
pyhwp Documentation

Release 0.1b16.dev0

mete0r

Apr 09, 2023

CONTENTS

1	pyhwp	3
1.1	Features	3
1.2	Installation	3
1.3	Requirements	3
1.4	Documentation & Development	3
1.5	Contributors	4
1.6	License	4
1.7	Disclosure	4
2	hwp5proc: HWPv5 processor	5
2.1	Named Arguments	5
3	Subcommands	7
3.1	version	7
3.2	header	7
3.3	summaryinfo	7
3.4	ls	8
3.5	cat	8
3.6	unpack	9
3.7	records	10
3.8	models	11
3.9	find	12
3.10	xml	12
3.11	rawunz	13
3.12	diststream	13
4	Converters (<i>Experimental</i>)	15
4.1	Requirements	15
4.2	hwp5odt: ODT conversion	15
4.3	hwp5html: HTML conversion	16
4.4	hwp5txt: text conversion	17
5	Hacking Guide	19
5.1	Setup development environment	19
5.2	Directory Layout	20
5.3	Hack & Test	21
6	CHANGES	23
6.1	0.1b16 (unreleased)	23
6.2	0.1b15 (2020-05-30)	23
6.3	0.1b14 (2020-05-17)	23
6.4	0.1b13 (2020-05-17)	23
6.5	0.1b12 (2019-04-08)	23
6.6	0.1b11 (2019-03-21)	24
6.7	0.1b10 (2019-03-21)	24

6.8	0.1b9 (2016-02-26)	24
6.9	0.1b8 (2014-11-03)	24
6.10	0.1b7 (2014-01-31)	24
6.11	0.1b6 (2014-01-20)	25
6.12	0.1b5 (2013-10-29)	25
6.13	0.1b4 (2013-07-03)	25
6.14	0.1b3 (2013-06-18)	25
6.15	0.1b2 (2013-06-08)	25
7	Indices and tables	27

Contents:

HWP Document Format v5 parser & processor.

1.1 Features

- Analyze and extract internal streams out from a HWP Document Format v5 file
- (*Experimental*) Conversion to OpenDocument format (.odt) or plain text (.txt)

1.2 Installation

from pypi:

```
virtualenv pyhwp
pyhwp/bin/pip install --pre pyhwp # Install pyhwp into a virtualenv directory
```

Or:

```
pip install --user --pre pyhwp # Install pyhwp into user's home directory
```

1.3 Requirements

- Python 2.7, 3.5, 3.6, 3.7 or 3.8
- cryptography
- lxml
- olefile

1.4 Documentation & Development

- Documentation: <https://pyhwp.readthedocs.io> [한국/조선어]
- Distribution: <https://pypi.org/project/pyhwp/>
- Development: <https://github.com/mete0r/pyhwp>
- Issue tracker: <https://github.com/mete0r/pyhwp/issues>
- Feedbacks & contributions are welcome!

1.5 Contributors

Maintainer: [mete0r](#)

1.6 License

Copyright (C) 2010-2023 [mete0r](#) <<https://github.com/mete0r>>



GNU Affero General Public License v3.0 (text version)

This program is free software: you can redistribute it and/or modify it under the terms of the GNU Affero General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Affero General Public License for more details.

You should have received a copy of the GNU Affero General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

1.7 Disclosure

This program has been developed in accordance with a public document named “HWP Binary Specification 1.1” published by [Hancom Inc.](#)

HWP5PROC: HWPV5 PROCESSOR

Do various operations on HWPv5 files.

```
usage: hwp5proc [-h] [--loglevel LOGLEVEL] [--logfile LOGFILE]
               {version,header,summaryinfo,ls,cat,unpack,records,models,find,xml,
↪rawunz,diststream}
               ...
```

2.1 Named Arguments

--loglevel	Set log level.
--logfile	Set log file.

SUBCOMMANDS

3.1 version

Print the file format version of .hwp files.

