

Flash 2 TYPO3

Extension Key: mwflash2t3

Language: en

Keywords: Flash, XML, Templating, Templates, Flexible Content Elements

Copyright 2000-2010, Maximilian Wambach, <info@der-gestalt.de>

This document is published under the Open Content License
available from <http://www.opencontent.org/opl.shtml>

The content of this document is related to TYPO3
- a GNU/GPL CMS/Framework available from www.typo3.org

Table of Contents

Flash 2 TYPO3.....	1	Remapping values.....	12
Introduction.....	3	Inserting the frontend-plugin with TypoScript.....	13
What does it do?.....	3	JavaScript information.....	13
Key Features.....	3	Further development.....	13
Screenshots.....	3	Configuration.....	14
Users manual.....	6	Reference.....	14
Target user group.....	6	Examples.....	15
How-To.....	6	Tutorial.....	16
Installation.....	6	Upload your files.....	16
Setup.....	6	Add storage folder.....	16
Adding static TypoScript template.....	6	Install the extension.....	16
Creation of XML File.....	6	Adding static TypoScript template.....	17
How to insert the plugin into a page.....	7	Create some flexible content elements.....	17
Mapping.....	8	Adding a XML file to the extension.....	17
Generating the XML Template.....	10	Add the plugin to your page.....	18
Subsequent information.....	10	Map the template.....	19
Regenerating the XML.....	10	Known problems.....	22
Administration.....	12	To-Do list.....	23
Extension dependencies.....	12	Wishlist.....	24
Available eTypes	12	ChangeLog.....	25
Checklist.....	12		

Introduction

What does it do?

This extension allows you to administrate your flash-module with TYPO3. If you either want to a content management system for your Full-Flash-Website or just want to administrate your single flash module – it can all be done with "Flash 2 TYPO3".

The requirement for this extension to work is that the content of your flash module is delivered by XML. Through a Point and Click interface you are able connect your xml nodes with flexible content elements and serve your flash with values of the fce's. Note that this is a very alpha version, so read the manual carefully.

Extension is based on lib, div and templavoila.

Key Features

- Point and Click mapping of XML nodes and attributes in TemplaVoilà style
- content delivered by flexible content elements
- Flash embedding with swfobject
- fully customizable through flashvars and flashparameters
- configuration through flexform and TypoScript
- For a live preview see <http://typo3fl.webseiten.cc>

Screenshots



Fully customizable emedding of Flash

Frontend Data	Configuration	Startingpoint
Flash Parameter		
quality = high wmode = transparent		
Flash Vars		
stylesheet = fileadmin/templates/css/main.css		
Width		
1152		
Height		
864		

Configuration section

```

Node <T3lovesFlash>
  Node <sites>
    Node <site Attribute checked="" id="0" Attribute checked="" p
      Node <content Attribute checked="" type="G
        Node <images>
          Node checked="" <image_
            pictures/typo3_
          </image_one>

```

Mapping section

XML Information

Now choose a field from found Flexible Content Elements.

Chosen Tag

Path to Tag

Choose the Flexible Content Element you would like to map

FCE Titel	Found Fields in FCE
 xml_fce [Template]	➡ field_site_id, ➡ field_parent_id, ➡ field_title,

FCE mapping

Users manual

Target user group

Since this extension is in alpha state the extension can only be used by admins. With further development the extension will open up for a larger group of users.

How-To

For a step-by-step explanation please consult the Tutorial section of this documentation.

Installation

The extension has to be installed like any other extension. Install the extension with the extension manager. After the extension manager created the new database tables and fields for mwflash2t3 you should see a page asking you for a storage folder where the extension is able to save its data.

Setup

Before you can use mwflash2t3 you have to create a storage folder, unless you haven't done this before the installation. After having created the folder you must enter it in the configuration section of the extension. For that go to the extension manager and click on the extension name. You now should see the form asking you to enter the pid of your newly created folder. After pressing update the extension is set.

Adding static TypoScript template

To make this extension work you have to include the static TypoScript template delivered with this extension. Normally you do this in your root template.

