

# EXT: web2print

Extension Key: **web2print**

Copyright 2013-2014,

Michael Gebhart, <[m.gebhart@connetation.at](mailto:m.gebhart@connetation.at)>  
Paul Leitner, <[p.leitner@connetation.at](mailto:p.leitner@connetation.at)>,  
Rüdiger Mausz, <[r.mausz@connetation.at](mailto:r.mausz@connetation.at)>

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.com](http://www.typo3.com)

## Table of Contents

<b>EXT: web2print</b> .....	<b>1</b>
<b>Introduction</b> .....	<b>2</b>
What does it do?.....	2
<b>Users manual</b> .....	<b>3</b>
Installation.....	3
Quick start:.....	4
Base Configuration.....	4
Single Configuration.....	4
<b>Administration</b> .....	<b>5</b>
Rights & Permissions.....	5
<b>Configuration</b> .....	<b>6</b>
Files:.....	6
Reference.....	7
plugin.tx_web2print_lib TS constants.....	7
plugin.tx_web2print_pi1 TS constants.....	7
plugin.tx_web2print_pi1 TS configuration.....	7
<b>Usage</b> .....	<b>8</b>
Backend.....	8
Creating a base configuration.....	9
Backend module.....	10
Creating a single configuration.....	11
Adding content to the single configuration.....	12
Downloads in a single configuration.....	16
Frontend.....	17
Previewing content.....	17
Starting the rendering process.....	20
<b>Extending web2print</b> .....	<b>21</b>
Render library interface.....	21
<b>List of web2print add-ons</b> .....	<b>22</b>
Render library for pages.....	22
Render library for PDF files.....	22
Render library for commerce categories/products/articles.....	22
Render library for table of contents.....	22
Render library for index page (commerce articles).....	22
<b>w2pservice</b> .....	<b>23</b>
<b>Sponsoring web2print development</b> .....	<b>24</b>
<b>Known problems</b> .....	<b>24</b>
<b>To-Do</b> .....	<b>24</b>
<b>Changelog</b> .....	<b>24</b>

# Introduction

## What does it do?

Web2Print is a framework to create HTML5 content out of different sources.

The extension displays the content in a preview environment which resembles the single rendered pages in the finished document.

Since rendering – especially for print production – is a time consuming task, the preview provides the user with the possibility to check the result.

The rendering process itself is finished by the w2pservice, a server-side ensemble of different tools and proprietary code taking care of converting existing RGB Images to CMYK colorspace if needed and creating a print-ready PDF-X3 document.

# Users manual

## Installation

Installation of the extension is done through the TYPO3 extension manager as usual. The supplied rendering libraries have some dependencies:


- commerce extension
- installed locales for needed money/time format
- ImageMagick/GraphicsMagick for rendering
- gdlib for text-size estimation

Information ▾

### Extension Manager

Extension: `c://connetation:// w2p` (web2print)

**Current status:**

**The extension is installed (loaded and running)!**  
Click here to remove the extension: 

**Configuration:**

*(Notice: You may need to clear the cache after the configuration of the extension. This is required if the extension adds TypeScript depending on these settings.)*

#### Enable features

**Web2Print PID** [w2pPid]  
PID where web2print configs and frontend plugin are located

**Web2Print BaseURL** [w2pBaseUrl]  
Frontend BaseURL for Web2Print View (e.g. subdomain http://web2print.<domain>.<tld>/)

**License Key** [licenseKey]

*Web2Print configuration in extension manager*

## Quick start:

After installation, the first step would be to create a page for both the frontend plugin and the needed configuration records. This page needs to include the static TS template for basic setup purposes. The pid of this page has to be set in the constant and in the extension configuration (extension manager) because the backend module has to know where single configuration records and the frontend plugin are inserted.

A user storage folder is needed as well. The pid of this folder has to be set on the page for web2print.

A rendering configuration consists of two hierarchical configuration types.

## Base Configuration

The base configuration defines general options such as the used language, paper-size, templates for content rendering and the included stylesheets for screen display and print output (which are subsequently used in the PDF rendering process).

Additionally permissions for this specific base configuration in frontend and backend are defined through usergroups.

## Single Configuration

A single configuration is always dependent on a base configuration.

The single configuration is the real definition of rendered content.

This is achieved by adding contents to the configuration.

The content-types correspond with a specific rendering library which takes care of set-up of content data-element as well as final rendering of HTML content.

Default settings for resolution and colorspace are also defined in the single configuration.

