTRAIT**, **LANDSCAPE**, **SQUARE** or **FREE**. If no value is specified, then a default of **LANDSCAPE** is used.

autoAdvance

Specifies if the next audio/video will be played automatically in playlist when one is finished. The default value is **true**.

videoPlayer

Specifies which video player instance should be used.

audioPlayer

Specific which audio player instance should be used.

2.25 wf-core:renderFragment

Renders a page fragment based on its loading policy. If the page fragment's loading policy is set to **onPage**, then this tag will render its content. If the page fragment's loading policy is set to **lazy**, however, then the content is replaced by a placeholder HTML fragment.

Syntax

```
<wf-core:renderFragment
  element="...">
  ...
</wf-core:renderFragment>
```

Attributes

element, mandatory

The `neo.xredsys.presentation.PresentationElement` for which the page fragment is to be loaded based on the loading policy.

2.26 wf-core:evalField

This tag takes the content-type field data as input and searches for bean properties.

Syntax

```
<wf-core:evalField
  id="..."
  inputFieldValue="...">
  ...
</wf-core:evalField>
```

Attributes

id, mandatory

This is the object where result be will placed.

inputFieldValue, mandatory

This is the input field data of content-type as String. The tag will search for this property in the bean.

2.27 wf-core:getCaptchaCustomHTML

This generates custom html code for captcha.

Syntax

```
<wf-core:getCaptchaCustomHTML>
  ...
</wf-core:getCaptchaCustomHTML>
```

2.28 wf-core:getConfigSectionPool

This tag resolves the **PresentationPool** for the current template.

Syntax

```
<wf-core:getConfigSectionPool>
  ...
</wf-core:getConfigSectionPool>
```

2.29 wf-core:getFieldValueByKey

This tag returns the value of field determined by key.

Syntax

```
<wf-core:getFieldValueByKey
  articleContent="..."?
  articleId="..."?
  articleSummary="..."?
  key="..."
  var="...">
  ...
</wf-core:getFieldValueByKey>
```

Attributes**var, mandatory, no runtime expressions**

Result is returned via this attribute.

articleSummary

If this attribute is present, the field value will be retrieved from the summary.

articleContent

If **articleSummary** attribute is not present and this attribute is present, then the field value will be retrieved from the article.

articleId

If **articleSummary** or **articleContent** attribute is absent, and this attribute is present, the field value will be retrieved from the article that has this id.

key, mandatory

The key for which the value is to be retrieved.

2.30 wf-core:getTagCloud

Generates tag cloud.

Syntax

```
<wf-core:getTagCloud
  id="..."
  max="..."?
  minimumFrequency="..."?
  sortCriteria="..."
  sortOrder="..."?
  tagStructureName="...">
  ...
</wf-core:getTagCloud>
```

Attributes

id, mandatory

Result is returned via this attribute.

tagStructureName, mandatory

The name of the tag structure

sortCriteria, mandatory

Sort criteria for the tag cloud, allowed values are **alphabetical**, **ranking**, and **random**.

sortOrder

Sort order for the tag cloud, allowed values are **ascending** and **descending**.

max

Maximum number of tags to be included.

minimumFrequency

Minimum frequency count for each tag.

2.31 wf-core:getVideoPreviewImageUrl

Gets the video preview image URL for a video article.

Syntax

```
<wf-core:getVideoPreviewImageUrl
  articleId="..."
```

```
imageVersion="..."?  
relationTypeName="..."  
var="...">  
...  
</wf-core:getVideoPreviewImageUrl>
```

Attributes

var, mandatory, no runtime expressions

Result is returned via this attribute.

articleId, mandatory

The id of the video article.

relationTypeName, mandatory

The **relation-type** name of the preview image.

imageVersion

The image version for which the URL will be retrieved.

2.32 wf-core:getYouTubeVideoUrl

This tag retrieves the URL of a Youtube video.

Syntax

```
<wf-core:getYouTubeVideoUrl  
embeddedUrl="..."  
var="..."  
videoCode="...">  
...  
</wf-core:getYouTubeVideoUrl>
```

Attributes

var, mandatory, no runtime expressions

Result is returned via this attribute.

videoCode, mandatory

The code of the Youtube video.

embeddedUrl, mandatory

If this attribute is set, the embedded video URL will be retrieved.