Creation of XML File

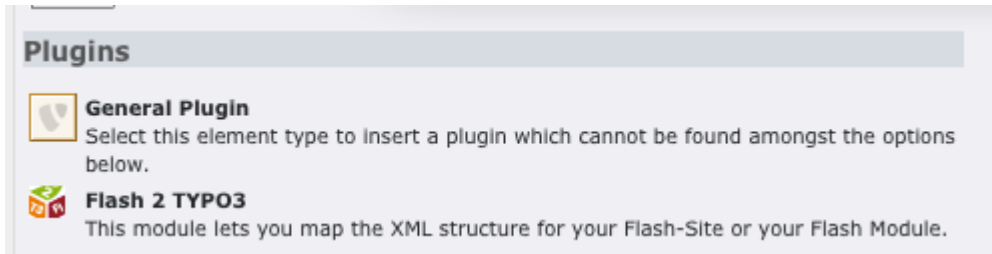
First of all you should start with creating a XML file entry in the database. For that reason navigate to your storage folder in the page tree. Click on it and say "Create new record". In the tree scroll to "Flash 2 TYPO3". There you should see the entry "XML File". Just click on it and you get to the form where you enter a title for your XML and choose your XML file in the fileadmin section.

After you have entered your information save it.

How to insert the plugin into a page

The plugin can be inserted in any area of your page. You can do this by TypoScript or by flexform. For a how to of the TypoScript version please consider reading Administration section.

To insert to the plugin create a new conten on your page. In the plugin section you should find the frontend-plugin "Flash 2 TYPO3".



Just click on the entry and you should come to the flexform of the plugin. The first tap lets you insert the frontend data. Just choose our swf-files and a XML file that has been saved in the database. It does not matter that there is no mapping data for the xml you choose. Mapping is done in the next steps.

There is no need to enter more than one swf. But to keep it flexible you can choose more than one SWF file if you need them. All files entered in the second section are uploaded to the extension upload folder. So if your main SWF file loads other SWFs dynamically you should enter them here.

The SWF entered in the first section "SWF File" is taken for embedding with swfobject.

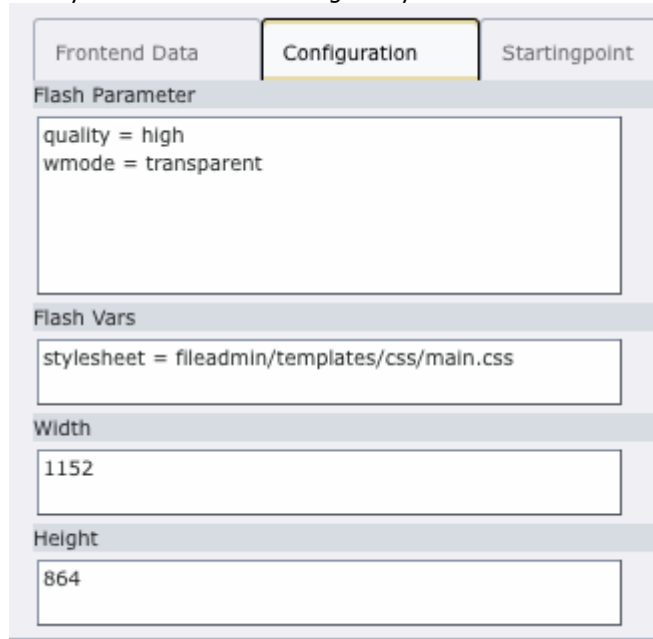
The next tab includes the configuration section. There you are able to enter FlashVars or Flash Parameters. Both are entered in the following way.

Example:

[nameOfVar] = [value]

[nextNameOfVar] = [thisValue]

Values are separated by backslash. Names of parameters or variables are separated by “=”. Furthermore you have the opportunity to enter width and height of your flash movie.

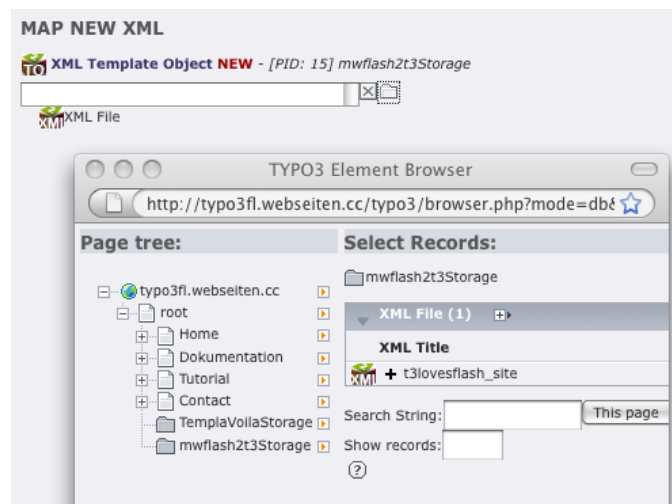


Startingpoint is useless at the moment since the needed functions are not implemented yet.

Mapping

To map your XML change to the “Flash 2 TYPO3” backend-module and navigate to your storage pid. Since you have not mapped anything you just see this interface.

