
zopyx.convert2 Documentation

Release 2.4

Andreas Jung

Sep 27, 2017

Contents

1	zopyx.convert2	3
1.1	Requirements	3
1.2	Installation	3
1.3	Supported platforms	4
1.4	Source code	4
1.5	Bug tracker	4
1.6	Usage	4
1.7	How zopyx.convert2 works internally	5
1.8	Environment variables	5
1.9	Known issues	5
1.10	Author	5
1.11	License	5
1.12	Contact	5
2	Changelog	7
2.1	2.4.5 (2012-11-05)	7
2.2	2.4.4 (2012-11-05)	7
2.3	2.4.3 (2012-06-20)	7
2.4	2.4.2 (2012-01-01)	7
2.5	2.4.1 (2011-11-11)	7
2.6	2.4 (2011-11-07)	8
2.7	2.3.2 (2011-08-23)	8
2.8	2.3.1 (2011-06-15)	8
2.9	2.3.0 (2011-06-05)	8
2.10	2.2.5 (2011-04-03)	8
2.11	2.2.4 (2010-12-16)	8
2.12	2.2.3 (2010-08-16)	8
2.13	2.2.2 (2010-08-15)	8
2.14	2.2.1 (2010-07-19)	8
2.15	2.2.0 (2010-05-15)	9
2.16	2.1.1 (2010-02-19)	9
2.17	2.1.0 (2009-09-05)	9
2.18	2.0.4 (2009-07-07)	9
2.19	2.0.3 (2009-07-05)	9
2.20	2.0.2 (2009-06-02)	9
2.21	2.0.1 (2009-06-02)	9

2.22	2.0.0 (2009-05-14)	9
2.23	2.0.0b3 (25.12.2008)	9
2.24	2.0.0b2 (05.10.2008)	10
2.25	2.0.0b1 (04.10.2008)	10
3	Indices and tables	11

The `zopyx.convert2` package helps you to convert HTML to PDF, EPUBV, RTF, ODT, DOCX and WML using XSL-FO technology or using PrinceXML. This package is used as the low-level API Produce & Publish Server (`zopyx.smartprintng.server`, <http://pypi.python.org/pypi/zopyx.smartprintng.server>)

Contents:

The `zopyx.convert2` package helps you to convert HTML to PDF, RTF, ODT, DOCX and WML using XSL-FO technology or using PrinceXML. This package is used as the low-level API for `zopyx.smartprintng.core`.

Requirements

- Java 1.5.0 or higher (FOP 0.94 requires Java 1.6 or higher)
- `csstoxslfo` (included)
- `XFC-4.0` (XMLMind) for ODT, RTF, DOCX and WML support (if needed)
- `XINC 2.0` (Lunasil) for PDF support (commercial)
- or `FOP 0.94` (Apache project) for PDF support (free)
- or `PrinceXML` (commercial) for PDF support

Installation

- install `zopyx.convert2` either using `easy_install` or by downloading the sources from the Python Cheeseshop. This will install automatically the BeautifulSoup and Elementtree modules if necessary.
- the environment variable `$XFC_DIR` must be set and point to the root of your XFC installation directory
- the environment variable `$XINC_HOME` must be set and point to the root of your XINC installation directory
- the environment variable `$FOP_HOME` must be set and point to the root of your FOP installation directory
- the 'prince' binary must be in the `$PATH` if you are using PrinceXML

Supported platforms

Windows, Unix

Source code

- <https://github.com/zopyx/zopyx.convert2>

Bug tracker

- <https://github.com/zopyx/zopyx.convert2/issues>

Usage

Some examples from the Python command-line:

```
from zopyx.convert2 import Converter
C = Converter('/path/to/some/file.html')
pdf_filename = C('pdf-xinc')['output_filename']      # using XINC
pdf2_filename = C('pdf-pisa')['output_filename']     # using PISA
pdf3_filename = C('pdf-fop')['output_filename']      # using FOP
pdf4_filename = C('pdf-prince')['output_filename']   # using FOP
rtf_filename = C('rtf-xfc')['output_filename']
pdt_filename = C('odt-xfc')['output_filename']
wml_filename = C('wml-xfc')['output_filename']
docx_filename = C('docx-xfc')['output_filename']
```

A very simple command-line converter is also available:

```
html-convert --format rtf --output foo.rtf sample.html
```