Print the file format version of <hwp5file>.

```
usage: hwp5proc version [-h] <hwp5file>
```

3.1.1 Positional Arguments

<hwp5file> .hwp file to analyze

3.2 header

Print file headers of .hwp files.

Print the file header of <hwp5file>.

```
usage: hwp5proc header [-h] <hwp5file>
```

3.2.1 Positional Arguments

<hwp5file> .hwp file to analyze

3.3 summaryinfo

Print summary informations of .hwp files.

Print the summary information of <hwp5file>.

```
usage: hwp5proc summaryinfo [-h] <hwp5file>
```

3.3.1 Positional Arguments

<hwp5file> .hwp file to analyze

3.4 ls

List streams in .hwp files.

List streams in the <hwp5file>.

```
usage: hwp5proc ls [-h] [--vstreams | --ole] <hwp5file>
```

3.4.1 Positional Arguments

<hwp5file> .hwp file to analyze

3.4.2 Named Arguments

--vstreams Process with virtual streams (i.e. parsed/converted form of real streams)
Default: False

--ole Treat <hwp5file> as an OLE Compound File. As a result, some streams will be presented as-is. (i.e. not decompressed)
Default: False

3.5 cat

Extract out internal streams of .hwp files

Extract out the specified stream in the <hwp5file> to the standard output.

```
usage: hwp5proc cat [-h] [--vstreams | --ole] <hwp5file> <stream>
```

3.5.1 Positional Arguments

<hwp5file> .hwp file to analyze

<stream> Internal path of a stream to extract

3.5.2 Named Arguments

--vstreams Process with virtual streams (i.e. parsed/converted form of real streams)
Default: False

--ole Treat <hwp5file> as an OLE Compound File. As a result, some streams will be presented as-is. (i.e. not decompressed)
Default: False

Example:

```
$ hwp5proc cat samples/sample-5017.hwp BinData/BIN0002.jpg | file -
$ hwp5proc cat samples/sample-5017.hwp BinData/BIN0002.jpg > BIN0002.jpg
$ hwp5proc cat samples/sample-5017.hwp PrvText | iconv -f utf-16le -t utf-8
$ hwp5proc cat --vstreams samples/sample-5017.hwp PrvText.utf8
$ hwp5proc cat --vstreams samples/sample-5017.hwp FileHeader.txt

ccl: 0
cert_drm: 0
cert_encrypted: 0
cert_signature_extra: 0
cert_signed: 0
compressed: 1
distributable: 0
drm: 0
history: 0
password: 0
script: 0
signature: HWP Document File
version: 5.0.1.7
xmltemplate_storage: 0
```

3.6 unpack

Extract out internal streams of .hwp files into a directory.

Extract out streams in the specified <hwp5file> to a directory.

```
usage: hwp5proc unpack [-h] [--vstreams | --ole] <hwp5file> [<out-directory>]
```

3.6.1 Positional Arguments

<hwp5file>	.hwp file to analyze
<out-directory>	Output directory

3.6.2 Named Arguments

--vstreams	Process with virtual streams (i.e. parsed/converted form of real streams)
	Default: False
--ole	Treat <hwp5file> as an OLE Compound File. As a result, some streams will be presented as-is. (i.e. not decompressed)
	Default: False

Example:

```
$ hwp5proc unpack samples/sample-5017.hwp
$ ls sample-5017
```

Example:

```
$ hwp5proc unpack --vstreams samples/sample-5017.hwp
$ cat sample-5017/PrvText.utf8
```

3.7 records

Print the record structure of .hwp file record streams.