2.33 wf-core:include

Renders a widget.

Syntax

```
<wf-core:include  
elementWidth="..."?  
widgetElement="..."?  
widgetName="..."?>  
...  
</wf-core:include>
```

```
</wf-core:include>
```

Attributes

widgetName, no runtime expressions

Name of the widget

widgetElement, no runtime expressions

Widget presentation element

elementWidth, no runtime expressions

Width of widget container

2.34 wf-core:includeESI

Renders a template inside an ESI block.

Syntax

```
<wf-core:includeESI
  templateFile="...">
  ...
</wf-core:includeESI>
```

Attributes

templateFile, mandatory, no runtime expressions

Template file to include.

2.35 wf-core:isVideoArticle

Determines if a given article is a video article or not.

Syntax

```
<wf-core:isVideoArticle
  articleContent="..."?
  articleId="..."?
  articleSummary="..."?
  var="...">
  ...
</wf-core:isVideoArticle>
```

Attributes

var, mandatory, no runtime expressions

Result is returned via is this attribute.

articleSummary

An article summary to check if it is a video article.

articleContent

An article to check if it is a video article.

articleId

The id of an article to check if it is a video article.

2.36 wf-core:notPresent

Syntax

```
<wf-core:notPresent  
  name="...">  
  ...  
</wf-core:notPresent>
```

Attributes

name, no runtime expressions

The name of the widget to look for.

2.37 wf-core:present

Determines if a widget is defined in a publication or not.

Syntax

```
<wf-core:present  
  name="...">  
  ...  
</wf-core:present>
```

Attributes

name, no runtime expressions

The name of the widget to look for.

2.38 wf-core:renderContext

Renders the template of the root group in the current config pool.

Syntax

```
<wf-core:renderContext>  
  ...  
</wf-core:renderContext>
```

2.39 wf-core:renderItemsInArea

Renders the templates of the items in a given area.

Syntax

```
<wf-core:renderItemsInArea
  areaName="..."
  elementwidth="..."?>
  ...
</wf-core:renderItemsInArea>
```

Attributes

areaName, mandatory

The name of the area for which the items are to be rendered.

elementwidth

The width of the element container.

2.40 wf-core:showItems

Renders the templates of a list of items found in the request scope.

Syntax

```
<wf-core:showItems
  level="..."?>
  ...
</wf-core:showItems>
```

Attributes

level

Defines the nesting limit for the items to be rendered.

2.41 wf-core:view

Renders a view of a widget.

Syntax

```
<wf-core:view
  addWrapper="..."?
  flush="..."?
  view="..."?
  widget="..."?
  widgetName="..."?>
  ...
</wf-core:view>
```

Attributes

widget

The widget article for which the view is to be rendered.

widgetName

The name of the widget for which the view is to be rendered.

view

The name of the view to render.

flush

If this attribute is set, the buffer will be flushed.

addWrapper

If this attribute is set to false, wrapper will not be added while rendering view of a widget. The default value is true

2.42 wf-core:widgetAjaxMarkup

This tag prints the link containing URL for ajax loading of the widget.

Syntax

```
<wf-core:widgetAjaxMarkup>
  ...
</wf-core:widgetAjaxMarkup>
```

2.43 wf-core:pageFragment

Generates page holder fragments for lazily loaded items.

Syntax

```
<wf-core:pageFragment
  element="...">
  ...
</wf-core:pageFragment>
```

Attributes**element, mandatory**

A lazily loaded element for which the page holder fragment will be generated.

2.44 wf-core:iteratorView

This tag iterates through a map of views to select a particular one for each item in a list of articles.

Syntax

```
<wf-core:iteratorView
  resultItems="..."
  selectableView="...">
  ...
</wf-core:iteratorView>
```

Attributes

selectableView, mandatory

The map of selectable views.

resultItems, mandatory

A list of teaser items, a view is to be selected for each item in the list.

2.45 wf-core:renderGridItem

Render the template of the provided item using the Teaser grid widget.

