
Produce & Publish Plone Client Connector Documentation

Release 2.1

Andreas Jung

February 17, 2015

1	Documentation	3
2	Source code	5
3	Bugtracker	7
4	Licence	9
5	Author	11
5.1	Installation	11
5.2	Resource directories	13
5.3	Adding custom content-types to the Plone Client Connector	14
5.4	Integration with PloneFormGen	15
5.5	Changelog	15
6	Indices and tables	25

The Produce & Publish Plone Client connector integrates the Plone CMS with the Produce & Publishing platform and supports the generation of PDF, EPUB and other office formats (depending on the installed external converters on the Produce & Publish server side).

Documentation

Primary documentation: <http://docs.produce-and-publish.com>

Produce & Publish website: <http://www.produce-and-publish.com>

Source code

See <https://github.com/zopyx/zopyx.smartprintng.plone>

Bugtracker

See <https://github.com/zopyx/zopyx.smartprintng.plone/issues>

Licence

Published under the GNU Public Licence Version 2 (GPL 2)

ZOPYX Limited
Charlottenstr. 37/1
D-72070 Tuebingen, Germany
info@zopyx.com
www.zopyx.com

Contents:

5.1 Installation

This documentation assumes that your installation of Plone/Zope is based on `zc.buildout`.

- edit your *buildout.cfg* - add *zopyx.smartprintng.plone* to the **eggs** options of your buildout.cfg:

```
# For Plone 3.x (Client Connector < 2.0)
find-links = ...
    http://username:password@sdist.zopyx.com

# For Plone 4.x (Client Connector >2.0)
find-links = ...
    http://username:password@sdist-pp.zopyx.com

eggs = ...
    zopyx.smartprintng.plone

# only needed for Plone 3.x
zcml = ...
    zopyx.smartprintng.plone
```

- in addition (and effective 17.08.2011 or later) you need to pin the version of the `BeautifulSoup` module:

```
[buildout]
versions = versions
...

[versions]
BeautifulSoup = 3.2.0
...
```

- re-run *bin/buildout*
- restart Zope/Plone
- When running the Produce & Publish server on a different server, you must adjust the SMARTPRINTNG_SERVER environment variables within your *.bashrc* file (or a similar file) or you put those variables into your buildout configuration using the *<environment>* section. Username and password are only needed when you run the Produce & Publish server behind a reverse proxy (requiring authentication).

```
export SMARTPRINTNG_SERVER=http://user:password@your.server:6543/
```

or

```
<environment>
    SMARTPRINTNG_SERVER=http://user:password@your.server:6543/
</environment>
```

5.1.1 Supported Plone content-types

- Document
- Folder (nested structure)
- News item
- PloneGlossary
- Collection

5.1.2 Usage

The Plone connector provides a dedicated @@asPlainPDF view that can be added to the URL of any of the supported content-types of Plone (Document, Folder, Newsitem, PloneGlossary). So when your document is for example associated with the URL

```
http://your.server/plone/my-page
```

you can generate a PDF by using the URL

```
http://your.server/plone/my-page/@@asPlainPDF
```

5.1.3 Parameters

The @@asPlainPDF view accepts the following parameters controlling certain aspects of the PDF conversion:

- **language** - can be set to 'de', 'en', 'fr' etc. in order to control language-specific aspects of the PDF conversion. Most important: this parameter controls the hyphenation. The Plone connector comes out-of-the-box with hyphenation tables for several languages. You can omit this URL parameter if the **Language** metadata parameter (of the top-level document) to be converted is set within Plone.
- **converter** - if you are using the Produce & Publish server with a converter backend other than PrinceXML you can specify a different name (default is *pdf-prince*). See *zopyx.convert2* documentation for details.
- **resource** - can be set in order to specify a registered resource directory to be used for running the conversion. The `resource` parameter must be identical with the `name` parameter of the related ZCML `<smartprintng:resourceDirectory>` directive.

- **template** - can be used to specify the name of template to be used for running the conversion. The `template` parameter usually refers to a `.pt` filename inside the `resource` directory.

5.1.4 Miscellaneous

You may set the `SMARTPRINTNG_LOCAL_CONVERSION` environment variable (to some value) in order to run the conversion locally inside the Plone process without using an external Produce & Publish server.

The environment variable `SMARTPRINTNG_ZIP_OUTPUT` can be set to export all resources used for the conversion into a ZIP file for debugging purposes. The path of the generated ZIP file is logged within the standard Zope/Plone logfile (or the console if Plone is running in foreground).