The downloads inside a single configuration resemble the finished rendering jobs and are attached through a callback mechanism controlled by the w2pservice.

# Administration

## Rights & Permissions

One important part is the permission setting on the page containing the configuration records. The backend usergroup defined in a base configuration needs write access to this page, otherwise no saving of single configurations is possible.

# Configuration

## Files:

File:	Description:
pi1/class.tx_web2print_pi1.php	Main PHP-class used to display the rendering overview and calling the specific rendering libraries
lib/class.tx_web2print_element.php	Base class for all types of elements (document, page, contentelement)
lib/class.tx_web2print_document.php	Document class, container for pages of a document
lib/class.tx_web2print_page.php	Page class, contains single contentelements, which can resemble arbitrary content
lib/class.tx_web2print_contentelement.php	ContentElement class, this is the main class processing content for a page
lib/class.tx_web2print_data.php	Data specific helper functions
lib/importlog_wizard.php	Backend wizard for matching proprietary excel file with existing commerce products
lib/PHPExcel.php	Main class of the PHPExcel library (resides in PHPExcel subfolder)
mod1	Dummy module to create top backend menu entry
mod2/index.php	Main class of backend module for single configuration management
renderlib/class.tx_web2print_renderinterface.php	Definition of the interface for rendering libraries
renderlib/class.tx_web2print_render_pdf.php	Renders a preview of given PDF file and submits the file to w2pservice for inclusion in the final document
renderlib/class.tx_web2print_render_page.php	Renders TYPO3 page content
renderlib/class.tx_web2print_render_toc.php	Renders the table of contents, renderlibs incorporate a renderToc method for this task
renderlib/class.tx_web2print_render_index.php	Renders an index page for existing commerce articles in the document
renderlib/class.tx_web2print_render_excel.php	Renders content based on excel files
renderlib/class.tx_web2print_render_commerce.php	Renders content for commerce categories, products and articles
resources	Contains stylesheets, templates and base files for the rendering process
static	Contains the static TS templates
filechecksum.php	used to check if transmission/retransmission of a file is needed
w2p.php	AJAX interface for the w2pservice

## Reference

### plugin.tx\_web2print\_lib TS constants

Property:	Data type:	Description:	Default:
w2pPid	int	PID of page containing the Web2Print configurations and the frontend plugin  <b>Example:</b> <code>plugin.tx_web2print_lib.w2pPid = 1</code>	0
userPid	int	PID of page containing the Web Site users for frontend login  <b>Example:</b> <code>plugin.tx_web2print_lib.userPid = 2</code>	0