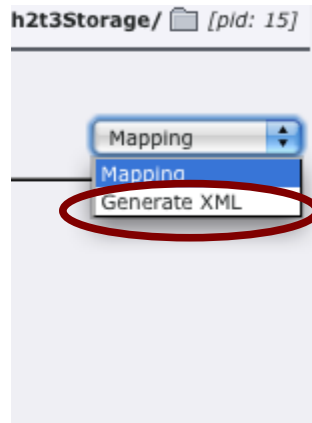
In the section “Map new XML” you can choose your desired XML File in your storage folder. If you are ready just save the new template object (save button on top).



your XML, so all you need to do, is to choose your root-node and go on mapping until all fields and nodes are linked.

Generating the XML Template

Once all your XML nodes or attributes are linked with the according fields of your FCE you can generate your XML template. You do this by choosing "Generate XML" from the option menu on the top right.



After that you see a status screen whether the generation of the xml template was successful or not. If there has been any failure this may be caused by insufficient writing permissions.

CHOOSE YOUR XML:

uid	Base XML	Mapping Data	Data Structure			
1	t3lovesflash_site	✓ Mapping Data found and valid	✓ Data Structure found and valid	♻️	✎	🗑️

MAD NEW XML

Subsequent information

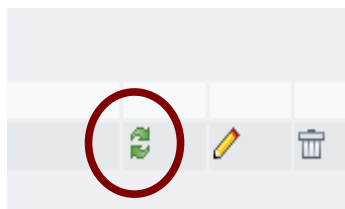
All data that belongs to "Flash 2 TYPO" is saved in its upload folder. So is the newly generated XML template. The FlashVar "xmlPath" is set with the Path to the newly generated XML. If you are not using FlashVar to tell Flash what XML to load you can hardcode the path to the XML.

Of course it is recommended to work with Flash Variables. Most notably you are able to change the value of "xmlPath" to whatever you want by TypoScript. For more information please read the configuration chapter.

Regenerating the XML

If you make changes to your FCE data you have to tell this to your XML. In this state of the development there is unfortunately no way to overcome this additional work.

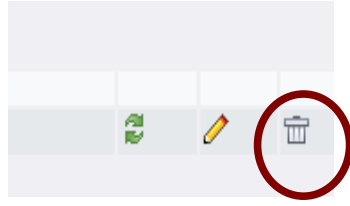
So if you make changes to your data you have to click on the regenerate button in the overview.



If you want to change the mapping data, you can do this by click on the edit pencil.



To delete the XML object choose the garbage icon.



Administration

Since the extension is only accessible for admin users the user's manual chapter includes administration of the extension. Nevertheless here will be given a more in depth view in the extension and solutions to problems that might be occur during installation etc.

Extension dependencies

"Flash 2 TYPO3" depends on lib, div and templavoila. So the frontend plugin implements the Model-View-Controller pattern. The dependency on templavoila might be deleted in future versions. In version 0.1.0 you are only able to map flexible content elements. For that reason templavoila has to be installed.

Available eTypes

The extension does not support every eType of TemmplaVoilà. The supported eTypes are listed below.

eType	Information
text	Supported and wrapped in CDATA sections
input	Supported and wrapped in CDATA sections
image	Supported and wrapped in CDATA sections. The image filename is prepended by the templavoila upload path.

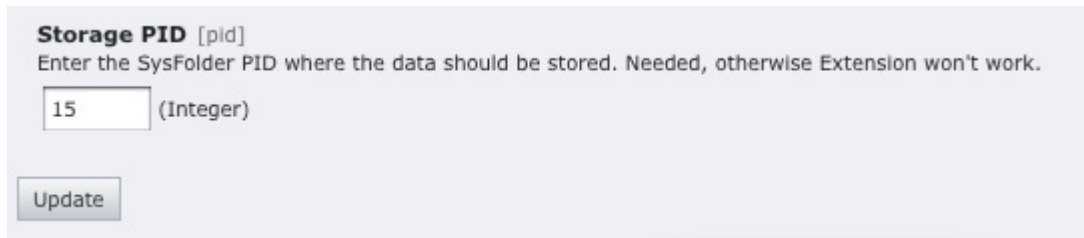
More eTypes will be supported in future versions. Just come back here for more information.

Checklist

If you run into problems this section might be helpful for you.

- Problem: I can't save my XML Template Object

Check if you have set your storage pid id in the information section of the extension



- Problem: After mapping my XML structure is repeated in a very strange way.

Did you choose the right XML root-node? Check that. If you find the failure here you have to remap your XML again. Just delete it in the overview and choose the XML again. Now map the right root.