5.2 Resource directories

The Plone Client connector allows you to define your own resource directories containing

- the PDF main template
- style sheets
- font files
- hyphenation files

5.2.1 Registering your own resource directory

First you need your own policy - e.g. *zopyx.theme*. Inside the `configure.zcml` file of your *zopyx.theme* you need to register a sub-directory using the `smartprintng:resourceDirectory` directive:

```
<configure
  xmlns="http://namespaces.zope.org/zope"
  xmlns:zcml="http://namespaces.zope.org/zcml"
  xmlns:smartprintng="http://namespaces.zopyx.com/smartprintng"
>

  <smartprintng:resourceDirectory
    name="zopyx_resource"
    directory="resources_pdf"
  />

</configure>
```

The registered `resources_pdf` directory must contain all resource files as flat structure (no sub-directories). The `name` parameter relates to the optional `resource URL` parameter as use for the `@asPlainPDF` browser view.

5.2.2 Naming conventions

- PDF template: `.pt`
- Stylesheets: `.css`, `.styles`
- Images: `.gif`, `.jpg`, `.png`
- Hyphenation files: `.hyp`
- Coverpage templates (only used with Authoring Environment): `.cover`

- Font files: .otf, .ttf

5.3 Adding custom content-types to the Plone Client Connector

This documentation explains how to extend the Plone Client Connector with your own or custom Plone content-types.

Custom content-types can be registered with the Produce & Publish server using the Zope Component Architecture. The one single contact of the P&P server with a content-type is the existence of a `@@asHTML` view for the related content-type. The `@@asHTML` view must return a HTML snippet that will be used by the P&P within the main body of its own rendering PDF template.

As an example look at the `@@asHTML` view for Plone news items.

The `@@asHTML` view is configured through ZCML (within your `configure.zcml` file):

```
<browser:page
  name="asHTML"
  for="Products.ATContentTypes.interface.news.IATNewsItem"
  permission="zope2.View"
  class=".newsitem.HTMLView"
/>
```

and implemented as browser view (`newsitem.py`):

```
from Globals import InitializeClass
from Products.Five.browser import BrowserView
from Products.Five.browser.pagetemplatefile import ViewPageTemplateFile

class HTMLView(BrowserView):
    """ This view renders a HTML fragment for the configured content type """

    template = ViewPageTemplateFile('newsitem_raw.pt')

    def __call__(self, *args, **kw):
        return self.template(self.context)

InitializeClass(HTMLView)
```

The related templates renders a snippet of code for a news item object:

```
<div class="type-newsitem document-body">
  <h1 class="title bookmark-title" tal:content="context/Title" />
  <div class="description" tal:content="context/Description" />
  <div>
    <div class="image-box" tal:condition="nocall: context/image | nothing">
      
      <div class="image-caption" tal:content="context/getImageCaption | nothing" />
    </div>

    <div class="body" tal:content="structure context/getText" />
  </div>
</div>
```

In addition your content-type implementation **must** provide the `zopyx.smartprinting.plone.interfaces.IPPContent` interface - either by specifying this interface as part of the class definition in your code

```
class MyContentType(...):

    implements(IPPContent)
```

or you add the interfaces as a marker interface through ZCML

```
<five:implements
    class="my.package.contents.mytype.MyContentType"
    interface="zopyx.smartprintng.plone.interfaces.IPPContent"
/>
```

Only content objects providing the `IPPContent` interface are being considered during the aggregation phase of the Plone Client Connector.

For further example code, please refer to the `zopyx/smartprintng/plone/browser` directory. The `folder` integration (*folder.py*) shows you a more complex example and involves aggregation of other content.

5.4 Integration with PloneFormGen

Using Produce & Publish with PloneFormGen - generating PDF documents from form data.