Syntax

```
<wf-core:renderGridItem
  item="...">
  ...
</wf-core:renderGridItem>
```

Attributes

item, mandatory

The item to be rendered.

2.46 wf-core:generateGroupId

Generate a group ID for rendering template

Syntax

```
<wf-core:generateGroupId
  counter="..."?
  fragmentToken="..."?
  groupName="..."?
  prefix="..."?
  scope="..."?
  var="..."/>
```

Attributes

groupName

Group name the id should be unique for

var, mandatory

Result will be returned in that variable

scope

Specify in which scope result will be set

prefix

Prefix to be added to the generated group id

fragmentToken

Fragmentation token of the group in subject

counter

Current counter of this group's occurrence in rendering template

2.47 wf-core:renderGroup

This tag looks for the group processors of the group provided in the `com.escenic.framework.descriptor.GroupDescriptorFactory` with the help of the `com.escenic.framework.descriptor.GroupDescriptor` to carry out any processing required by the group, and renders the template of the group.

Syntax

```
<wf-core:renderGroup
  group="..."?
  templatePath="..."?>
  ...
</wf-core:renderGroup>
```

Attributes

group

The group for which to render the template.

templatePath

An optional parameter to specify the path to the group template. If not specified, the template will be located from the path `/template/framework/group/<group-name>.jsp`.

2.48 wf-core:profileContent

This tag finds the profile content of a person

Syntax

```
<wf-core:profileContent
  personId="..."
  scope="..."?
  var="..." />
```

Attributes

var, mandatory, no runtime expressions

personId, mandatory

The ID of the person

scope

Specify in which scope result will be set

2.49 wf-core:profilePhoto

This tag finds the profile photo of a person. Returns null if profile or profile photo cannot be found.

Syntax

```
<wf-core:profilePhoto  
  personId="..."  
  scope="..."?  
  var="..."/>
```

Attributes

var, mandatory, no runtime expressions

personId, mandatory

The ID of the person

scope

Specify in which scope result will be set

3 wf-functions Tag Library

This library contains widget framework functions tags.

3.1 wf-functions:getRssSource

Syntax

```
| wf-functions:getRssSource(java.lang.String)
```

Return type: `java.lang.String`

Method executed

[com.escenic.framework.tag.util.EWFUtil:getRssSource\(java.lang.String\)](#)

3.2 wf-functions:getCaptchaHTML

Syntax

```
| wf-functions:getCaptchaHTML()
```

Return type: `java.lang.String`

Method executed

[com.escenic.framework.tag.util.EWFUtil:getCaptchaHTML\(\)](#)

3.3 wf-functions:convertFahrenheitToCelsius

Syntax

```
| wf-functions:convertFahrenheitToCelsius(int, boolean)
```

Return type: `java.lang.Number`

Method executed

[com.escenic.framework.tag.util.EWFUtil:convertFahrenheitToCelsius\(int, boolean\)](#)

3.4 wf-functions:getRelativeDate

Syntax

```
wf-functions:getRelativeDate(java.util.Date, long)
```

Return type: `java.util.Date`

Method executed

[com.escenic.framework.tag.util.EWFUtil:getRelativeDate\(java.util.Date, long\)](#)

3.5 wf-functions:getDateObject

Syntax

```
wf-functions:getDateObject(java.lang.String)
```

Return type: `java.util.Date`

Method executed

[com.escenic.framework.tag.util.EWFUtil:getDateObject\(java.lang.String\)](#)

3.6 wf-functions:getDateString

Syntax

```
wf-functions:getDateString(java.lang.String)
```

Return type: `java.lang.String`

Method executed

[com.escenic.framework.tag.util.EWFUtil:getDateString\(java.lang.String\)](#)

3.7 wf-functions:getTermFromTagIdentifier

Syntax

```
wf-functions:getTermFromTagIdentifier(java.lang.String, java.lang.String)
```

Return type: `java.lang.String`

Method executed

[com.escenic.framework.tag.util.EWFUtil:getTermFromTagIdentifier\(java.lang.String, java.lang.String\)](#)