- Problem: There is no content to map

Make sure that you have flexible content elements on your page. If not generate some and map your XML again. Now you should see flexible content elements in the overview.

- Problem: I don't know how to start.

Just have a look in the tutorial section of this document.

Remapping values

After you have mapped your XML, you don't have to worry about changes in the structure of your flexible content elements. You always have the possibility to edit your mapping data by clicking the edit pencil.



In the next screen you can remap your values. Be aware that until version 0.1.0 this is not possible for the root-node.

Inserting the frontend-plugin with TypoScript

These lines of TypoScript belong into the setup area of your template.

```
temp.configuration < plugin.tx_mwflash2t3.configuration
temp.configuration {
    xmlFile = 30
    swf = main.swf
    additionalSwf = T3lovesFlash.swf
}

plugin.tx_mwflash2t3.controller_default < temp.configuration
plugin.tx_mwflash2t3.controller_default {
    userFunc = tx_mwflash2t3_controller_default->main
    defaultAction = initAction
}
lib.flash < plugin.tx_mwflash2t3.controller_default
```

This loads the main.swf file with the additional SWF file T3lovesFlash.swf. Of course you have set your own values here. The XML file has to be referenced by the UID of your template object! This is very important to understand. Just go to list view or the "Flash 2 TYPO3" backend module to find out the UID.

You can set the width and height of your flash movie by entering these lines into the constants section.

```
plugin.tx_mwflash2t3 {
    width = 1152
    height = 750
}
```

For more information have a look at the configuration chapter.

JavaScript information

The flashvar object is named flashvars. The object that holds the parameters is called params. This is hardcoded at the moment. If you need this to be set by TypoScript just send me a mail, so will change this immediately. Otherwise this will be a feature of the next version.

Further development

The extension as it is in the moment needs more development. What I wanted to show is that there are possibilities to turn TYPO3 in a fully customizable Flash CMS. Since the extension has been developed while I was writing my thesis there was only a given amount of time. So the extensions is a very early alpha version but will become more powerful with further development.

Configuration

The extension is fully customizable through TypoScript. All values that can be set in the flexform can be set by TypoScript. Furthermore there are some properties that can only be set by TS.

Reference

TypoScript setup section

These properties are available under

`plugin.tx_mwflash2t3.configuration`

Property:	Data type:	Description:	Default:
pathToTemplateDirectory	string	Path to the Template Directory	EXT:mwflash2t3/templates/
pathToLanguageFile	string	Path to language file	EXT:mwflash2t3/locallang.xml
pathToUploadFolder	string	Path to upload folder of extension	uploads/tx_mwflash2t3/
basicTemplate	string	The name of the frontend template without ".php"	basic
divId	string	The id of the html div to which swfobject maps the flash file. So you are able to set your desired div#id. Example: <code><div id="yourvalue"></div></code>	tx_mwflash2t3
startPoint	integer	The start point in your page tree. NOT IMPLEMENTED YET!	25
swf	string	The main SWF file. Either you enter an relative path to your main SWF file in the upload folder of the extension	test.swf
xmlFile	integer	The UID of the XML Template Object.	0
flashVersion	string	Needed Flash version to display your flash movie. Can be point separated Example: 9.0.1	9
additionalSwf	string	Additional SWF files in the upload folder. Can be more files separated by " ". Example: additionalSwf = first.swf second.swf third.swf	0
flashparams	string/array	Flash Parameters to be inserted in the JavaScript. Must be formatted like this <pre>flashparams { scale = noscale salign = tl wmode = window quality = high }</pre>	scale= noscale salign=tl wmode=window quality=high
flashvars	string/array	Flash Variables to be inserted in the JavaScript. Add them like this <pre>flashvars { xmlFile = xml css = css }</pre>	
flashVarXMLPath	string	The FlashVar name that that holds the path to the mapped XML Example flashvar.[yourname]= "path/to/xml/file.xml" With default value: flashvar.xmlPath= "path/to/xml/file.xml"	xmlPath
width	integer	Can be set in the constants too. Width of your Flash movie	800
height	integer	Can be set in the constants too. Height of your Flash movie	600

TypoScript constants section

These properties are available under

`plugin.tx_mwflash2t3.`

Property:	Data type:	Description:	Default:
width	integer	Width of your Flash movie	800
height	integer	Height of your Flash movie	600

