

Definition of Translate5 internal Tags

Translate5 internal used HTML Tags

The following Document describes how translate5 uses some defined HTML Tags internally to markup additional meta information to textual parts of a segment.

Terminology

Terminology is marked up by a ordinary HTML <div> Tag. The markup contains a defined set of attributes. The whole phrase which holds terminology is encapsulated by the tag.

177

If you built Apache from source, as described above, the below example will match your path for apxs, but if you installed Apache some other way, you'll need to adjust the path to apxs accordingly.

178

Note that some Apache license.

Terminology is rendered with different colors, depending on the type and "found in target" state. If hovered with the mouse the explanation of the term is shown.

Tag

One <div> Tag pair surround the text with terminology. The surrounded text is highlighted optically.

Attributes

All Attributes are mandatory, even if they are empty.

Attribute	Type	Description
class	String	Contains the following structure: <i>term</i> <i>TERMTYPE</i> <i>TRANSFOUND</i> where: "term" is a fixed string to identify the div as terminology <i>TERMTYPE</i> is one of the following strings: - supersededTerm - preferredTerm - admittedTerm - legalTerm - regulatedTerm - standardizedTerm - deprecatedTerm - stemmer to identify terms found with stemming - fuzzy to identify terms found with fuzzy in general - fuzzy-XXX to deliver the fuzzy value - TRANSFOUND is only used in the segment source. Two values are then allowed: "transFound" and "transNotFound". The flag marks if a translation to the term is found in one of the targets or not.
data-tbxid	String	ID of the term in the tbx-file
title	String	The textual description/explanation of the term. Can be empty if no description is given.

Examples

... text ... <div class="term supersededTerm" id="f_term_de-19" title="Bezeichnet einen Prozess oder Modul des Apache, welches genutzt wird, bevor der Prozess sich verzweigt. Englischer Begriff, sollte im Deutschen nicht verwendet werden.">Prefork</div> ... text ...

...<div class="term preferredTerm" data-tbxid="term_03_1_de_1_00005-4" title="">eine</div> ...

... <div class="term admittedTerm" data-tbxid="term_04_1_de_1_00007-5" title="">keiner</div> ...

... <div class="term admittedTerm" data-tbxid="term_04_1_de_1_00007-6" title="">eines</div> ...

... <div class="term preferredTerm" data-tbxid="term_03_1_de_1_00005-7" title="">Sein</div> ...

... <div class="term preferredTerm transFound" data-tbxid="term_11_1_en_1_00019-8" title="Apache is the standard webserver and is released as open source under the Apache license.">Apache</div> ...

... <div class="term supersededTerm" data-tbxid="f_term_de-19" title="Bezeichnet einen Prozess oder Modul des Apache, welches genutzt wird, bevor der Prozess sich verzweigt. Englischer Begriff, sollte im Deutschen nicht verwendet werden.">Prefork</div> ...

MQM Tags

MQM Tags are marking the textual quality of the desired parts of a segment. Therefore tags are used. Opening and closing tags are mapped semantically by the class and data-seq attribute and optically by the displayed image content. A correct order is ensured by the frontend, the correction of the tag nesting is planned but currently not implemented.



MQM Tags are visualized as an opening and closing tag pair, showing directly the MQM issue type number. When hovering with the mouse the MQM issue type name (Internationalization) and the Severity are shown. The optional comment is missing in the screenshot, but would be rendered below of the severity.

Tag

One single Tag before and one after the marked up text.

Attributes

All Attributes are mandatory, even if they are empty.