5.4.1 Installation

- Install **PloneFormGen**

5.4.2 Converting form data to PDF

- use a script adapter with the following code

```
view = context.restrictedTraverse('@@asPFGPDF')
pdf = view()
R = context.REQUEST.RESPONSE
R.setHeader('content-type', 'application/pdf')
R.setHeader('content-length', len(pdf))
R.setHeader('content-disposition', 'attachment; filename=%s.pdf' % context.getId())
R.write(pdf)
```

- You can access PFG form values within your PDF template using (in this case we have a form parameter `fullname`)

```
<span tal:replace="options/request/fullname | nothing">Fullname</span>
```

5.5 Changelog

5.5.1 2.1.26 (2012-08-24)

- some fixes/workaround for exporting (embedded) images from special types like News Item

5.5.2 2.1.25 (2012-08-14)

- added `.mode-flat` and `mode-nested` CSS classes depending on the folder aggregation mode in order to fix a problem with the `makeImagesLocal` transformation for nested folders

5.5.3 2.1.23 (2012-05-08)

- fix in image resolver

5.5.4 2.1.22 (2012-04-04)

- locale aware sorting of index terms in addIndex()

5.5.5 2.1.21 (2012-04-03)

- fixed bug in endnote number handling in convertEndNotes()

5.5.6 2.1.20 (2012-02-26)

- fixed CSS counter bug (related to PrinceXML 7.1 vs. 8.0)

5.5.7 2.1.19 (2012-02-20)

- unicode fix in HTML splitter code

5.5.8 2.1.18 (2012-02-10)

- folder aggregator now support (experimental) document filter based on UIDs (filter_uid parameter)
- makeImageLocal() transformation removed parts of the document while replacing the old image with an image container
- added experimental addIndexList transformation

5.5.9 2.1.17 (2011-12-28)

- checking html input in transformation machinery for empty strings

5.5.10 2.1.16 (2011-12-21)

- better footnote handling

5.5.11 2.1.15 (2011-12-20)

- convertWordEndnotes() transformation added

5.5.12 2.1.14 (2011-12-19)

- fixed image width 100% for images inside an image-container (PrinceXML 8 compatibility)

5.5.13 2.1.13 (2011-12-18)

- some fixes discovered using PyFlakes

5.5.14 2.1.12 (2011-12-12)

- added some transformations for better Word import

5.5.15 2.1.11 (2011-11-23)

- update trove classifiers

5.5.16 2.1.10 (2011-11-17)

- improved aggregator for nested content folders
- support for content hierarchies up to level 8
- support for new environment variable SMARTPRINTNG_ZIP_OUTPUT

5.5.17 2.1.9 (2011-11-11)

- fixed bug in makeImagesLocal() transformation where the document root has not been used properly for finding images by traversal

5.5.18 2.1.8 (2011-11-11)

- support for new SMARTPRINTNG_LOCAL_CONVERSION environment variable

5.5.19 2.1.7 (2011-11-08)

- removed some baggage in order to make distro smaller

5.5.20 2.1.6 (2011-11-07)

- minor fixes in handling of generated files for download

5.5.21 2.1.5 (2011-11-07)

- first public (open-source) release

5.5.22 2.1.4 (2011-10-25)

- fixed unicode/utf-8 issue in makeImagesLocal transformation

5.5.23 2.1.3 (2011-10-14)

- added fixAmpersand transformation

5.5.24 2.1.2 (2011-10-10)

- transformations for dealing with converted footnotes from Word

5.5.25 2.1.1 (2011-10-08)

- compatibility with Dexterity

5.5.26 2.1.0 (2011-09-22)

- final 2.1.0 release

5.5.27 2.0.9 (2011-09-20)

- fixed bug in xpath_query() (using relative query)

5.5.28 2.0.8 (2011-09-11)

- more cleanup

5.5.29 2.0.7 (2011-09-10)

- some ZCML fixes in order to avoid Plone 4.x startup failures under some conditions
- restored compatibility with Plone 3.3.X

5.5.30 2.0.6 (2011-09-08)

- image exporter did not deal proper with duplicate image ids
- minor fixes

5.5.31 2.0.5 (2011-09-02)

- new lxml backed transformation pipeline
- more tests

5.5.32 2.0.4 (2011-08-26)

- logging resource registration using INFO severity
- new lxml dependency

5.5.33 2.0.3 (2011/08/15)

- catching HTTPError in image resolver
- fixed another BeautifulSoup misbehaviour in fixHeadingAfterOfficeImport()

5.5.34 2.0.2 (2011-08-02)

- minor fix

5.5.35 2.0.1 (2011-08-02)

- integration with new zip client version (Proxy2 implementation)

5.5.36 2.0.0 (2011-07-25)

- final release

5.5.37 2.0.0rc2 (2011-07-04)

- fix in logger call in folder.py

5.5.38 2.0.0rc1 (2011-07-01)