Print the record structure of the specified stream.

```
usage: hwp5proc records [-h]
                        [--simple | --json | --raw | --raw-header | --raw-payload]
                        [--range <range> | --treegroup <treegroup>]
                        [<hwp5file>] [<record-stream>]
```

3.7.1 Positional Arguments

<hwp5file>	.hwp file to analyze
<record-stream>	Record-structured internal streams. (e.g. DocInfo, BodyText/*)

3.7.2 Named Arguments

--simple	Print records as simple tree Default: False
--json	Print records as json Default: False
--raw	Print records as is Default: False
--raw-header	Print record headers as is Default: False
--raw-payload	Print record payloads as is Default: False
--range	Specifies the range of the records. N-M means “from the record N to M-1 (excluding M)” N means just the record N
--treegroup	Specifies the N-th subtree of the record structure.

Example:

```
$ hwp5proc records samples/sample-5017.hwp DocInfo
```

Example:

```
$ hwp5proc records samples/sample-5017.hwp DocInfo --range=0-2
```

If neither <hwp5file> nor <record-stream> is specified, the record stream is read from the standard input with an assumption that the input is in the format version specified by -V option.

Example:

```
$ hwp5proc records --raw samples/sample-5017.hwp DocInfo --range=0-2 > tmp.rec
$ hwp5proc records < tmp.rec
```

3.8 models

Print parsed binary models of .hwp file record streams.

Print parsed binary models in the specified <record-stream>.

```
usage: hwp5proc models [-h] [--file-format-version <version>]
                        [--simple | --json | --format <format> | --events]
                        [--treegroup <treegroup> | --seqno <treegroup>]
                        [<hwp5file>] [<record-stream>]
```

3.8.1 Positional Arguments

<hwp5file> .hwp file to analyze

<record-stream> Record-structured internal streams. (e.g. DocInfo, BodyText/*)

3.8.2 Named Arguments

--file-format-version, -V Specifies HWPv5 file format version of the standard input stream

--simple Print records as simple tree
Default: False

--json Print records as json
Default: False

--format Print records formatted

--events Print records as events
Default: False

--treegroup Specifies the N-th subtree of the record structure.

--seqno Print a model of <seqno>-th record

Example:

```
$ hwp5proc models samples/sample-5017.hwp DocInfo
$ hwp5proc models samples/sample-5017.hwp BodyText/Section0

$ hwp5proc models samples/sample-5017.hwp docinfo
$ hwp5proc models samples/sample-5017.hwp bodytext/0
```

Example:

```
$ hwp5proc models --simple samples/sample-5017.hwp bodytext/0
$ hwp5proc models --format='% (level)s % (tagname)s\\n' \\
  samples/sample-5017.hwp bodytext/0
```

Example:

```
$ hwp5proc models --simple --treegroup=1 samples/sample-5017.hwp bodytext/0
$ hwp5proc models --simple --seqno=4 samples/sample-5017.hwp bodytext/0
```

If neither <hwp5file> nor <record-stream> is specified, the record stream is read from the standard input with an assumption that the input is in the format version specified by -V option.

Example:

```
$ hwp5proc cat samples/sample-5017.hwp BodyText/Section0 > Section0.bin
$ hwp5proc models -V 5.0.1.7 < Section0.bin
```

3.9 find

Find record models with specified predicates.

Find record models with specified predicates.

```
usage: hwp5proc find [-h] [--from-stdin]
                    [--model <model-name> | --tag <hwptag>] [--incomplete]
                    [--format <format>] [--dump]
                    [<hwp5files> [<hwp5files> ...]]
```

3.9.1 Positional Arguments

<hwp5files> .hwp files to analyze

3.9.2 Named Arguments

--from-stdin	get filenames from stdin Default: False
--model	filter with record model name
--tag	filter with record HWPTAG
--incomplete	filter with incompletely parsed content Default: False
--format	record output format
--dump	dump record Default: False

Example: Find paragraphs:

```
$ hwp5proc find --model=Paragraph samples/*.hwp
$ hwp5proc find --tag=HWPTAG_PARA_TEXT samples/*.hwp
$ hwp5proc find --tag=66 samples/*.hwp
```

Example: Find and dump records of HWPTAG_LIST_HEADER which is parsed incompletely:

```
$ hwp5proc find --tag=HWPTAG_LIST_HEADER --incomplete --dump samples/*.hwp
```

3.10 xml

Transform .hwp files into an XML.