Attribute	Type	Description
class	String	Contains the following structure: <i>SEVERITY qmflag ownttip TYPE qmflag-N</i> where: - SEVERITY is one of the following strings to define the severity of the MQM issue. This list can be edited in the configuration of the translate5 instance! - critical - major - minor "qmflag" and "ownttip" are fixed strings to identify the as MQM Tag and enable the tooltip feature in the frontend. - TYPE is either "open" or "close" to define if its an opening or closing MQM Tag. "qmflag-N" is consists of the fixed string "qmflag-" and a numeric suffix represented here by the letter "N", which will be the integer ID of the selected MQM Type. The ID depends on the MQM Subset defined at the import time of the task..
data-seq	Integer	The data-seq is the unique ID of the created MQM Issue. An open / close tag pair has therefore the same data-seq value. The data-seq is only an integer after saving the segment to the DB. If the Frontend sends a new taig pair, data-seq contains a string which the backend replaces then with ID after saving.
data-comment	string	A free text provided by the user on MQM Tag creation. Can be empty.

Examples

```
contains
```

Content Tags

Content Tags are marking special content which should not be edited by the user, but can be moved in the segment or removed from it. Content Tags are represented by a whole structure of HTML Tags, as described below. In general Content Tags are either an opening and a closing tag pair or a standalone single tag. The HTML structure of the all three content tag types are the same, only the attribute values are differing.

151	The <code><1><2><3></code> Apache Documentation <code></1></code> is the most authoritative source of information on the Apache 2.x server.	<code><1></code> and <code></1></code> are a tag pair in short tag mode shown only with numbers. The users gets the tag content by hovering over the tag. <code><2></code> and <code><3></code> are independent single tags.
151	The <code>&raquo;<space></code> Apache Documentation <code></code> is the most authoritative source of information on the Apache 2.x server.	The same segment in full tag mode. Instead the short numbers of above the whole content of tag is shown. The <code><1></1></code> tag pair contains a masked HTML <code><a></code> tag pair. <code><2></code> masks a special HTML character and <code><3></code> a special space character.

Tag Structure

```
<div>
<span>SHORT_CONTENT</span>
<span>LONG_CONTENT</span>
</div>
```

This whole HTML structure represents one content tag.

LONG_CONTENT is the HTML encoded original content of the imported Content Tag.

SHORT_CONTENT is a HTML encoded numeric value representing the original Content Tag. The User sees the original Content by hovering the Tag with the mouse, or by enabling the long tag view.

`<div>` Attributes

Attribute	Type	Description
class	String	Contains the following structure: <i>TYPE DATA</i> where: - TYPE is one of the following strings to define the tag type: - open - close - single - DATA is a encoded version of the tag content.

SHORT_CONTENT `<span>` Attributes

Attribute	Type	Description
class	String	Contains the following fix value: "short"
title	String	Contains the HTML encoded LONG_CONTENT value

LONG_CONTENT `<span>` Attributes

Attribute	Type	Description
class	String	Contains the following fix value: "full"

id	String	The id has the following structure: TAGID-TAGCOUNT-FILENAMEHASH where TAGID is the ID of the tag defined in the import container, or a given string for automatically added tags like for spaces. TAGCOUNT is a counter of tags in the imported file, and with the hash of the filename in FILENAMEHASH the id is getting unique in the whole task. The FILENAMEHASH is also used as the filename of the full tag images when the segment is opened in the editor.
----	--------	--

Examples

Real life example:

For information on why, read the related FAQ entry on using Apache2 with a threaded MPM	For information on why, read the related FAQ entry on using Apache2 with a threaded MPM
Longtag Content visible	Shorttag Content visible

For information on why, read the related FAQ entry on using

<span title="" class="short"><1></div>Apache2 with a threaded MPM <div class="close 2f67"><span title="<a>" class="short"><1><a></div>

Formatted Examples:

For information on why, read the related FAQ entry on using
<div class="open 672069643d2231333822">
<span title="" class="short"><1>

</div>
Apache2 with a threaded MPM
<div class="close 2f67">
<span title="<a>" class="short"><1>
<a>
</div>

Examples of Single Tags:

<div class="single 73706163652074733d2263326130222f">
<span class="short" title="<space/>"><1>
<space/>
</div>
<div class="single 782069643d2231343722202f">
<2>
»
</div>