- don't extend images an authoring project
- remove class attributes from headings after office import
- added ignoreHeadingsForStructure transformation

5.5.39 2.0.0b2 (2011-06-16)

- minor fixes related to office data import

5.5.40 2.0.0b1 (2011-05-24)

- fixes related to office format input

5.5.41 2.0.0a3 (2011-05-17)

- added some workaround for image resolver in order to deal with images referenced through a fully specified URL with a redirection included (TQM issue)

5.5.42 2.0.0a2 (2011-05-14)

- minor fix in safe_get()

5.5.43 2.0.0a1 (2011-05-10)

- simplification and refactoring

5.5.44 0.7.0 (2011-02-11)

- updated for use with zopyx.authoring 1.5.X
- added GenericDownloadView aka '@@ppConvert'
- exported images now contain a proper extension (fixes issues with the XFC converter depending on extension for determining the image format)

5.5.45 0.6.24 (2010-12-09)

- added addDocumentLinks() transformation
- including content ids of aggregated content

5.5.46 0.6.23 (2010-09-10)

- addImageCaptionsInHTML(): honour excludeFromImageEnumeration

5.5.47 0.6.22 (2010-09-09)

- fixed improper stripping of image names using an image scale (causing issues in the consolidated HTML view of the authoring environment)

5.5.48 0.6.21 (2010-08-09)

- added support '++resource++' image references (Patrick Gerken)
- added support for FSImage (Patrick Gerken)

5.5.49 0.6.20 (2010-08-05)

- added 'removeComments' transformation
- added 'makeImageSrcLocal' transformation

5.5.50 0.6.19 (2010-07-13)

- fixed race condition in makeImagesLocal()

5.5.51 0.6.18 (2010-06-14)

- images got a new PDF conversion option "Exclude from image enumeration"

5.5.52 0.6.17 (2010-06-12)

- inserting H1 title for consolidated HTML
- added extra class to folder title for consolidated HTML

5.5.53 0.6.16 (2010-05-29)

- inserting space for found anchors

5.5.54 0.6.15 (2010-04-15)

- minor fix in image handling

5.5.55 0.6.14 (2010-04-14)

- minor tweaks for image caption markup

5.5.56 0.6.13 (2010-03-26)

- support for `span.footnoteText`

5.5.57 0.6.12 (2010-03-21)

- support for image urls `'resolveuid/<uid>'`
- minor fixes and tweaking in image handling (caption generation)

5.5.58 0.6.11 (2010-03-10)

- added document extender
- document option for suppressing the title in PDF
- image caption support
- changed default transformations (to `makeImagesLocal` only)
- removed TOC from default PDF template

5.5.59 0.6.10 (2010-03-03)

- support for `request/transformations` parameter
- various fixes

5.5.60 0.6.9 (2010-02-22)

- added `<em>[[text:footnote-text]]</em>` support for generating footnotes
- various changes related to `zopyx.authoring` integration

5.5.61 0.6.8 (2010-02-03)

- Folder aggregation now works with all folderish objects providing IATFolder

5.5.62 0.6.7 (2009-11-30)

- makeImagesLocal: better dealing with virtual hosting

5.5.63 0.6.6 (2009-11-15)

- fixed CSS issue with TOC markup

5.5.64 0.6.5 (2009-11-12)

- always use images in their original resolution
- optional content information with link to the edit mode of the aggregated document (you must change the visibility of the .content-info class through CSS)
- a request parameter 'show-debug-info' will enable the additional content-info view
- better error handling
- better logging
- tweaked markup of generated TOC

5.5.65 0.6.3 (2009-10-27)

- refactored language handling
- refactored PDF view in order to provide a low-level view returning a reference to the generated PDF file instead providing it for HTTP download

5.5.66 0.6.2 (2009-10-24)

- setting anti-cache headers
- locale-aware sorting in PloneGlossary code

5.5.67 0.6.1 (2009-10-23)

- PloneGlossary integration: compare title case-insensitive (IDG project)

5.5.68 0.6.0 (2009-10-21)

- refactored and simplified transformation machinery

5.5.69 0.5.0 (2009-10-09)

- major rewrite

5.5.70 0.3.0 (2009-09-24)

- refactored views

5.5.71 0.2.0 (2009-09-23)

- more hyphenation dicts
- restructured resources directory

5.5.72 0.1 (xxxx-xx-xx)

- Initial release

Indices and tables

*/ genindex * modindex * search*