3.8 wf-functions:getImageDimension

Syntax

```
wf-functions:getImageDimension(java.lang.String)
```

Return type: `java.util.Map`

Method executed

[com.escenic.framework.tag.util.EWFUtil:getImageDimension\(java.lang.String\)](#)

3.9 wf-functions:parseFeedForImage

Syntax

```
wf-functions:parseFeedForImage(java.lang.String)
```

Return type: `java.lang.String`

Method executed

[com.escenic.framework.tag.util.EWFUtil:parseFeedForImage\(java.lang.String\)](#)

3.10 wf-functions:getConfigSectionByInheritance

Syntax

```
wf-functions:getConfigSectionByInheritance(neo.xreditsys.api.Section, java.lang.String,  
int)
```

Return type: `neo.xreditsys.api.Section`

Method executed

[com.escenic.framework.tag.util.EWFUtil:getConfigSectionByInheritance\(neo.xreditsys.api.Section, java.lang.String, int\)](#)

3.11 wf-functions:isContainedInList

Syntax

```
wf-functions:isContainedInList(java.util.List, java.lang.Object)
```

Return type: `boolean`

Method executed

[com.escenic.framework.tag.util.EWFUtil:isContainedInList\(java.util.List, java.lang.Object\)](#)

3.12 wf-functions:getImageRepresentation

Syntax

```
wf-  
functions:getImageRepresentation(java.lang.String, neo.xreditsys.api.Publication, java.lang.String)
```

Return type: `java.lang.String`

Method executed

[com.escenic.framework.tag.util.EWFUtil:
getImageRepresentation\(java.lang.String, neo.xreditsys.api.Publication, java.lang.String\)](#)

3.13 wf-functions:getSuitableRepresentation

Syntax

```
wf-  
functions:getSuitableRepresentation(java.lang.String, neo.xreditsys.api.Publication, java.lang.String, java.lang.String)
```

Return type: `java.lang.String`

Method executed

[com.escenic.framework.tag.util.EWFUtil:
getSuitableRepresentation\(java.lang.String, neo.xreditsys.api.Publication, java.lang.String, java.lang.String\)](#)

3.14 wf-functions:isFieldAccessible

Determines whether a premium field is accessible or not.

Do not use this function. Premium content is not currently supported by the Widget Framework, and use of this function can therefore cause errors. Moreover, the function is deprecated and will be removed in a future release of the Widget Framework.

Syntax

```
wf-functions:isFieldAccessible(int, java.lang.String, java.lang.String)
```

Return type: `boolean`

Method executed

[com.escenic.framework.tag.util.EWFUtil:isFieldAccessible\(int, java.lang.String, java.lang.String\)](#)

3.15 wf-functions:getDateDifference

Syntax

```
| wf-functions:getDateDifference(java.util.Date,java.util.Date)
```

Return type: `java.lang.String`

Method executed

[com.escenic.framework.tag.util.EWFUtil:getDateDifference\(java.util.Date,java.util.Date\)](#)

3.16 wf-functions:handleLineBreaks

Syntax

```
| wf-functions:handleLineBreaks(java.lang.String)
```

Return type: `java.lang.String`

Method executed

[com.escenic.framework.tag.util.EWFUtil:handleLineBreaks\(java.lang.String\)](#)

3.17 wf-functions:isValidName

Syntax

```
| wf-functions:isValidName(java.lang.String)
```

Return type: `boolean`

Method executed

[com.escenic.framework.tag.util.EWFUtil:isValidName\(java.lang.String\)](#)

3.18 wf-functions:isNumeric

Syntax

```
| wf-functions:isNumeric(java.lang.String)
```

Return type: `boolean`

Method executed

[com.escenic.framework.tag.util.EWFUtil:isNumeric\(java.lang.String\)](#)

3.19 wf-functions:filterArticleList

Syntax

```
wf-functions:filterArticleList(java.util.Collection,java.lang.String,java.lang.String)
```

Return type: `java.util.List`

Method executed

[com.escenic.framework.tag.util.EWFUtil:filterArticleList\(java.util.Collection,java.lang.String,java.lang.String\)](#)

3.20 wf-functions:getSubList

Syntax

```
wf-functions:getSubList(java.util.List,java.lang.String,java.lang.String)
```