### plugin.tx\_web2print\_pi1 TS constants

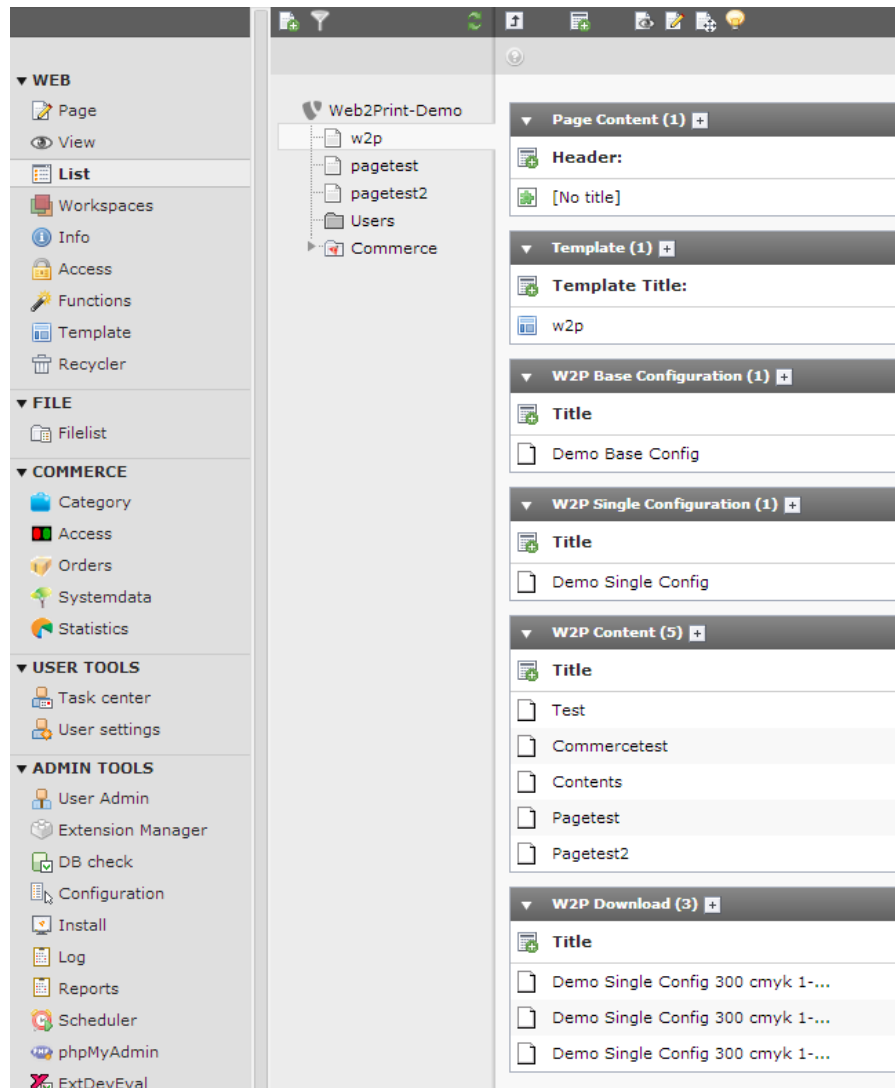
Property:	Data type:	Description:	Default:
templateFile	resource	Template file for Web2Print Frontend Framework (login/header area, rendering panel, javascript for handling rendering and status updates)  <b>Example:</b> <pre>plugin.tx_web2print_pi1 {   templateFile =   fileadmin/templates/w2p.html }</pre>	EXT:/web2print/pi1/template.html
loginTemplateFile	resource	Template file for login form in header of Web2Print frontend plugin, basically a modified felogin template file  <b>Example:</b> <pre>plugin.tx_web2print_pi1 {   loginTemplateFile = fileadmin/templates/   w2plogin.html }</pre>	EXT:/web2print/pi1/loginform.html

### plugin.tx\_web2print\_pi1 TS configuration

Property:	Data type:	Description:	Default:
loginPid	int	PID of page where users can login (usually same as the Web2Print frontend plugin page)  <b>Example:</b> <code>plugin.tx_web2print_pi1.loginPid = 1</code>	{\$plugin.tx_web2print_lib.w2pPid}
loginForm	USER	Contains a configured instance of plugin.tx_felogin_pi1 with assigned storagePid and templateFile according to above set constants	instance of plugin.tx_felogin_pi1
userGroup.render	int	Usergroup with permission to render a document	1
userGroup.view	int	Usergroup with permission to preview a document	1
userGroup.finalversion	int	Usergroup with permission to render a document as final version	1
renderParameters		Parameters for rendering libraries, especially for commerce rendering library defining default height of categories and products for space calculation and positioning	

# Usage

## Backend



*Records inside the w2p page (configurations, contents of single configuration, downloads)*



## Creating a base configuration

The base configuration defines, as the name suggests, the basic properties of a document you want to render. Those include the page size as well as a bleed setting, the language for the rendered document, HTML template to use, general CSS stylesheets and stylesheets for print (which get applied upon rendering) and the permission settings for frontend and backend through usergroups.

