

# Inline editing

Document styling allows you to show a document's parts, fields and metadata in a visually appealing way. The inline editing feature allows you to edit your documents while using the same layout.

## Usage

The inline editing feature is always on. To use it, you have to add some attributes to your document-styling stylesheet (located at `{wikidata}/resources/skins/{skin}/document-styling/html/{documentType}.xsl`).

## Overview

<code>xmlns:ie="http://outerx.org/daisy/1.0#inlineeditor"</code>	Namespace declaration for the inline editing features. Just add this to the declarations on the root element of your stylesheet.
<code>@ie:form="editdoc"</code>	(Required) This attribute must be present to work with the feature (it gets replaced with a <code>ft:form-template</code> element)
<code>@ie:mode="view edit"</code>	Elements with this attribute will only be shown in the specified mode.
<code>@ie:field="{fieldName}"</code>	Elements with this attribute will be replaced with a field-editing widget when editing the document.
<code>@ie:part="{partName}"</code>	Elements with this attribute will be replaced with a part-editing widget when editing the document.
<code>@ie:name="..."</code>	An element with this attribute will be replaced with an input field for editing the document name
<code>&lt;ie:action type="edit"/&gt;</code>	When viewing a document, this element places an [edit] button on the page. When editing, the element is dropped.
<code>&lt;ie:action type="save"/&gt;</code>	When editing a document, this element places a [save] button on the page. When viewing, the element is dropped.
<code>&lt;ie:action type="cancel"/&gt;</code>	When editing a document, this element places a [cancel] button on the page. When viewing, the element is dropped.

## Caveats

1. You should always test that `/document/@isIncluded= 'false'` before using the "ie:" elements and attributes. (Because the inline editor has no way of telling whether a specific element belongs to the target document or to another document included in the publisher response.)

## Example

Suppose we have a document styling that shows a document with a part on the left and two fields on the right:

```
<xsl:template match="d:document">
  <table>
    <tr>
      <td><xsl:apply-templates select="d:parts/d:part[@name='SimpleDocumentContent']"/></td>
    </tr>
  </table>
</xsl:template>
```

```

    <tr>
      <td>
        <div><xsl:apply-templates select="d:fields/d:field[@name='Foo']"/></div>
        <div><xsl:apply-templates select="d:fields/d:field[@name='Bar']"/></div>
      </td>
    </tr>
  </table>
</xsl:template>

```

Becomes:

```

<xsl:template match="d:document" xmlns:ie="http://outerx.org/daisy/1.0#inlineeditor">
  <div ie:form="editdoc">
    <xsl:if test="/document/p:publisherResponse/d:aclResult/d:permissions/
d:permission[@type='write' and @action='grant']">
      <ie:action type="edit"/>
    </xsl:if>
    <ie:action type="save"/>
    <ie:action type="cancel"/>
    <table>
      <tr>
        <td><div ie:part="SimpleDocumentContent">
          <xsl:apply-templates select="d:parts/d:part[@name='SimpleDocumentContent']/>
        </div></td>
      </tr>
      <tr>
        <td>
          <div ie:field="Foo">
            <xsl:apply-templates select="d:fields/d:field[@name='Foo']"/>
          </div>
          <div ie:field="Bar">
            <xsl:apply-templates select="d:fields/d:field[@name='Bar']"/>
          </div>
        </td>
      </tr>
    </table>
  </div>
</xsl:template/>

```

#### NOTE

A test was added around the `<ie:action type="edit" />` element to hide the button from users who don't have write access.

## Background info

- The inline-editor is started using a POST request to `{mountPoint}/{siteName}/.../{documentId}/inline-edit` (or a GET request to the same URL, with a `"startWithGet=true"` query parameter)
- New documents can be created using the url `{mountPoint}/{siteName}/.../new/inline-edit?documentType={documentType}`
  - Be aware that the `publisherRequestSet` (as configured in the `siteconf.xml`) is not applied when editing new documents. Internally a 'fake' publisher response is used which only contains the most essential data needed to render the new document.
  - You can also use `?template={documentId}` as an alternative to the `documentType` query parameter.
- You can add the `branch` & `language` query parameters to edit a particular variant (for existing and new documents)