Return type: `java.util.List`

Method executed

[com.escenic.framework.tag.util.EWFUtil:getSubList\(java.util.List,java.lang.String,java.lang.String\)](#)

3.21 wf-functions:getTopicURL

Syntax

```
wf-functions:getTopicURL(java.lang.String,java.lang.String,java.lang.String)
```

Return type: `java.lang.String`

Method executed

[com.escenic.framework.tag.util.EWFUtil:getTopicURL\(java.lang.String,java.lang.String,java.lang.String\)](#)

3.22 wf-functions:matchFileExtension

Syntax

```
wf-functions:matchFileExtension(neo.xredsys.presentation.PresentationArticle,java.lang.String)
```

Return type: `boolean`

Method executed

[com.escenic.framework.tag.util.EWFUtil:matchFileExtension\(neo.xredsys.presentation.Pres](#)

3.23 wf-functions:getDescBreakpointsAsList

Syntax

```
wf-functions:getDescBreakpointsAsList (java.lang.String[])
```

Return type: `java.util.ArrayList`

Method executed

[com.escenic.framework.tag.util.EWFUtil:getDescBreakpointsAsList \(java.lang.String\[\]\)](#)

3.24 wf-functions:getDefaultAspectVariant

Syntax

```
wf-functions:getDefaultAspectVariant ()
```

Return type: `java.lang.String`

Method executed

[com.escenic.framework.tag.util.EWFUtil:getDefaultAspectVariant \(\)](#)

3.25 wf-functions:getArticleSummary

Syntax

```
wf-functions:getArticleSummary (int)
```

Return type: `java.lang.String`

Method executed

[com.escenic.framework.tag.util.EWFUtil:getArticleSummary \(int\)](#)

3.26 wf-functions:findGroupByName

Syntax

```
wf-functions:findGroupByName (neo.xredsys.presentation.PresentationPool,  
java.lang.String, int)
```

Return type: `neo.xreditsys.presentation.PresentationElement`

Method executed

[com.escenic.framework.tag.util.EWFUtil:
findGroupByName \(neo.xreditsys.presentation.PresentationPool,
java.lang.String,int\)](#)

3.27 wf-functions:generatePageURLWithoutQueryString

Syntax

```
wf-functions:generatePageURLWithoutQueryString(javax.servlet.http.HttpServletRequest)
```

Return type: `java.lang.String`

Method executed

[com.escenic.framework.tag.util.EWFUtil:generatePageURLWithoutQueryString \(javax.servlet.](#)

3.28 wf-functions:isImageContent

Syntax

```
wf-functions:isImageContent (neo.xreditsys.presentation.PresentationArticle)
```

Return type: `boolean`

Method executed

[com.escenic.framework.tag.util.EWFUtil:isImageContent \(neo.xreditsys.presentation.Presenta](#)

3.29 wf-functions:getRelativeTime

Syntax

```
wf-functions:getRelativeTime (java.util.Date)
```

Return type: `java.util.Map`

Method executed

[com.escenic.framework.tag.util.EWFUtil:getRelativeTime \(java.util.Date\)](#)

3.30 wf-functions:getClientTypeDefinition

Syntax

```
wf-functions:getClientTypeDefinition(neo.xreditsys.api.Section)
```

Return type: `com.escenic.common.util.JSONMap`

Method executed

[com.escenic.framework.tag.util.EWFUtil:getClientTypeDefinition\(neo.xreditsys.api.Section\)](#)

3.31 wf-functions:getImageSizes

Syntax

```
wf-functions:getImageSizes(int)
```

Return type: `java.util.Set`

Method executed

[com.escenic.framework.tag.util.EWFUtil:getImageSizes\(int\)](#)

3.32 wf-functions:tagPageUrlWithSection

Syntax

```
wf-functions:tagPageUrlWithSection(neo.xreditsys.api.Section,  
neo.xreditsys.presentation.PresentationTag)
```

Return type: `String`

Method executed

[com.escenic.framework.tag.util.EWFUtil:getTagPageUrl\(neo.xreditsys.api.Section,
neo.xreditsys.presentation.PresentationTag\)](#)