Transform <hwp5file> into an XML.

```
usage: hwp5proc xml [-h] [--embedbin] [--no-xml-decl] [--output <file>]
                   [--format <format>] [--no-validate-wellformed]
                   <hwp5file>
```

3.10.1 Positional Arguments

<hwp5file> .hwp file to analyze

3.10.2 Named Arguments

--embedbin Embed BinData/* streams in the output XML.
Default: False

--no-xml-decl Do not output <?xml ... ?> XML declaration.
Default: False

--output Output filename.

--format “flat”, “nested” (default: “nested”)

--no-validate-wellformed Do not validate well-formedness of output.
Default: False

Example:

```
$ hwp5proc xml samples/sample-5017.hwp > sample-5017.xml
$ xmllint --format sample-5017.xml
```

With **--embedbin** option, you can embed base64-encoded BinData/* files in the output XML.

Example:

```
$ hwp5proc xml --embedbin samples/sample-5017.hwp > sample-5017.xml
$ xmllint --format sample-5017.xml
```

3.11 rawunz

Deflate an headerless zlib-compressed stream.

Deflate an headerless zlib-compressed stream

```
usage: hwp5proc rawunz [-h]
```

3.12 diststream

Decode a distribute document stream.

Decode a distribute document stream.

```
usage: hwp5proc diststream [-h] [--sha1 | --key] [--raw]
```

3.12.1 Named Arguments

--sha1	Print SHA-1 value for decryption. Default: False
--key	Print decrypted key. Default: False
--raw	Print raw binary objects as is. Default: False

CONVERTERS (*EXPERIMENTAL*)

Convert HWPv5 documents into other document formats.

4.1 Requirements

The conversions are performed with [XSLT](#) internally and verified with [Relax NG](#) if possible.

For these processing, the converters requires [lxml](#) ([homepage](#)) or [libxml2](#)'s `xsltproc` / `xmllint` programs.

For `lxml` installation:

```
pip install --user lxml # install to user directory
pip install lxml       # install with virtualenv
```

or see [Installing lxml](#).

(Currently conversions with `lxml` 2.3.5 is tested and verified to be working. `lxml` versions below that may work too, but those are not tested.)

For `xsltproc` / `xmllint` installation:

```
sudo apt-get install xsltproc libxml2-utils # Debian/Ubuntu
```

Optional environment variables `PYHWP_XSLTPROC` and `PYHWP_XMLLINT` specifies the paths of the each programs. (If not set, `xsltproc` and/or `xmllint` should be in the one of the directories specified in `PATH`.)

4.2 hwp5odt: ODT conversion

HWPv5 to odt converter

```
usage: hwp5odt [-h] [--version] [--loglevel LOGLEVEL] [--logfile LOGFILE]
              [--output OUTPUT] [--styles | --content | --document]
              [--embed-image | --no-embed-image]
              <hwp5file>
```

4.2.1 Positional Arguments

<hwp5file> .hwp file to convert

4.2.2 Named Arguments

--version	show program's version number and exit
--loglevel	Set log level.
--logfile	Set log file.
--output	Output file
--styles	Generate styles.xml Default: False
--content	Generate content.xml Default: False
--document	Generate .fodt Default: False
--embed-image	Embed images in output xml. Default: False
--no-embed-image	Do not embed images in output xml. Default: False

4.3 hwp5html: HTML conversion

HWPv5 to HTML converter

```
usage: hwp5html [-h] [--version] [--loglevel LOGLEVEL] [--logfile LOGFILE]
               [--output OUTPUT] [--css | --html]
               <hwp5file>
```