**Edit W2P Base Configuration "Demo Base Config" on page "w2p"**

**Hide:**

**Title**  
Demo Base Config

**Language**  
▼

**Market**  
▼

**Paperwidth (mm)**  
210

**Paperheight (mm)**  
297

**Bleed left (mm)**  
3

**Bleed right (mm)**  
3

**Bleed top (mm)**  
3

**Bleed bottom (mm)**  
3

**Template**  
demotemplate.html

HTML

**Base Stylesheet**  
styles\_prince.css    
styles\_prince\_a4.css

CSS

**Print Stylesheet**  
styles\_prince.print\_a4.css

CSS

W2P

Website Usergroup

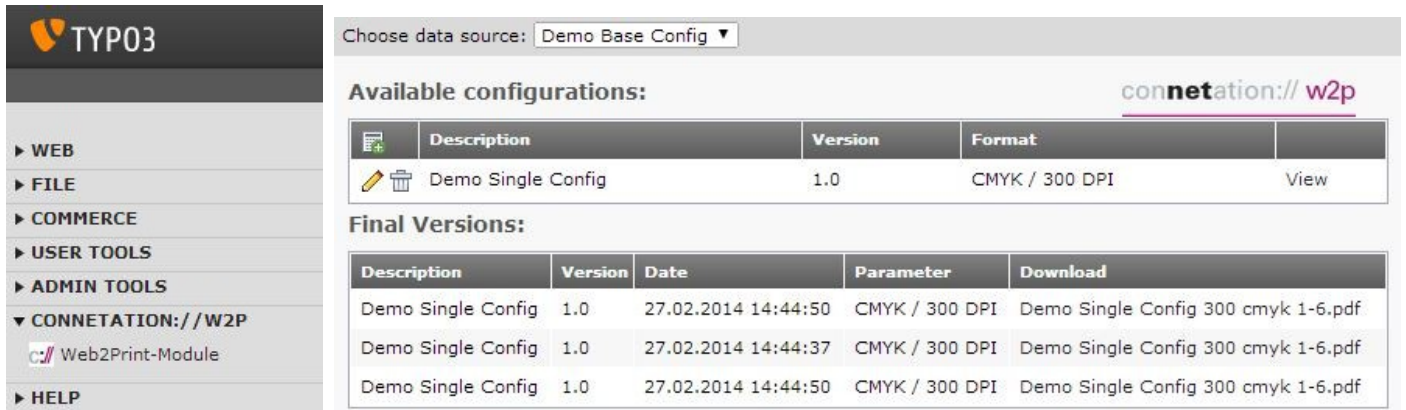
Backend usergroup

*Base configuration with assigned values*

## Backend module



The Web2Print backend module provides an easy way to handle the different single configurations of a given base configuration.

After selecting the data source (base configuration) all available single configurations are displayed as well as final versions of documents rendered with these single configurations.



Choose data source:

**Available configurations:** [connetation:// w2p](#)

Description	Version	Format	
  Demo Single Config	1.0	CMYK / 300 DPI	<a href="#">View</a>

**Final Versions:**

Description	Version	Date	Parameter	Download
Demo Single Config	1.0	27.02.2014 14:44:50	CMYK / 300 DPI	<a href="#">Demo Single Config 300 cmyk 1-6.pdf</a>
Demo Single Config	1.0	27.02.2014 14:44:37	CMYK / 300 DPI	<a href="#">Demo Single Config 300 cmyk 1-6.pdf</a>
Demo Single Config	1.0	27.02.2014 14:44:50	CMYK / 300 DPI	<a href="#">Demo Single Config 300 cmyk 1-6.pdf</a>

*Backend module in TYPO3 navigation*

*Web2Print backend module view*

## Creating a single configuration

Each base configuration can have various single configurations with different content or settings. The original idea was to have the possibility of creating different versions of the same document.

Inside a single configuration the actual content of the document is defined. This is done by adding different content types to the document and specifying the required properties. The default resolution and colorspace is also set in the single configuration.

*Basic single configuration, already filled with content and rendering*

**Edit W2P Single Configuration "Demo Single Config" on page "w2p"**

















**Hide:**

Demo Base Config ▾





























**Title**  
Demo Single Config

**Version**  
1.0

**Downloads**  
Create new

- Demo Single Config 300 cmyk 1-...     
- Demo Single Config 300 cmyk 1-...      
- Demo Single Config 300 cmyk 1-...     

**Contents**  
Create new

- Contents     
- Test      
- Commercetest      
- Pagetest      
- Pagetest2     

**Default Resolution**  
300dpi ▾

**Default Colorspace**  
CMYK ▾

*results (downloads)*

## Adding content to the single configuration

Content is added to a single configuration through the IRRE area “Contents”.

The extension already has some basic content rendering libraries providing the following types:

Content Type:	Description:
Page	Renders a TYPO3 page
PDF	Renders a given PDF document
Commerce	Renders commerce categories and products/articles
Table of Contents	Renders a table of contents page, each content type has a checkbox to activate/deactivate inclusion in the table of contents
Index	Renders index pages of commerce articles In the document

