ColdFusion Tag Specifications

Specification changes

If you edit this page, please upload a new PDF-export of these specifications to the repository folder:
https://source.rhinofly.net:8443/svn/Development/ColdFusion/Projects/DocumentationGenerator/trunk/site/data/apiDoc
as readme.pdf

Tag specifications

Metadata

In ColdFusion, there are two ways to generate metadata for the purpose of reference documentation

- set attributes native to ColdFusion
- include tags in the hint of a component, interface, function, or property

The hints in CFScript are placed directly in front of the definition, in between the comment symbols "/**" and "*/". Each new line of comment can start with a "*", but it's not necessary. In principle, a tag "@foo" within such a hint must be placed at the beginning of a line, and is converted to an attribute with the name "foo" and the value equal to whatever follows the tag in the form of a string. However, multiple uses of the same tag results in a single value corresponding to the last usage of the tag. Any comments not preceded by a tag are put into the "hint" attribute, unless the "@hint" tag is explicitly used.
Do not follow the tag by a tab character! Use a space character in stead. For some reason I cannot fathom, ColdFusion considers tab characters to be a part of the tag...
Tags that have the same name as an argument of the function following the comment are special in the sense that they set the "hint" attribute for the corresponding argument under "parameters". No new attribute of the function is created.
Tags that are placed in the hints or descriptions of *tag-based* ColdFusion code are not automatically converted to metadata.

Tags

The tags in this specification can be placed in the code regardless of the coding method that's used. If they don't properly end up in the metadata, they will be parsed anyway.

@author

`@author name`
Provides the author name for the component, or interface. Can also be used for properties and functions, but preferably not. Use version control to keep track of added functionalities.

@date

`@date date`
Provides the date on which the component, or interface was written. Can also be used for properties and functions, but preferably not. Use version control to keep track of added functionalities.

@default

`@default value`
Tag-based CF: Please use the DEFAULT attribute in the CFPROPERTY tag.
CFScript: Puts the value into the DEFAULT attribute corresponding to the property. It both sets the default value and includes it in the documentation.

@argument name

`@argument name hint`
Tag-based CF: Please use the HINT attribute in the CFARGUMENT tag.
CFScript: Puts the hint text into the HINT attribute corresponding to the name *ARGUMENT NAME* of the PARAMETERS metadata section of the associated function.
Give the description for this function argument.

@return

`@return hint`
Give the description for the return value of a function.

@throws

`@throws exceptionType hint`
Adds the exception to the list in the documentation page of exceptions that can be thrown by a function, together with its description.
CFScript: In CFBuilder, Adobe only allows you to use each distinct tag once. Therefore, in order to establish multiple instances, put additional exception types and descriptions on consecutive lines following the initial tag line. Each line will be treated as starting with the @throws tag, until another tag is used.

@hint

`@hint hint`
Tag-based CF: All comments are already enclosed in the HINT attribute.
CFScript: Puts the hint text into the HINT attribute of the component, interface, function, or property, instead of (the untagged part of) the entire comment.

@internal

`@internal hint`
Provides hint text that is not included in the documentation. Also removes subsequent untagged text from the documentation.

@private

`@private`
Excludes the the component, interface, function, or property from documentation entirely.

@inheritDoc

`@inheritDoc`
Copies all descriptions and comments of an overridden function from the ancestor it extends or implements. Any tags used in addition to this one will be ignored.

@see

`@see link`
Adds the link to the list in the documentation page of related documentation. The expression *link* can be a URL, a component/interface from the same library, or a Java class. Use a "#" followed by an ID to link to a certain section of a page.
CFScript: In CFBuilder, Adobe only allows you to use each distinct tag once. Therefore, in order to establish multiple instances, put additional links on consecutive lines following the initial tag line. Each line will be treated as starting with the @see tag, until another tag is used.

{@link}

`{@link} link`
Creates a hyperlink in the documentation page. The expression for *link* has the same format as the usage of the "@see" tag. However, the "{@link}" tag doesn't have to be placed at the beginning of a comment line. Essentially, it replaces the HTML hyperlink tag.