Modes in Content Editor and Article Editor - SysAdmin

Quick overview of the custom-content-handler elements

Attributes

Handler "asset"

Possible values:

```
<custom-content-handler element-name = "image" handler-id = "asset" > <config mode="
id|xlink+rest|xlink+rest+versioned " [link-role="..."]/> </custom-content-handler>
```

Example of default mode "ID" – View of Content-Editor/Article-Editor

The modes available are:

- "**id**" (which is the default mode):

```
<image asset-id = "censhare:/service/assets/asset/id/446750/version/1/storage/master/file" />
```

- "**xlink+rest**"

```
<image xlink:href = "censhare:///service/assets/asset/id/446750" xmlns:xlink = "http://www.w3.org/1999/xlink" />
```

- "**xlink+rest+versioned**"

```
<image xlink:href = "censhare:///service/assets/asset/id/446750/version/1" xmlns:xlink = "http://www.w3.org
/1999/xlink" />
```

Optional : config attribute for modes "xlink" modes: "**link-role**":

```
<config mode = "xlink+rest" link-role = "censhare:///service/masterdata/asset_rel_typedef;key=actual." /> <imag
e xlink:href = "censhare:///service/assets/asset/id/446750" xlink:role = "censhare:///service/masterdata
/asset_rel_typedef;key=actual." xmlns:xlink = "http://www.w3.org/1999/xlink" />
```

Optional : Changing the name of the link attribute (default "asset-id") for mode "**id**":

```
<custom-content-handler element-name = "image" handler-id = "asset" asset-ref-attribute = "link" > <config mode
= "id" /> </custom-content-handler>
```

or

```
<custom-content-handler element-name = "image" handler-id = "asset" > <config mode = "id" asset-ref-attribute =
"link" /> </custom-content-handler> <image link = "censhare:/service/assets/asset/id/446750/version/1/storage
/master/file" />
```

Important note: You cannot change the name of an xlink reference attribute. It's always "xlink:href".

Handler "asset-link"

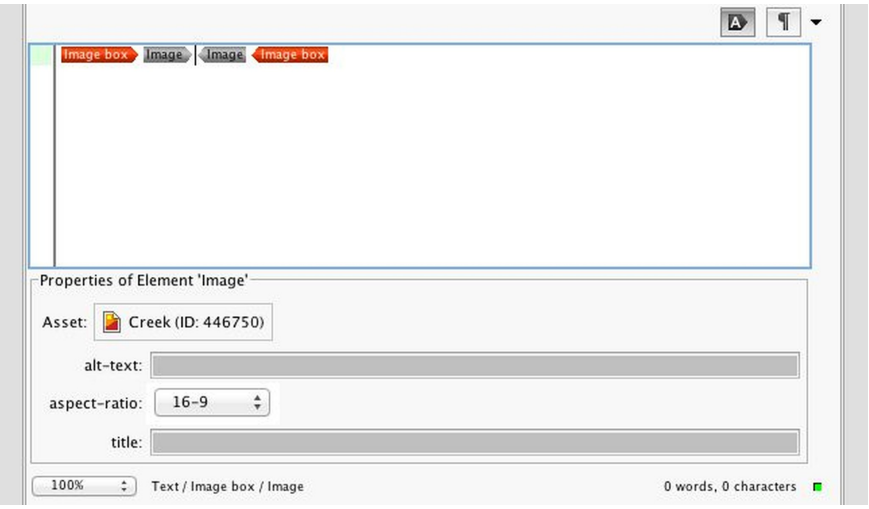
This handler seems to be identical to a handler with id "asset" but with one exception: The link element itself ("image" tag in below's example) is shown in the editor around the preview of the placed asset.

Example with handler "asset-link"

The link structure inside the XML content generated by this handler is identical to the handler "asset". This is also true for the available configuration parameters. However, it's recommended not to use this handler for content in content placements where it's normally undesired to see the xi:include element before and after the included text.

Handler "asset-link-nopreview"

Same as handler "asset", but does not display a preview of the placed asset inline in the content. Only the element containing the link ("image" in below's example) is visualized. The placed asset is visible by icon, name and id in the properties panel of the corresponding element.



Example: "asset-link-nopreview"

Content-in-content structures

The CustomContentHandler for element "**xi:include**" is responsible how to visualize a content in content placement.