**Contents**  
Create new

Contents

**Hide:**

**Title**  
Contents

**Type**  
Table of Contents ▼

**Show Page Number**

**Header**  
Contents

**Show Header**

**Footer**  
Footer ToC

**Show Footer**

*Table of contents in single configuration*

**Contents**

Create new

Contents

Test

**Hide:**

**Title**  
Test

**Type**  
PDF

**File (PDF 300dpi)**  
PB-FR-4.pdf

\*  
-PHP -PHP3  
Datei auswählen Keine ausgewählt

**Page Range**

**Show Page Number**

**Show in table of contents**

**Header**  
Testheader

**Show Header**

**Footer**  
Testfooter

**Show Footer**

*PDF file as content to single configuration*

**Contents**

Create new

Contents ⓘ 📄 ⌵ 💡 🗑️

Test ⓘ 📄 ⬆️ ⬆️ 💡 🗑️

Commercetest ⓘ 📄 ⬆️ ⬆️ 💡 🗑️

**Hide:**

**Title**  
Commercetest

**Type**  
Commerce ▾

**Commerce Category**  
2

**Price Datafile**  
 ✕ 📁 Import Log

XLSX  
Datei auswählen Keine ausgewählt

**Show Page Number**

**Show in table of contents**

**Header**

**Show Page Number**

**Show in table of contents**

**Header**


**Show Header**




























**Footer**  
Footer Products

**Show Footer**

*Commerce categories/products/articles as content in single configuration*

**Contents**

 Create new

 Contents	    
 Test	     
 Commercetest	     
 Pagetest	     

**Hide:**

**Title**

**Type**

**Page ID**

**Show Page Number**

**Show in table of contents**

**Header**

**Show Header**

**Footer**

**Show Footer**

*TYPO3 page content in single configuration*

## Downloads in a single configuration

Each single configuration contains an IRRE area called “Downloads”. This is where all the finished renderings are present. The Downloads itself are read only and represent the settings of the finished rendering. The URL points to the rendered PDF document on the w2pservice rendering server.

**Downloads**

Create new

Demo Single Config 300 cmyk 1-...

Hide:

**Creation date**  
14:44 27-02-2014

**Title**  
Demo Single Config 300 cmyk 1-6.pdf

**URL**  
[http://w2posserver.connetation.at/output/1418895ad321717fcf6709f74c12177c\\_530f3cc2381db/Demo\\_Single\\_Config\\_300\\_cmyk\\_1-6.pdf](http://w2posserver.connetation.at/output/1418895ad321717fcf6709f74c12177c_530f3cc2381db/Demo_Single_Config_300_cmyk_1-6.pdf)

**Final Version**

**Default Resolution**  
300dpi

**Default Colorspace**  
CMYK

Demo Single Config 300 cmyk 1-...

*Download in single configuration (finished rendering)*



# Frontend

## Previewing content

Content preview is done in the frontend part of the extension.

After logging in using a previously created frontend (Web Site) user and assigning a group with appropriate rights, you can select the single configuration you want to preview.

After configuration selection the default values for resolution and colorspace are loaded and the first page is displayed.

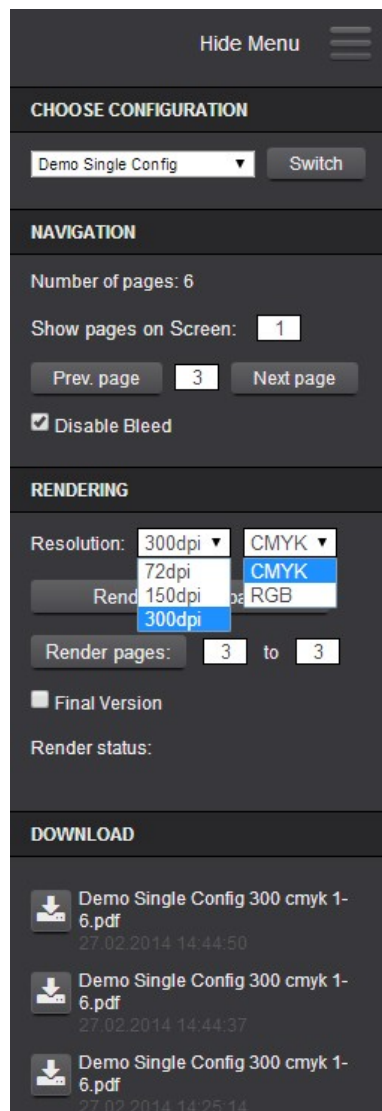
The navigation part allows navigation through pages as well as defining how much pages should be displayed at once.

Additionally it is possible to disable the bleed which was set in the base configuration, for example to render a RGB version for web view only.