3.33 wf-functions:tagPageUrlWithPublication

Syntax

```
wf-functions:tagPageUrlWithPublication(neo.xreditsys.api.Publication,  
neo.xreditsys.presentation.PresentationTag)
```

Return type: String

Method executed

[com.escenic.framework.tag.util.EWFUtil:getTagPageUrl\(neo.xredsys.api.Publication, neo.xredsys.presentation.PresentationTag\)](#)

3.34 wf-functions:taggingEntity

Syntax

```
wf-functions:taggingEntity(neo.xredsys.presentation.PresentationTag)
```

Return type: String

Method executed

[com.escenic.framework.tag.util.EWFUtil:getTaggingEntity\(neo.xredsys.presentation.PresentationTag\)](#)

3.35 wf-functions:tagUriHashCode

Syntax

```
wf-functions:tagUriHashCode(neo.xredsys.presentation.PresentationTag)
```

Return type: int

Method executed

[com.escenic.framework.tag.util.EWFUtil:getTagUriHashCode\(neo.xredsys.presentation.PresentationTag\)](#)

3.36 wf-functions:createSectionIDPath

Syntax

```
wf-functions:createSectionIDPath(neo.xredsys.api.Section)
```

Return type: String

Method executed

[com.escenic.framework.tag.util.EWFUtil:createSectionIDPath\(neo.xredsys.api.Section\)](#)

3.37 wf-functions:hasProperty

Syntax

```
wf-functions:hasProperty (neo.xreditsys.presentation.PresentationArticle,  
    java.lang.String)
```

Return type: `boolean`

Method executed

[com.escenic.framework.tag.util.EWFUtil:hasProperty \(neo.xreditsys.presentation.PresentationArticle, java.lang.String\)](#)

3.38 wf-functions:hasLazyFragment

Syntax

```
wf-functions:hasLazyFragment (neo.xreditsys.presentation.PresentationElement)
```

Return type: `boolean`

Method executed

[com.escenic.framework.tag.util.EWFUtil:hasLazyFragment \(neo.xreditsys.presentation.PresentationElement\)](#)

3.39 wf-functions:getRelatedElements

Syntax

```
wf-  
functions:getRelatedElements (neo.xreditsys.presentation.PresentationArticle, java.lang.String, java.lang.  
    java.lang.Integer)
```

Return type: `java.util.List`

Method executed

[com.escenic.framework.tag.util.EWFUtil:getRelatedElements \(neo.xreditsys.presentation.PresentationArticle, java.lang.String, java.lang.Integer\)](#)

3.40 wf-functions:isMediaContent

Syntax

```
wf-functions:isMediaContent (neo.xreditsys.presentation.PresentationArticle)
```


Return type: `boolean`

Method executed

[com.escenic.framework.tag.util.EWFUtil:isMediaContent\(neo.xredsys.presentation.Presentation\)](#)

3.41 wf-functions:getSection

Syntax

```
wf-functions:getSection(java.lang.String, java.lang.String)
```

Return type: `neo.xredsys.api.Section`

Method executed

[com.escenic.framework.tag.util.EWFUtil:getSection\(java.lang.String, java.lang.String\)](#)

3.42 wf-functions:getMasterWidgetNestingLimit

Syntax

```
wf-functions:getMasterWidgetNestingLimit(neo.xredsys.api.Section)
```

Return type: `int`

Method executed

[com.escenic.framework.tag.util.EWFUtil:getMasterWidgetNestingLimit\(neo.xredsys.api.Section\)](#)

3.43 wf-functions:isValidMasterWidgetNestingLimit

Syntax

```
wf-functions:isValidMasterWidgetNestingLimit(java.lang.String, java.lang.String)
```

Return type: `boolean`

Method executed

[com.escenic.framework.tag.util.EWFUtil:isValidMasterWidgetNestingLimit\(java.lang.String, java.lang.String\)](#)

3.44 wf-functions:decreaseMasterWidgetNestingLevel

Syntax

```
| wf-functions:decreaseMasterWidgetNestingLevel (java.lang.String)
```

Return type: `int`

Method executed

[com.escenic.framework.tag.util.EWFUtil:decreaseMasterWidgetNestingLevel \(java.lang.String\)](#)