4.3.1 Positional Arguments

<hwp5file> .hwp file to convert

4.3.2 Named Arguments

--version	show program's version number and exit
--loglevel	Set log level.
--logfile	Set log file.
--output	Output file
--css	Generate CSS Default: False
--html	Generate HTML Default: False

4.4 hwp5txt: text conversion

HWPv5 to txt converter

```
usage: hwp5txt [-h] [--version] [--loglevel LOGLEVEL] [--logfile LOGFILE]
               [--output OUTPUT]
               <hwp5file>
```

4.4.1 Positional Arguments

<hwp5file>	.hwp file to convert
-------------------------	----------------------

4.4.2 Named Arguments

--version	show program's version number and exit
--loglevel	Set log level.
--logfile	Set log file.
--output	Output file

HACKING GUIDE

Standard procedures to hacking on pyhwp.

Contents:

5.1 Setup development environment

5.1.1 1. Install prerequisites

- CPython 2.7
- *virtualenv*
- GNU *Make*

5.1.2 2. Clone the source repository

```
$ git clone https://github.com/mete0r/pyhwp.git
```

5.1.3 3. Initialize the environment

Bootstrap development environment:

```
$ make bootstrap  
$ . bin/activate
```

5.1.4 4. Check basic stuffs

Run *hwp5proc*:

```
$ hwp5proc --help
```

To run tests:

```
$ tox
```

5.2 Directory Layout

```
pyhwp                Project Root
|
+-- pyhwp/           Source packages root
|   |
|   +-- hwp5/        Source package
|   |
|   +-- pyhwp-tests/ Test packages root
|       |
|       +-- hwp5_tests/ Test package
|
+-- docs/            Documentations, i.e. this document!
|
+-- bin/             hwp5proc, hwp5odt, build/testing scripts, etc.,
|
+-- etc/             development configuration files
|
+-- misc/            development configuration templates / helper scripts
|
+-- tools/           development helper packages
|
.
. (various directories)
.
```

After the initial invocation of `buildout` completes successfully, your directory will have a few more new generated directories, e.g. `bin/`, `develop-eggs/`. These are the standard `buildout` directories, which we will not cover the every details of them here. For general information, see [Directory Structure of a Buildout](#).

Followings are `pyhwp` specific informations:

5.2.1 / - project root directory

The project root directory contains project configuration files.

buildout.cfg `buildout` configuration file.

setup.py, **setup.cfg** `pyhwp` setup files.

tox.ini `tox` configuration file. This file will be automatically generated from `tox.ini.in` by **bin/buildout**. See `[tox]` parts in `buildout.cfg`.

tox.ini.in `tox` configuration template file. If you want to modify `tox` configuration, edit this file and run **bin/buildout** again.

5.2.2 bin/ - Buildout generated scripts

This directory will be populated with scripts generated from the `pyhwp` package and the various development helper packages/scripts.

`pyhwp` generate following scripts:

hwp5proc HWP format version 5 files processor. See *[hwp5proc: HWPv5 processor](#)*.

hwp5odt, **hwp5txt**, **hwp5html** Experimental converters. See *[Converters \(Experimental\)](#)*.

Development helper scripts (incomplete):

buildout (Re)generate the development environment.

test-core Run a quick unit test.

5.2.3 `tools/` - Development helper packages

`discover.python/` `discover.lxml/` `discover.jre/` `discover.lo/` `install.jython/`

Discover multiple python versions, lxml, JRE, Libreoffice to use in the developement environment.
Provides `zc.buildout` recipes.

`xslttest/`

an XSLT test runner.

`oxt.tool/`

Build and test `.oxt` packages with the LibreOffice.

5.3 Hack & Test

If you modify some modules in `hwp5` package in the `pyhwp/` directory, you can test the modification with the `hwp5proc` script in the `bin/` directory.