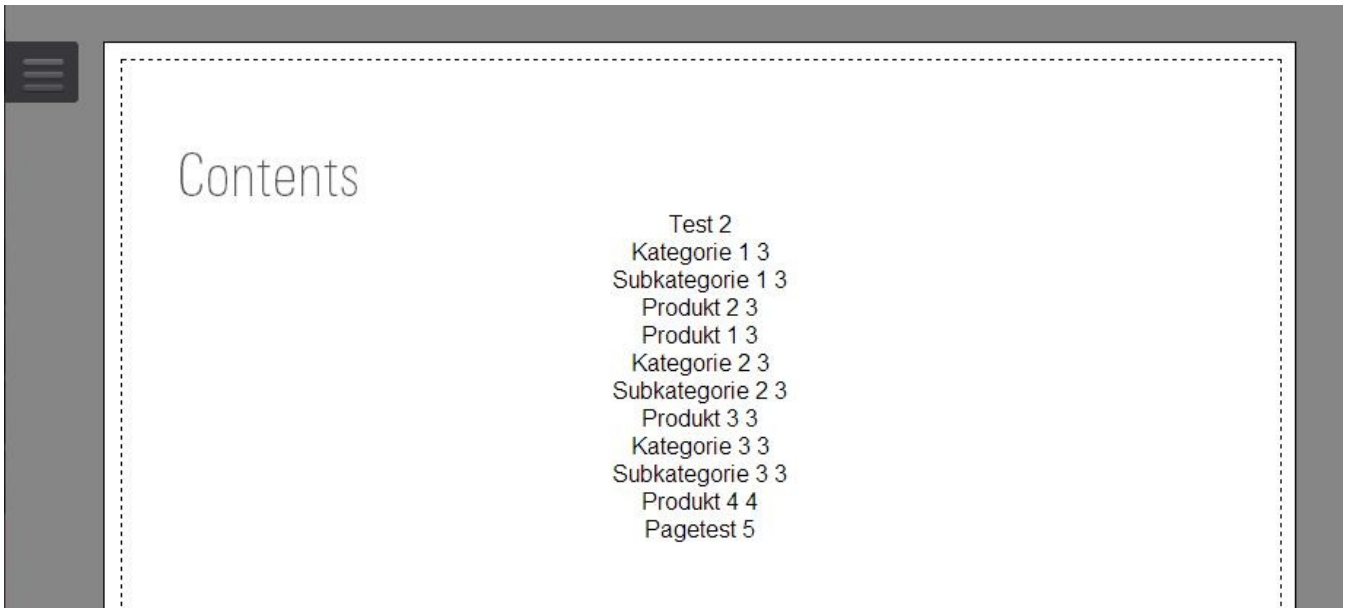
The rendering section defines settings for the rendering process, the most important being the resolution and colorspace of the resulting PDF file. By default the rendering is only in preview mode, where watermarks are placed in the document.

To render without watermarks the “Final Version” checkbox has to be set. In this mode special limits apply.