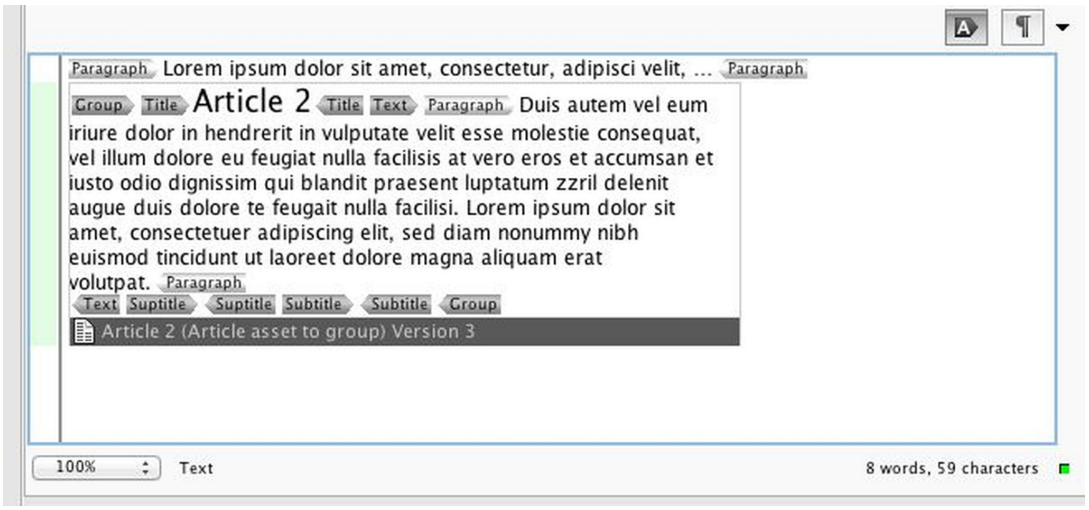
Example

```
<custom-content-handler element-name = "xi:include" handler-id = "asset" filter-key = "asset" > <config asset-ref-attribute = "href" /> </custom-content-handler>
```

This will create this link element:

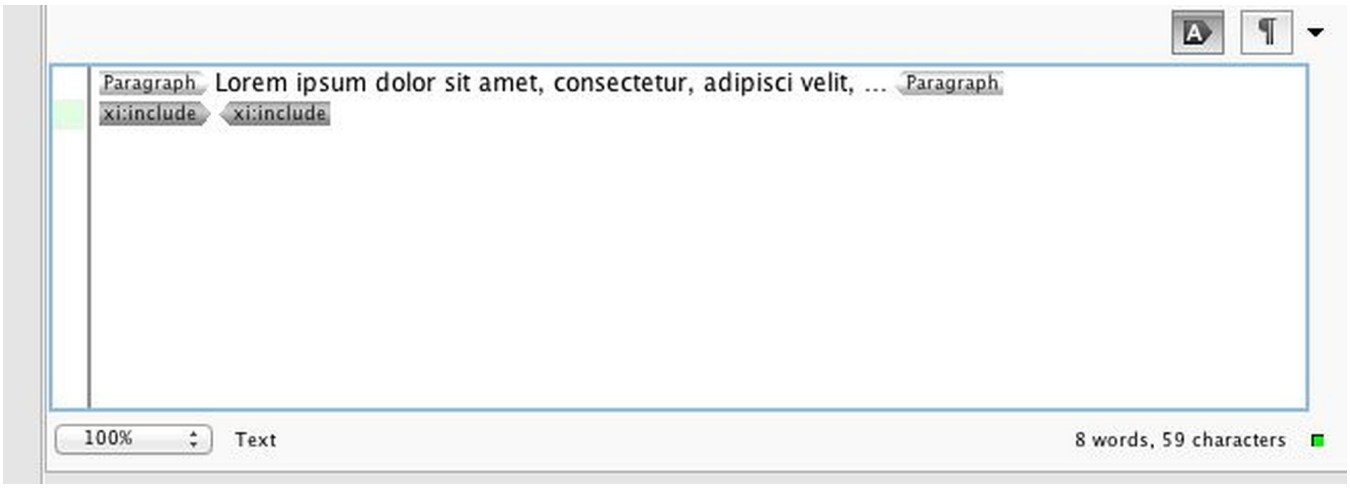
```
< xi:include xmlns:xi = "http://www.w3.org/2001/XInclude" href = "censhare:/service/assets/asset/id/675233/version/3/transform;key=transform:article-asset-to-group;format=xml" element-name = "group" />
```