Examples

How to insert "Flash 2 TYPO3" with TypoScript

You can insert the plugin to your page using the following TypoScript.

```
temp.configuration < plugin.tx_mwflash2t3.configuration
temp.configuration {
    xmlFile = 30
    swf = main.swf
    additionalSwf = T3lovesFlash.swf
}

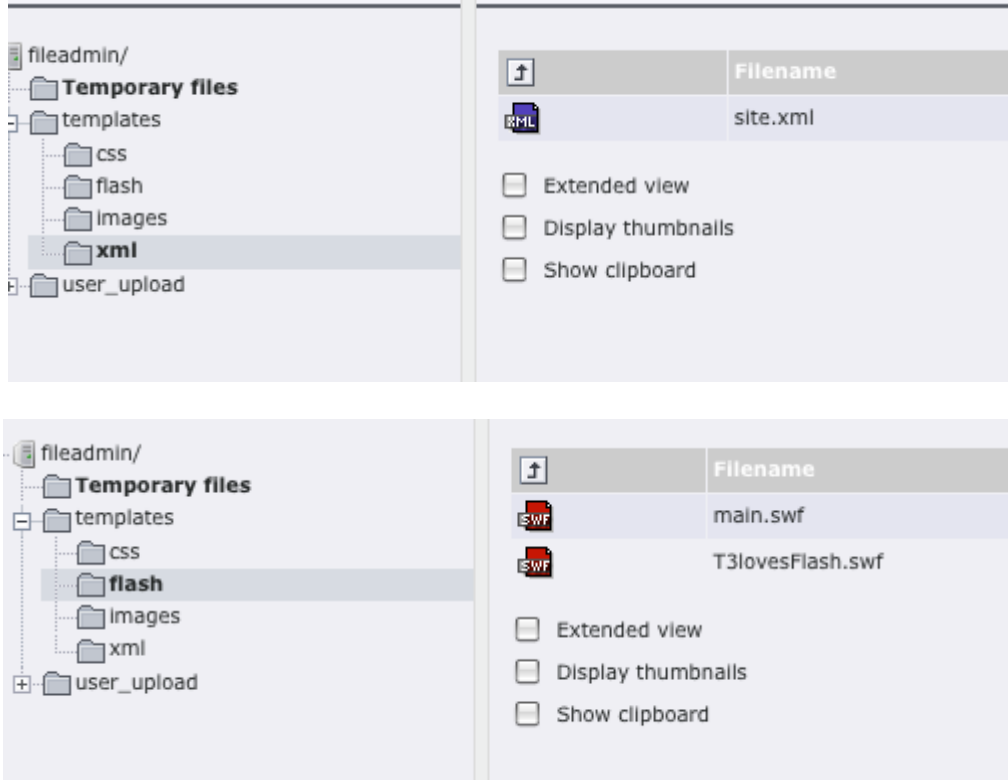
plugin.tx_mwflash2t3.controller_default < temp.configuration
plugin.tx_mwflash2t3.controller_default {
    userFunc = tx_mwflash2t3_controller_default->main
    defaultAction = initAction
}
lib.flash < plugin.tx_mwflash2t3.controller_default
```

Tutorial

This provides a complete walk through the mapping process. So let's start.

Upload your files

First of all you should upload your files to the server. Choose the fileadmin folder (or whatever you have access to) and upload your XML and SWF files.

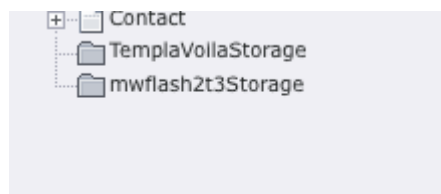


The XML should look similar to this.

```
<T3lovesFlash>
  <sites>
    <site id="0" title="Home">
      <content type="0">
        <images>
          <image_one>pictures/typo3_logo.png</image_one>
          <image_two>pictures/flash_logo.png</image_two>
        </images>
        <text><![CDATA[<h2>A Headline</h2><p>And some text with a <a href="http://www.typo3.org">link</a></p>]]></text>
      </content>
    </site>
  </sites>
</T3lovesFlash>
```

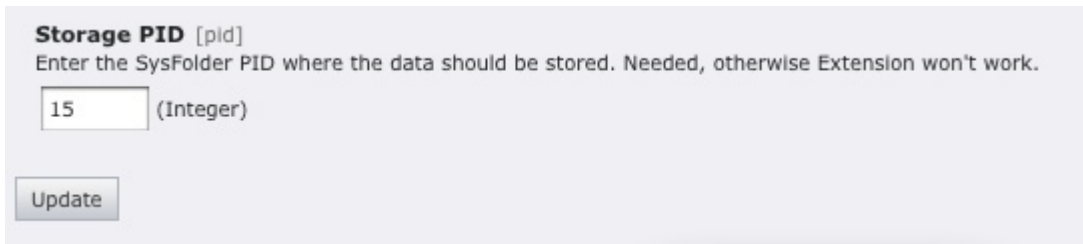
Add storage folder

Add a storage folder for the extension to your page tree. Find out its PID and write it down.