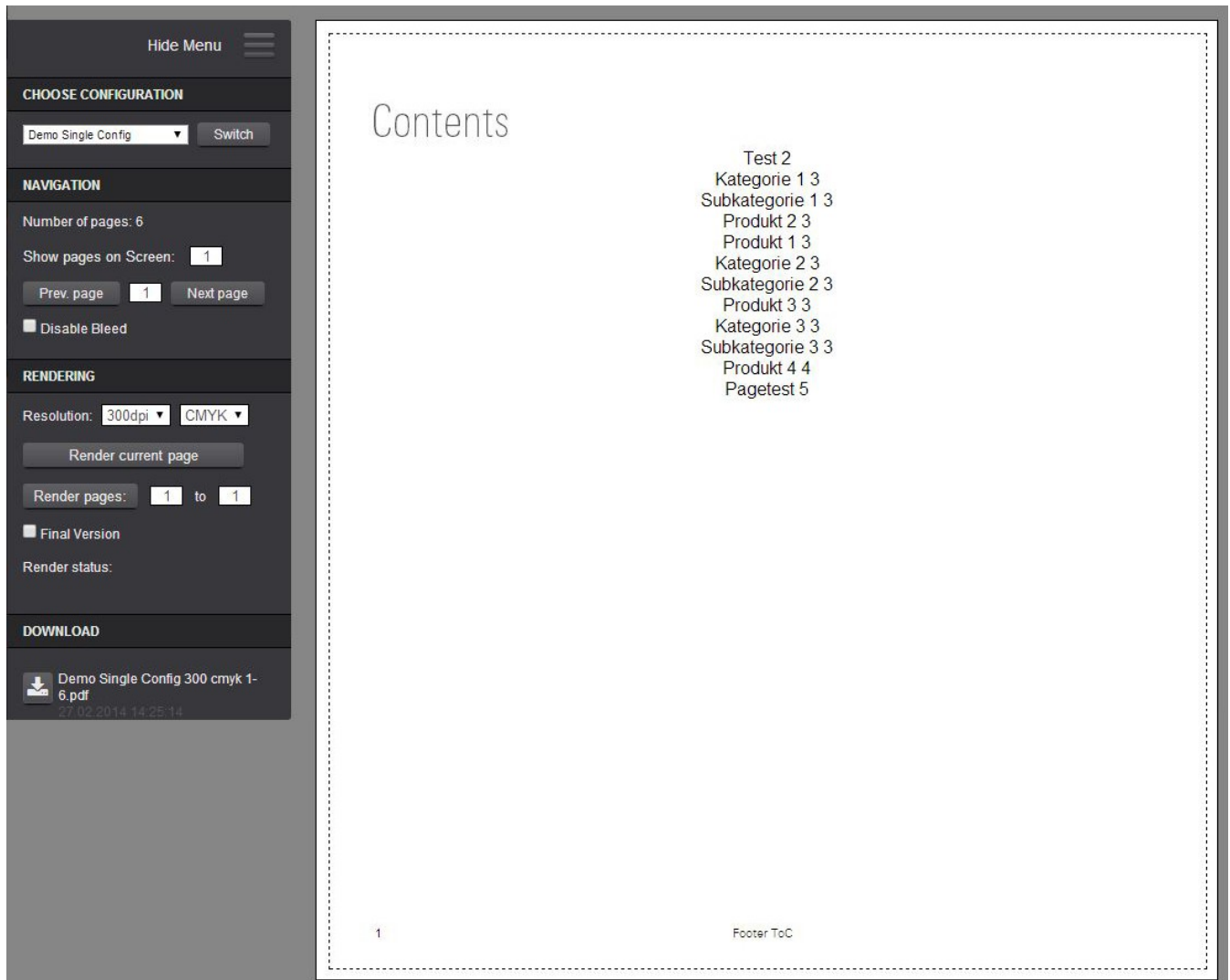
Beneath is a list of the last rendered documents.




*Web2Print rendering panel*





*Rendering panel may be hidden to provide better overview of content*



*Preview of table of contents in demo setup (lacking CSS)*

Hide Menu 

**CHOOSE CONFIGURATION**

Demo Single Config  

**NAVIGATION**

Number of pages: 6

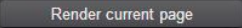
Show pages on Screen:

Prev. page  Next page

Disable Bleed

**RENDERING**

Resolution:





Render pages:  to


Final Version

Render status:

**DOWNLOAD**

 Demo Single Config 300 cmyk 1-6.pdf  
27.02.2014 14:44:50

 Demo Single Config 300 cmyk 1-6.pdf  
27.02.2014 14:44:37


 Demo Single Config 300 cmyk 1-6.pdf  
27.02.2014 14:25:14

**Commercetest**

Kategorie 1

Subkategorie 1

Produkt 2



Produkt 1

Kategorie 2

Subkategorie 2

Produkt 3

Kategorie 3

Subkategorie 3

*Preview of commerce content*

## Starting the rendering process

After checking the content in preview, the rendering process itself is started by selecting the desired resolution and colorspace settings and the page-range to render. If only the current page should be rendered a simple click on the button "Render current page" is sufficient.

If the document shall be printed later on, selecting "Final Version" is recommended to get a resulting document without watermarks. It should be mentioned, that for the demo license limits apply to rendering final versions of a document.

After invoking the rendering, continuous status queries through Javascript provide the current status to the user. The possible status values are:

Status:	Description
Starting rendering process	Job was added to queue and will be processed soon
1 of 5 / Downloading resources	Processing of sourceUrl and fetching all needed resources including linked CSS stylesheets and images
2 of 5 / HTML5 rendering	Rendering of HTML5 content to PDF is initiated
3 of 5 / CI-Color mapping	If a color conversion file was provided mapping of RGB to CMYK colors is executed (only if target colorspace is CMYK)
4 of 5 / RGB → CMYK color conversion	Conversion of remaining RGB parts in the document to CMYK colorspace (only if target colorspace is CMYK)
5 of 5 / Merging PDF-Template	Merging provided PDF files into resulting document
Download ready	Rendering process finished, resulting PDF can be downloaded
Authentication failed	The provided license key is invalid

# Extending web2print

To allow rendering of other than the already supported content types (page, pdf, commerce, toc, index) it is possible to extend the render capabilities with your own library.

