

Customization

This chapter describes the possibilities to customize the PDF catalogues provided by the PDF catalogue extension. One section describes the customization of dedicated PDF catalogues while another one describes the customization of the common PDF catalogue. It is assumed that you already downloaded and installed the extension and that you have basic experience with Cascading Stylesheets, eZ Publish and it's administration interface.



Dedicated PDF catalogues

A dedicated PDF catalogue mainly exists of two parts, a target zone and a source zone. The target zone provides an XHTML tag (most likely as an unordered list) that serves as the drop zone of the drag and drop operation used to put documents into the dedicated PDF catalogue. The source zone by default shows up all documents (most likely also as an unordered list) that are part of an dedicated PDF catalogue and can be dragged into the target zone.

Target zone

By default, the target zone is designed to replace the left menu within a default Website interface installation of eZ Publish. The template used to render that part is

```
extension/abpdfcatalogue/design/standard/override/templates/pdf_catalogue_left_menu.tpl
```

If you intend to integrate the target zone somehow else (not in the left menu), you can simply copy the content of this template to any other template. The only requirement for it is, that the content node of the dedicated PDF catalogue is "known" within the other template. By default, this content node is fetched within the left menu template using the template variable `$current_node_id`. This variable is defined in the `pagelayout.tpl` from which the left menu template is included.

If you want to integrate the content of the left menu template for example in the full view template for dedicated PDF catalogues (also mentioned below), you can also replace all occurrences the template variable `$pdf_catalogue_node` (used in the left menu template) with the template variable `$node` (commonly used in full view templates) and remove the definition of the template variable `$pdf_catalogue_node` at the top of the left menu template.

In other cases you have the option to fetch the content node of the dedicated PDF catalogue using a fetch function. For this to work, you will have to know

the content node ID of the dedicated PDF catalogue. To let the template work as expected, just replace the definition of the `$pdf_catalogue_node` template variable as follows.

```
{def $current_node_id = fetch( 'content', 'node', hash( 'node_id', <NODE_ID> ) )}
```

The place holder `<NODE_ID>` has to be replaced with the content node ID of the dedicated PDF catalogue.

If just copying the content of the left menu template somewhere else does not fit your needs, you can also modify the XHTML code within that template. However, when doing so, some requirements still have to be met. Mainly there should be the following three elements available to get a valid target zone.

- An unordered list: `<ul id="pdfCatalogueTargetList">`
- An input field serving as a submit button:
`<input id="pdfCatalogueGenerateButton" type="..." />`
- An image: ``

All these elements are identified by their ID, therefore the `id` attributes shown in the examples above, are required. The unordered list is the target of the drag and drop operations. If you would like to use something else than an unordered list, you have to modify the Javascript code provided by the extension. This should not be necessary if you have experience with Cascading stylesheets allowing you to modify size and design of the elements.

Source zone

The source zone is rendered in a full view template of content nodes that are connected to a content object of the content class `ab_pdf_catalogue`. The full view template is located in

```
extension/abpdfcatalogue/design/standard/override/templates/pdfcatalogue.tpl
```

The related override rule, to make eZ Publish use this template when viewing a content node connected to a content object of the content class `ab_pdf_catalogue` is defined in the file

```
extension/abpdfcatalogue/settings/siteaccess/eng/override.ini.append.php
```

Perhaps you modified the location of this file, if you are using another siteaccess than `eng`. In this case you must either replace `eng` with the name of the siteaccess you are using the dedicated PDF catalogue with or find the file at another location (for example `settings/siteaccess/`).

If you intend to integrate the source zone somehow else (not in the full view of a `ab_pdf_catalogue` content object), you can simply copy the content of this template to any other template. The only requirement for it is, that the content node of the dedicated PDF catalogue is "known" within the other template.

In such cases you will have to make sure that the content node of the dedicated PDF catalogue is available in the template variable `$node`. This can for example be achieved using the `fetch` function described in the previous subsection.

If just copying the content of the full view template somewhere else does not fit your needs, you can also modify the XHTML code within that template. However, when doing so, some requirements still have to be met. Mainly there should be the following elements available to get a valid source zone.