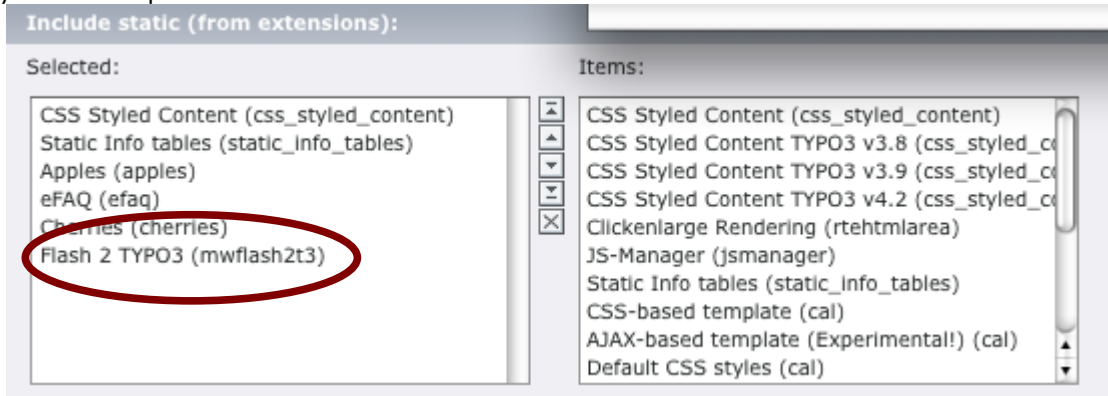
Install the extension

Now install the extension as you would do with any other extension. After you installed it and pressed the update button there the extension should ask you for the storage PID. Enter it and press "Update".



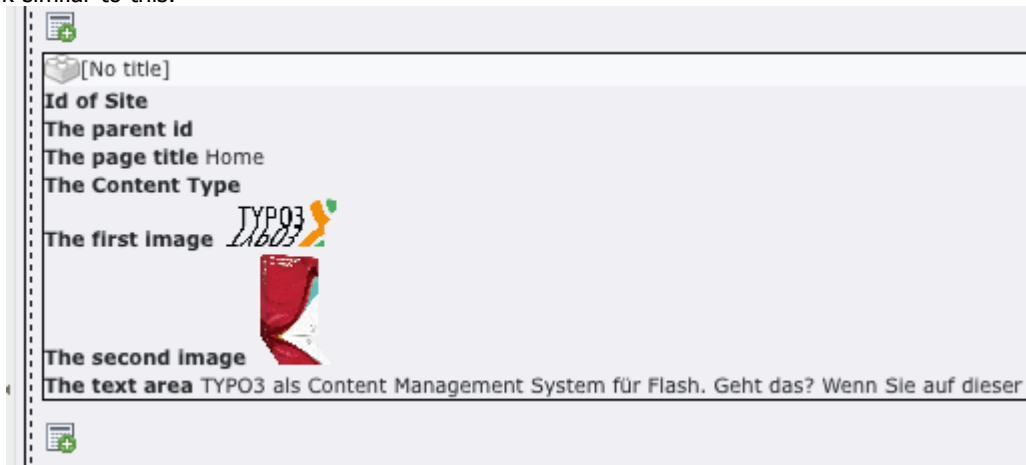
Adding static TypoScript template

To make this extension work you have to include the static TypoScript template delivered with this extension. Normally you do this in your root template.



Create some flexible content elements

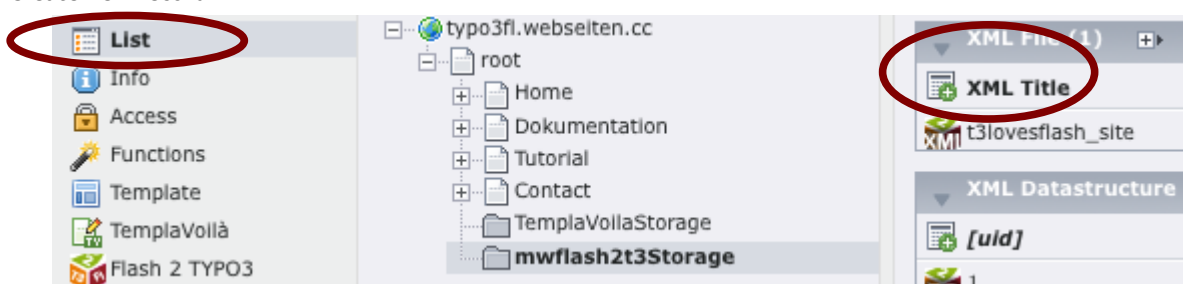
If already have done this you can stop reading and go on to the next section. Otherwise create a flexible content element that might look similar to this:



Please be aware that only eTypes "text", "input" or image are supported by now. Keep that in mind if you are trying to make a suitable flexible content element.