First you have to add a statement similar to a hook statement in your `ext_localconf.php`. Additional fields in the `tx_web2print_content` table can also be specified.

The TCA for the new type has to be configured as well to define the visible fields.

The Interface to the web2print extension consists of function to populate the content to be render in a hierarchical structure as well as the real rendering which is based on a give template, attached to a base configuration.

The populate method is called for each encountered content element inside a single configuration.

After populating content into the structure the post-processing method is called for each content element. This is needed for example for the table of contents since the actual page order can only be determined after inserting the prepared content.

The rendering itself happens in `renderContentElement` and `renderPage` as well as in `renderToc` for the table of contents entry if valid.

## Render library interface

```
public function populate(&$document, $content, $singleConfig, &$pObj);  
  
public function postProcess(&$document, $content, $singleConfig, &$pObj);  
  
public function renderContentElement($ce, &$pObj);  
  
public function renderPage($page, $output, &$pObj);  
  
public function renderToc($contentElement, &$pObj);
```

# List of web2print add-ons

## Render library for pages

Creates the content for one page from a TYPO3 by parsing HTML and incorporating the content of the body tag and all stylesheets in the head into the w2p frontend view.

## Render library for PDF files

PDF files get submitted to the w2pservice and a preview of the inserted PDF page is also shown in the frontend view

## Render library for commerce categories/products/articles

Beginning with the given commerce master category the tree of categories and products with articles is traversed and each category/product is translated into a content element on a page. With preset size definitions a placement on the page is achieved, aided by text size measurement using gdlib ttf rendering.

## Render library for table of contents

All content elements inside the document are captured and if the flag for table of contents is set, get included in the rendered version.

The generated content elements include a reference to the actual content element, therefore it's easy to get the correct page for the rendering process.

## Render library for index page (commerce articles)

Simple index table based on all the commerce articles in the document.

# w2pservice

The connection to the w2pservice is established through SOAP (URL: <http://w2posserver.connetation.at/w2pservice.wsdl>).

The available methods are addJob and getStatus each accepting on parameter containing json encoding data.

The addJob method adds the rendering job to the queue and returns a unique has for this job for subsequent status queries.

The should be checked continuously, although there is also a callback which is called upon completion of the rendering.

```
function addJob($jsonData)
```

Available options for jsonData	
sourceUrl	Source URL for of the document to be rendered
licenseKey	Assigned license key for w2pservice
finalVersion	Render preview or final version (without watermarks) (0/1)
pageWidth	Width of a page in the document (unit mm)
pageHeight	Height of a page in the document (unit mm)
disableBleed	If set 1, no additional cropping marks are added on the rendered document (0/1)
finalFileName	Filename of the final rendered PDF file
colorSpace	Colorspace to be used in the rendering process (rgb/cmyk)
resolution	Resolution for the rendering process (image rendering) (72/150/300)
pdfPages	Array of PDF pages which are inserted on give page after rendering of regular content (options for each elements: pdffile, startpage, pagecount)
fileChecksumScript	URL to a script accepting a parameter filesrc resulting in the md5 hash of the given file. Used to minimize file transfers to the rendering server
callbackUrl	URL to a script handling callback after finished rendering, parameters are uuid (the unique id of the finished rendering) and finalVersion (indicating if the rendered document was a final version without watermarks)
colorConversionFile	URL of the color conversion xml file (conversion of RGB to CMYK colors)

JSON return value	
uuid	unique id generated based on sourceUrl and current rendering time, identified a unique rendering instance. Used for status queries.
messages	detailed debug messages about the rendering process
error	if an error occurred this value will be present

```
function getStatus($jsonData)
```

Available options for jsonData	
uuid	unique id, returned by the addJob method
sourceUrl	sourceUrl for the document, if the unique id is not known

JSON return value	
renderstatus	current status of the rendering process
messages	detailed debug messages about the rendering process
outputAddress	the final URL for the rendered PDF file
error	if an error occurred this value will be present

# Sponsoring web2print development

In the backend as well as in the frontend part of the web2print extension there are donate buttons for donations via Paypal.

## Known problems

Sometimes the AJAX calls to get the status fail. Since there is also a callback when the rendering job is finished, the PDF document should finally be available in the frontend or backend view as a "Download".

## To-Do

## Changelog

Initial release