- A tab loader: `<div id="pdfCatalogueTabLoader"></div>`
- A tab container: `<div id="pdfCatalogueTabContainer" class="yui-navset">`
- An unordered list within the tab container: `<ul class="yui-nav">`
- Several list items representing the dedicated PDF catalogue chapters within the unordered list: ``
- A container for the tab contents: `<div class="yui-content">`
- A container for the content of each tab within the tab contents: `<div id="pdfCatalogueTab<INDEX>">` (where the placeholder `<INDEX>` must be replaced with the number of the tab starting with zero)
- An unordered list for each dedicated PDF catalogue chapter within the container for the tab contents: `<ul id="pdfCatalogueSourceList_<INDEX>">` (replace the placeholder `<INDEX>` again)
- Several list items representing a document of the PDF catalogue in the current chapter: `<li id="document_<CHAPTER_ID>_<DOCUMENT_ID>_<FILE_ID>">` (where the placeholders `<CHAPTER_ID>`, `<DOCUMENT_ID>` and `<FILE_ID>` must be replaced with the content object id of: the dedicated PDF catalogue chapter, the dedicated PDF catalogue document and the dedicated PDF catalogue file.

Some of the HTML tags mentioned so far contain several other tags that are mainly used for design reasons but are not required for the dedicated PDF catalogue to work. It is up to you to use them or not.

Finally a list of hidden tags is required to configure the dedicated PDF catalogue and let it work as expected. By default `` tags are used, but it is also possible to use any other kind of non-empty-tag. Make sure the text within the tag does not contain any line breaks or surrounded spaces. Each tag must have an `id` attribute which defines what the text within the tag configures. The following id attributes are required

- `pdfCatalogueCatalogueID`: The content node id of the dedicated PDF catalogue
- `pdfCatalogueDocumentList`: A JSON encoded list of document IDs (see the last item in the above list). Each chapter has an anonymous array with IDs and the arrays of each chapter are put together in another anonymous array
`array(array('document_0_123', ...), array('document_1_254', ...))`
- `pdfCataloguePDFFileName`: The filename that should be used when creating and downloading the dedicated PDF catalogue
- `pdfCatalogueNoDocumentsWarning`: A warning message that could be displayed if the user did not select any document but clicks on the create catalogue button
- `pdfCatalogueDownloadMode`: The download mode which can for now only be SYNC
- `pdfCatalogueDownloadURL`: The URL used to download a dedicated PDF catalogue, most likely `/pdfcatalogue/download`
- `pdfCatalogueGenerateURL`: The URL used to generate a dedicated PDF catalogue, most likely `/pdfcatalogue/generate`

These settings will be read and interpreted automatically by the Javascript library shipped with the PDF catalogue extension. That means it will only work

if the following file is loaded by the page in which the template code has been included

```
extension/abpdfcatalogue/design/standard/javascript/pdfcatalogue.js
```

The evaluation of the hidden fields will start once the page has been loaded completely.

Common PDF catalogue

The common PDF catalogue also has a source and a target zone. But in contrast to a dedicated PDF catalogue, the source zone only consists of one document (the content object connected the content node displayed in the internet browser of the user). The target zone provides an XHTML tag (most likely as an unordered list) that serves as the drop zone of the drag and drop operation used to put documents into the dedicated PDF catalogue. The source zone by default shows up one document (most likely also as an unordered list with one list item) that represents the currently displayed content node and can be dragged into the target zone.

Target zone

By default, the target zone is designed to be integrated into the bread crumbs within a default Website interface installation of eZ Publish (keep in mind that the bread crumbs are not shown on any page). The template used to render that part is

```
extension/abpdfcatalogue/design/standard/templates/pdfcatalogue/commoninclude.tpl
```

This template also contains the definition of the source zone described below. If you intend to integrate the target zone somehow else (not in an include template of the `pagelayout.tpl`), you must make sure, that there is a hidden form with the `id pdfCatalogueCommonGenerateForm` available in the current page. This hidden form must call the well known eZ Publish URL `/content/action`, use the POST method and contain the following hidden input fields

Name	ID	Value
ContentObjectID	-	0
PDFCatalogueItemNodeIDString	pdfCatalogueCommonItemNodeIDString	-
PDFCatalogueGetItemListTranslation	-	<LOCALE_CODE>
PDFCatalogueGenerateCommonBotton	-	PDFCatalogueGenerateCommonBotton

The placeholder `<LOCALE_CODE>` must be replaced with the locale code used by the siteaccess from which the common PDF catalogue is used. For the target zone to be shown somewhere else, you can simply copy the following part of this template to any other template

```
<div id="pdfCatalogueCommonTargetArea"
onmouseover="PDFCatalogueLibrary.showCommonTargetArea();"></div>
```

This tag has no further contents and will be inserted automatically according to the settings configured using the content of the tags with the following `id` attributes.