Adding a XML file to the extension

To add an your XML file to the extension go to "List" and choose your storage folder. Now add an XML file by clicking on "Create new record".



In the next screen enter a title and set the path to your XML file.

The screenshot shows a configuration form with three sections:

- XML Title:** A text input field containing the value "t3lovesflash_site".
- XML File:** A file selection field containing the value "site.xml".
- XML:** An empty text input field.

Add the plugin to your page

Now we are ready to add the frontend-plugin to the page. If you are administrating a full flash site the plugin should be added at the root page. Otherwise place it wherever you like.

The screenshot shows the 'Plugins' section of a CMS interface. It contains two plugin options:

- General Plugin:** Select this element type to insert a plugin which cannot be found amongst the options below.
- Flash 2 TYPO3:** This module lets you map the XML structure for your Flash-Site or your Flash Module.

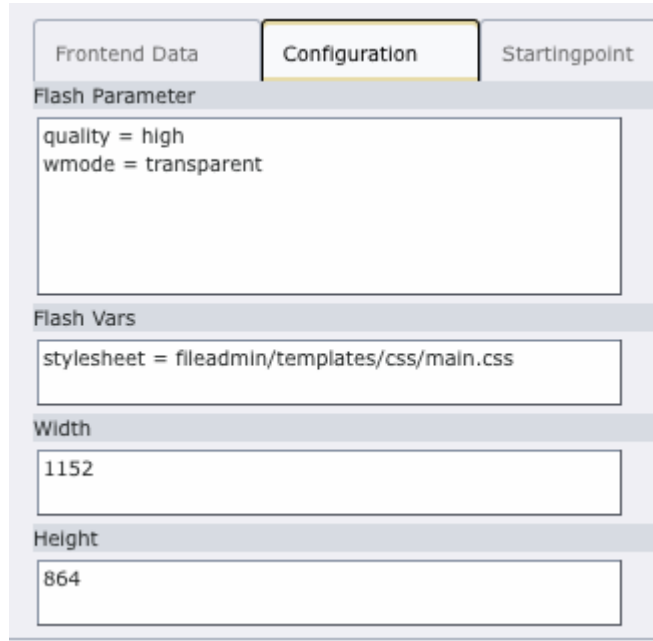
After having selected "Flash 2 TYPO3" you get to the flexform configuration.

Enter your values according to the user manual. In the XML File section you have to choose the XML inside your storage folder.

The screenshot shows the 'Frontend Data' configuration tab for the Flash 2 TYPO3 plugin. It includes the following sections:

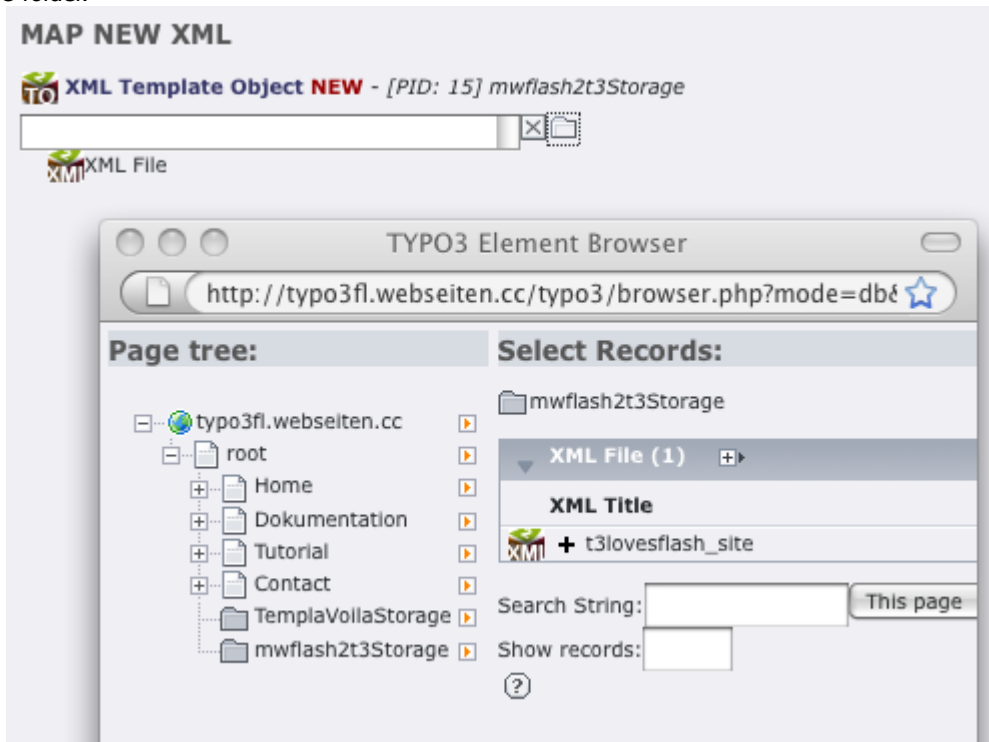
- SWF File:** A file selection field containing the value "main.swf".
- SWF -PHP -PHP3:** An empty text input field.
- Additional SWF Files (if needed):** A list box containing the value "T3lovesFlash.swf".
- SWF -PHP -PHP3:** An empty text input field.
- XML File:** A file selection field containing the value "t3lovesflash_site".