3.45 wf-functions:increaseMasterWidgetNestingLevel

Syntax

```
| wf-functions:increaseMasterWidgetNestingLevel (java.lang.String)
```

Return type: `int`

Method executed

[com.escenic.framework.tag.util.EWFUtil:increaseMasterWidgetNestingLevel \(java.lang.String\)](#)

3.46 wf-functions:getMediaInfoLink

Syntax

```
| wf-functions:getMediaInfoLink (neo.xreditsys.presentation.PresentationArticle)
```

Return type: `java.lang.String`

Method executed

[com.escenic.framework.tag.util.EWFUtil:getMediaInfoLink \(neo.xreditsys.presentation.PresentationArticle\)](#)

3.47 wf-functions:isEmbeddedContent

Syntax

```
| wf-functions:isEmbeddedContent (neo.xreditsys.presentation.PresentationArticle)
```

Return type: `boolean`

Method executed

[com.escenic.framework.tag.util.EWFUtil:isEmbeddedContent \(neo.xreditsys.presentation.PresentationArticle\)](#)

3.48 wf-functions:resolveAMPLink

Syntax

```
| wf-functions:resolveAMPLink(java.lang.String, java.lang.String)
```

Return type: `String`

Method executed

[com.escenic.framework.tag.util.EWFUtil:resolveAMPLink\(java.lang.String, java.lang.String\)](#)

3.49 wf-functions:getAMPGATrackerConfig

Syntax

```
| wf-functions:getAMPGATrackerConfig(neo.xredsys.api.Section)
```

Return type: `com.escenic.common.util.JSONMap`

Method executed

[com.escenic.framework.tag.util.EWFUtil:getAMPGATrackerConfig\(neo.xredsys.api.Section\)](#)

4 wf-util Tag Library

This library contains widget framework utility tags.

4.1 wf-util:wrap

Syntax

```
<wf-util:wrap
  customAttributes="..."?
  htmlTag="..."
  if="..."?
  styleClass="..."?
  styleId="..."?
  ...
</wf-util:wrap>
```

Attributes

htmlTag, mandatory

The HTML tag to wrap with.

styleId

The value of "Id" attribute

styleClass

The value of 'class' attribute

if

Condition to check whether to add the wrapper element at all.

customAttributes

The map of custom attributes

4.2 wf-util:link

Syntax

```
<wf-util:link
  click="..."?
  loadingDevices="..."?
  showLink="..."
  styleClass="..."?
  url="..."?>
  ...
</wf-util:link>
```

Attributes

url

The url where the page should redirect to.

click

The onclick event for the link.

showLink, mandatory

Whether to render the link element or not.

styleClass

The value of 'class' attribute

loadingDevices

The devices on which inline loading will be enabled.

4.3 wf-util:use

Syntax

```
<wf-util:use  
  value="..."  
  var="...">  
  ...  
</wf-util:use>
```

Attributes

var, mandatory, no runtime expressions

The name of the attribute

value, mandatory

The value of the attribute

4.4 wf-util:sanitize

Syntax

```
<wf-util:sanitize  
  markup="..." />
```

Attributes

markup, mandatory

The HTML code to sanitize

4.5 wf-util:embedFileContent

Syntax

```
<wf-util:embedFileContent  
  filePath="..."?  
  href="..."?>  
  ...  
</wf-util:embedFileContent>
```

Attributes

href

The url of file to be embedded.

filePath

The relative path of the file within the server to be embedded.

5 Deprecated Tags

Following is a list of tags marked as deprecated.

- `getPaginationRange.tag`, use Search Component widget.
- `getSearchResult.tag`, use Search Component widget.
- `getSearchResultLink.tag`, use Search Component widget.
- `getTagSearchResult.tag`, use Teaser widget.
- `getTeaserImageMap.tag`, use Teaser widget.
- `getTeaserImageMapList.tag`, use Teaser widget.
- `iteratorView.tag`, use `wf-core:iterateViews` tag.
- `pageFragment.tag`, use `wf-core:renderFragment` tag.

It is highly recommended to replace these tags with the suggested functionality. In some future version of Widget Framework, these tags might be removed.