- pdfCatalogueCommonTargetAreaInsertID: The value of an id attribute of an HTML tag into which the mentioned div tag with the id attribute pdfCatalogueCommonTargetArea should be inserted (the default for the Website interface is "path")
- pdfCatalogueCommonTargetAreaInsertPosition: If the value is "first" the mentioned tag will be inserted as the first child, if it is set to last, it will be inserted as the last child (the default for the Website interface is "first")

The insertion will be done by the Javascript library shipped with the PDF catalogue extension. That means it will only work if the following file is loaded by the page in which the template code has been included

```
extension/abpdfcatalogue/design/standard/javascript/pdfcatalogue.js
```

Furthermore the tag mentioned above will be filled automatically with content, once the current page has been loaded. The content will be rendered using the following template

```
extension/abpdfcatalogue/design/standard/templates/pdfcatalogue/commontargetlist.tpl
```

You can modify this template to change the layout of the target zone according to your requirements, as long as the following elements are available

- An unordered list: `<ul id="pdfCatalogueTargetList" onmouseout="PDFCatalogueLibrary.hideCommonTargetArea();" >`
- A list item in the unordered list that contains the generate button: `<li id="pdfCatalogueGenerateButton">`
- A link in that list item used to generate the common PDF catalogue: ``
- A list item in the unordered list for each document that has already been selected by the user for the common PDF catalogue: `<li id="document_<NODE_ID>">` (the placeholder `<NODE_ID>` must be replaced with the content node ID of the document)
- Optionally a link to remove a document from the target list: `<a href="#remove" onClick="PDFCatalogueLibrary.removeCommonListItemById('document_<NODE_ID>'); return false;">` (the placeholder `<NODE_ID>` must be replaced with the content node ID of the document)
- A list item in the unordered list which is shown if no document has been added to the list so far: `<li id="pdfCatalogueNoItemEntry">`
- A hidden tag pointing out if the user is allowed (1) to generate a common PDF catalogue or not (0): `1`
- A hidden tag that contains a comma separated list of the content node IDs representing the documents in the common PDF catalogue: `2,43`

Within the template there is a variable named `$nodeNameList` available which contains all the documents the current user already put into his common PDF catalogue. Each item in this document list (which is an anonymous array) provides the content node ID (`.node_id`) and the name of a content node (`.name`) in the common PDF catalogue.

Source zone

By default, the source zone is automatically integrated into the full view of any content node in a default Website interface installation of eZ Publish after the page has been loaded. The source zone is rendered in the same template as the target zone

`extension/abpdfcatalogue/design/standard/templates/pdfcatalogue/commoninclude.tpl`

If you intend to integrate the source zone somehow else (not in an include template of the `pagelayout.tpl`), you must make sure that there is an (at first) hidden unordered list available somewhere in the page. This unordered list must have the following characteristics

- An id `pdfCatalogueCommonSource`
- Exactly one list item `<li id="document_<NODE_ID>` (the placeholder `<NODE_ID>` must be replaced with the ID of the content node that should be draggable into the common PDF catalogue
- Optionally a link to remove a document from the target list: `<a href="#remove" onClick="PDFCatalogueLibrary.removeCommonListItemById('document_<NODE_ID>'); return false;">` (the placeholder `<NODE_ID>` must be replaced with the content node ID of the document)

This unordered list will be inserted within an HTML tag according to the settings configured using the content of the tags with the following `id` attributes.

- `pdfCatalogueCommonSourceAreaInsertID`: The value of an `id` attribute of an HTML tag into whichs parent tag the unordered list with the `id` attribute `pdfCatalogueCommonSource` should be inserted (the default for the Website interface is "main-position")
- `pdfCatalogueCommonSourceAreaInsertPosition`: If the value is "first" the mentioned tag will be inserted as the first child, if it is set to last, it will be inserted as the last child of the parent (the default for the Website interface is "first")

Therefore you must make sure that there is exactly one tag having an `id` attribute with the value configured using `pdfCatalogueCommonSourceAreaInsertID`.

Source and target zone

A list of hidden tags is required to configure the common PDF catalogue and let it work as expected. These hidden tags must be available somewhere in the current page. By default `` tags are used, but it is also possible to use any other kind of non-empty-tag. Make sure the text within the tag does not contain any line breaks or surrounded spaces. Each tag must have an `id` attribute which defines what the text within the tag configures. The following `id` attributes are required

- `pdfCatalogueNodeID`: The ID of the content node that is currently shown and should be draggable into the common PDF catalogue
- `pdfCatalogueCommonTargetActionURL`: The well known `/content/action` URL of eZ Publish
- `pdfCatalogueCurrentLocale`: The locale code used by the current siteaccess