You can test the `hwp5` package by executing `bin/test-core`, but it's just a quick test and not a complete test suite. If you want to run a full-blown test suite, run `tox`, which tries to test `pyhwp` in various `virtualenv`-isolated python platforms, including Python 2.5, 2.6, 2.7, Jython 2.5 and PyPy.

```
$ bin/buildout
(...)
$ vim pyhwp/hwp5/proc/__init__.py
(HACK HACK HACK)
$ bin/test-core
$ bin/hwp5proc ...
$ bin/tox
```


CHANGES

6.1 0.1b16 (unreleased)

- [CVE-2023-0286] Depends on cryptography >= 40.0.1
- [CVE-2022-2309] Depends on lxml >= 4.9.2

6.2 0.1b15 (2020-05-30)

- Unknown Numbering.Kind value of 6, which is not described in the official specification docs, has been added. See #177.

6.3 0.1b14 (2020-05-17)

- Fix xmldump_flat for Python 3.8

6.4 0.1b13 (2020-05-17)

- Replace docopt with argparse.
- Workaround for BinData decompression (#175, #176)

6.5 0.1b12 (2019-04-08)

- Add Python 3.x support.
- Add an optional dependency on colorlog for colorful logging
- Remove dependency on hypua2jamo, resulting no automatic conversion of Hanyang PUA to Hangul Jamo

6.6 0.1b11 (2019-03-21)

- Remove dependency on PyCrypto. - [CVE-2013-7458], [CVE-2018-6594]
- Add dependency on cryptography.

6.7 0.1b10 (2019-03-21)

- Drop support for Python 2.5, 2.6.
- Prefer 'olefile' to 'OleFileIO_PL'.
- Fix 'Dutmal' control attribute names.
- hwp5html: represent path names in bytes
- Declare some dependencies with environment markers: olefile, lxml, pycrypto
- Update dependency on hypua2jamo >= 0.4.4

6.8 0.1b9 (2016-02-26)

- hwp5html: serveral improvements - lang-* classes of span elements and associated css font-family - horizontal page layouts - Single page layout - enhance horizontal positioning of TableControl, GShapeObject
- distdoc: fix sha1offset (by Hodong Kim)

6.9 0.1b8 (2014-11-03)

- hwp5view: experimental viewer with webkitgtk+
- hwp5proc: xml --formats ("flat", "nested")
- hwp5proc: models --events (experimental)
- hwp5proc: models --seqno --format (incompatible changes)
- hwp5proc: find --from-stdin
- hwp5proc: find --format
- binmodels: GShapeObjectCaption
- olestorage: Gsf implementation through python-gi
- olestorage: use new olefile instead of OleFileIO_PL

6.10 0.1b7 (2014-01-31)

- support distribution docs. (based on [Changwoo Ryu's algorithm](#))

6.11 0.1b6 (2014-01-20)

- binmodel: change type of TableCell dimensions to signed integer
- hwp5odt: fix NCName for style:name (close #140)
- hwp5proc: fix with-statement in 'xml' command for Python 2.5
- hwp5proc: mark 'xml' command experimental

6.12 0.1b5 (2013-10-29)

- close #134
- hwp5html generates .xhtml instead of .html
- hwp5proc: new '--no-xml-decl' option
- hwp5odt: fix to not use '/' in resulting style names
- hwp5proc: IdMappings.memoshape only if version > 5.0.1.6

6.13 0.1b4 (2013-07-03)

- hwp5proc records: new option '--raw-header'
- hwp5odt: new '--document' option produces single ODT XML files (*.fodt)
- hwp5odt: new '--styles', '--content' option produces styles/content XML files
- ODT XSL files restructured

6.14 0.1b3 (2013-06-18)

- Fix IdMappings (#125)
- hwp5proc records: new option '--raw-payload'
- hwp5proc xml: FlagsType as xsd:hexBinary
- Various binary/xml models changes

6.15 0.1b2 (2013-06-08)

- Add PyPy support

INDICES AND TABLES

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