The content is visualized like the following:



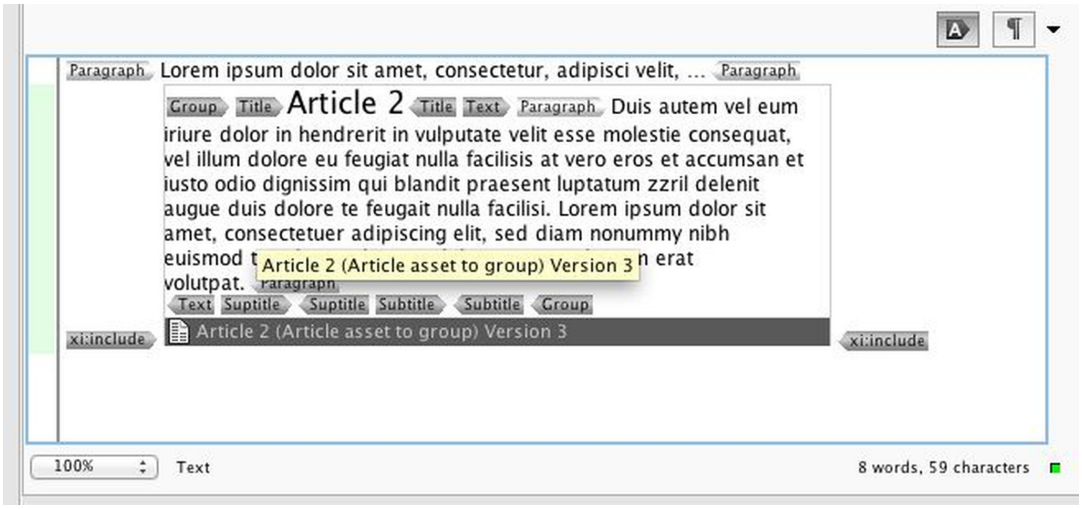
Content-in-Content

If a handler definition for that element is missing, the placed content is displayed solely by the ***xi:include*** tag and no preview of the placed content is visible:



Missing handler definition: Content-in-Content

Note: The Attribute asset-ref-attribute="href" must be present on the config element and it must be set to value "href". Otherwise the content-in-content placement is not correctly visualized.
It is possible to use the handler with id "asset-link" instead of handler "asset". The only difference is that the link tag "xi:include" is shown in the editor around the placed content:



xi:include-Tag

However, it is recommended not to use this handler for content in content structures.
AssetRefWidget and StructureEditorWidget now both again write the ID of an Asset correctly when operating in mode "ID".
In Clients before 4.6.16 the StructureEditor did write an REST-URL instead of the Asset-ID. This has been fixed in Client 4.6.17, 4.7.10 and above.

Structure editor: Snippets inserted automatically via drag and drop of an asset (import-def in input formular) cannot be used to produce versioned REST URLs (xlink references), only URLs without asset version are possible.
Now a new variable ***${asset-version*** } is available, which can be used inside the snippet as placeholder for the version of the dropped asset.
Both variables ***${asset-id}*** and ***${asset-version}*** can be used to construct valid REST URLs for xlink href attributes.

Unversioned

```
xlink:href="censhare:///service/assets/asset/id/${asset-id}"
```

Versioned

```
xlink:href="censhare:///service/assets/asset/id/${asset-id}/version/${asset-version}"
```

Example of a complete snippet

```
<snippet key = "image-box" menu-path = "/" scope = "text,group" title = "${image-box}" >

<image-box>

<image xlink:href = "censhare:///service/assets/asset/id/${asset-id}/version/${asset-version}"  xmlns:xlink = "
http://www.w3.org/1999/xlink"  xlink:role = "censhare:///service/masterdata/asset_rel_typedef;key=actual." />

<caption>

<paragraph>

<__select__/>

</paragraph>

</caption>

</image-box>

</snippet>
```

Available variables

- ***${asset-id}*** : ID of the dropped asset
- ***${asset-version}*** : Version of the dropped asset
- ***${url}: (Deprecated)*** Versioned REST URL to the master storage of the dropped asset
- ***${storage-uri}: (Deprecated)*** Same as ${url}
- ***${service-url} : (Deprecated)*** Same as ${url}