`html-convert` has a `--test` option that will convert some sample HTML. If everything is ok then you should see something like that:

```
>html-convert --test
Entering testmode
pdf: /tmp/tmpuOb37m.html -> /tmp/tmpuOb37m.pdf
rtf: /tmp/tmpuOb37m.html -> /tmp/tmpuOb37m.rtf
docx: /tmp/tmpuOb37m.html -> /tmp/tmpuOb37m.docx
odt: /tmp/tmpuOb37m.html -> /tmp/tmpuOb37m.odt
wml: /tmp/tmpuOb37m.html -> /tmp/tmpuOb37m.wml
pdf: /tmp/tmpZ6PGo9.html -> /tmp/tmpZ6PGo9.pdf
rtf: /tmp/tmpZ6PGo9.html -> /tmp/tmpZ6PGo9.rtf
docx: /tmp/tmpZ6PGo9.html -> /tmp/tmpZ6PGo9.docx
odt: /tmp/tmpZ6PGo9.html -> /tmp/tmpZ6PGo9.odt
wml: /tmp/tmpZ6PGo9.html -> /tmp/tmpZ6PGo9.wml
```

How zopyx.convert2 works internally

- The source HTML file is converted to XHTML using mxTidy
- the XHTML file is converted to FO using the great “csstoxslfo” converter written by Werner Donne.
- the FO file is passed either to the external XINC or XFC converter to generate the desired output format
- all converters are based on Java technology making the conversion solution highly portable across operating systems (including Windows)

Environment variables

The following environment variables can be used to resolve OS or distribution specific problems:

`ZOPYX_CONVERT_SHELL` - defaults to `sh` and is used to as shell command to execute external converters

`ZOPYX_CONVERT_EXECUTION_MODE` - default to `process` and refers to the method of Python executing external command (by default using the `process` module. Other value: `system`, `commands`)

Known issues

- If you are using zopyx.convert2 together with FOP: use the latest FOP 0.94 only. Don't use any packaged FOP version like the one from MacPorts which is known to be broken.
- Ensure that you have read the `csstoxslfo` documentation. `csstoxslfo` has several requirements about the HTML markup. Don't expect that it is the ultimate HTML converter. Any questions regarding the necessary markup are documented in the `csstoxslfo` documentation and will not be answered.

Author

`zopyx.convert2` was written by Andreas Jung for ZOPYX Ltd., Tuebingen, Germany.

License

`zopyx.convert2` is published under the Zope Public License (ZPL 2.1). See `LICENSE.txt`.

Contact

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

2.4.5 (2012-11-05)

- fixed typo

2.4.4 (2012-11-05)

- creating tidied file inside the existing folder instead of in \$TMPDIR. This error caused that some style files could not be loaded with PDFreactor

2.4.3 (2012-06-20)

- fixed logger (mis-)usage
- fixed API documentation

2.4.2 (2012-01-01)

- experimental support for PDFreactor

2.4.1 (2011-11-11)

- fixed BeautifulSoup dependency

2.4 (2011-11-07)

- documentation updated in order to reflect changes to the first public release of the Plone Client Connector

2.3.2 (2011-08-23)

- added support for `%(WORKDIR)s` substitution in Calibre converter

2.3.1 (2011-06-15)

- support for PISA (pdf-pisa-bin) - requires that 'pisa' is found in the \$PATH

2.3.0 (2011-06-05)

- support for PISA (pdf-pisa)

2.2.5 (2011-04-03)

- calibre converter now honors the `commandlineoptions.txt` file

2.2.4 (2010-12-16)

- stripping of BASE tag for XFC-based converters

2.2.3 (2010-08-16)

- made stripping of the BASE tag specific to pdf-fop

2.2.2 (2010-08-15)

- pdf-fop converter not registered properly

2.2.1 (2010-07-19)

- support `$ZOPYX_CONVERT_SHELL`

2.2.0 (2010-05-15)

- dedicated ConversionError exception added

2.1.1 (2010-02-19)

- relaxed tidy check for existence of images

2.1.0 (2009-09-05)

- Calibre integration
- API change: convert() now returns a richer dict with all related conversion results

2.0.4 (2009-07-07)

- pinned BeautifulSoup 3.0.x

2.0.3 (2009-07-05)

- fix in fop.py

2.0.2 (2009-06-02)

- fixed broken path for test data files

2.0.1 (2009-06-02)

- added environment variable ZOPYX_CONVERT_EXECUTE_METHOD to control the usage of the process module vs. os.system() (in case of hanging Java processes). Possible values: 'process' (default), 'system'

2.0.0 (2009-05-14)

- final release

2.0.0b3 (25.12.2008)

- tidy: rewrite image references relative to the html file to be converted

2.0.0b2 (05.10.2008)

- fixed some import errors
- now working with `zopyx.smartprintng.core`

2.0.0b1 (04.10.2008)

- initial release
- complete new reimplementation of `zopyx.convert`
- added support for PrinceXML

CHAPTER 3

Indices and tables

- `genindex`
- `modindex`
- `search`