Startingpoint can be ignored by now.



Map the template

Now we are ready to map the XML template. Navigate to the backend-module of "Flash 2 TYPO3". You should see a screen that tells you that there is no mapping data yet. Click on the folder symbol of the data relation field and choose your xml from the storage folder.



Now click the save button on the top. You should be directed to your XML with your nodes and attributes marked up. The extension asks you to choose the root of your XML.

Map Root Element

First Map your root element. Please be aware that until this state of development



Important: the root node is the node that is repeated if more flexible content elements are found on your page.

Let's say you have the following XML structure:

```
<data>
  <innerData>
    <image></image>
    <text></text>
  </innerData>
</data>
```

and you map

```
<innerData>
```

as your root element and you have two different flexible content elements on your site then the generated structure looks like this:

```
<data>
  <innerData>
    <image>image from first FCE</image>
    <text>text from first FCE</text>
  </innerData>
  <innerData>
    <image>image from second FCE</image>
    <text>text from second FCE</text>
  </innerData>
</data>
```

In most cases this is the desired output. Until now there is no possibility to set different root-nodes.

After you have chosen the root you see the same XML again but with the root-node checked. You cannot change the root by now. This will be possible in future versions.

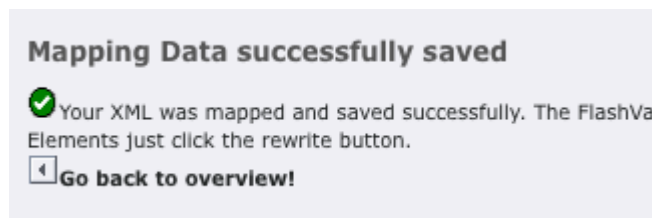
Now you can start mapping. Select a node or attribute. The next screen shows the found flexible content elements. The fields on the right side are clickable. If you click on your desired field the XML node or attribute are linked with the field and the field's value will be taken.



You can keep on mapping until all fields are linked. If you have finished navigate to the upper right select menu and choose "Generate XML".

Your XML is generated and filled with the values of your flexible content element.

You should see this screen:



The new XML is written to the extensions upload folder. The FlashVar "xmlPath" is set to the path of your newly mapped XML. If you look at the source code of your page you should see something like this

```
var flashvars = {};
flashvars.stylesheet = "fileadmin/templates/css/main.css";
flashvars.xmlPath="uploads/tx_mwflash2t3/t3lovesflash_site.xml";
flashvars.relPathToApplication = "uploads/tx_mwflash2t3/";
var params = {};
params.quality = "high";
params.wmode = "transparent";
swfobject.embedSWF("uploads/tx_mwflash2t3/main.swf","tx_mwflash2t3","1152","864","9","../typo3conf/ext/mwflash2t3/flash/expressInstall.swf",flashvars, params)
```

Known problems

- If you interrupt the mapping process by using the back button of your browser the mapping can be damaged.
- If you have a XML node with the same name twice and both are on the same level only one of them is mapped.

To-Do list

- Mapping of more eTypes of TemplaVoilà
- More than one root-node
- Mapping of different content types
- Adding more flexibility by adding more TypoScript variables

Wishlist

- API for external extensions so that they can subscribe to the functions of "Flash 2 TYPO3"

ChangeLog

- Initial upload on March, 4th 2010.