
innoconv-mintmod Documentation

Release 0.0.2

Innocampus

Jan 13, 2021

CONTENTS

1	What is innoConv (mintmod)?	1
2	Table of contents	3
2.1	Installation	3
2.1.1	Prerequisites	3
2.1.2	Dependencies	3
2.1.2.1	Python interpreter	3
2.1.2.2	Pandoc	3
2.1.2.3	Virtual environment	3
2.2	How to use innoconv-mintmod	4
2.2.1	Command line arguments	4
2.2.1.1	Positional Arguments	4
2.2.1.2	Named Arguments	4
2.3	Converting legacy mintmod content	5
2.3.1	Adjust commands	5
2.3.1.1	Remove \iftt{...}\else{...}\fi commands	5
2.3.1.2	Unwanted LaTeX commands	6
2.3.1.3	Including other modules	6
2.3.2	Replace strings	6
2.3.3	Clean up code	7
2.4	Architecture	7
2.4.1	The command line interface	7
2.4.2	panzer	8
2.5	Module overview	8
2.5.1	innoconv_mintmod.constants	8
2.5.2	innoconv_mintmod.errors	9
2.5.3	innoconv_mintmod_mintmod.mintmod_filter	9
2.5.3.1	innoconv_mintmod.mintmod_filter.commands	9
2.5.3.2	innoconv_mintmod.mintmod_filter.elements	14
2.5.3.3	innoconv_mintmod.mintmod_filter.environments	15
2.5.3.4	innoconv_mintmod.mintmod_filter.filter_action	16
2.5.3.5	innoconv_mintmod.mintmod_filter.math	16
2.5.4	innoconv_mintmod.runner	17
2.5.5	innoconv_mintmod.utils	17
2.5.6	generate_innодoc	19
3	Indices and tables	21
	Python Module Index	23

CHAPTER
ONE

WHAT IS INNOCONV (MINTMOD)?

This package converts mintmod-flavoured LaTeX into Markdown.

It can be seen as a shim for `mintmod.tex` and handles important mintmod commands by translating them to regular Pandoc elements.

Technically spoken it's essentially a wrapper to Pandoc.

Read more about innoconv-mintmods [*Architecture*](#).

TABLE OF CONTENTS

2.1 Installation

2.1.1 Prerequisites

innoconv-mintmod is mainly used on Linux machines. It might work on Mac OS, Windows/Cygwin/WSL. You are invited to share experiences in doing so.

2.1.2 Dependencies

The only dependencies you have to provide yourself is Pandoc and the Python interpreter.

All others can be installed into a [Virtual environment](#).

2.1.2.1 Python interpreter

While other versions of Python might work fine, innoconv-mintmod was tested with **Python 3.9**. Make sure you have it installed.

2.1.2.2 Pandoc

You need to make sure to have a recent version of the pandoc binary available in PATH (**Pandoc 2.11.x** at the time of writing). There are [several ways on installing Pandoc](#).

Older versions of Pandoc might not work.

2.1.2.3 Virtual environment

Setup and activate a virtual environment in a location of your choice.

```
$ python3 -m venv venv
$ source venv/bin/activate
```

Install innoconv-mintmod in your virtual environment using pip.

```
$ pip install --process-dependency-links -e git+https://gitlab.tubit.tu-berlin.de/
˓→innodoc/innoconv-mintmod.git#egg=innoconv-mintmod
```

If everything went fine you should now have access to the `innoconv-mintmod` command.

```
$ innoconv-mintmod
usage: innoconv-mintmod [-h] [-o OUTPUT_DIR_BASE]
                         [-f {latex+raw_tex,markdown}]
                         [-t {html5,json,latex,markdown,asciidoc}] [-l {de,en}]
                         [-d] [-i] [-r] [-s]
                         source
innoconv-mintmod: error: the following arguments are required: source
```

Congratulations!

2.2 How to use innoconv-mintmod

You can run the converter in your content directory.

```
$ innoconv-mintmod .
```

This will trigger the conversion for this folder.

2.2.1 Command line arguments

```
usage: innoconv_mintmod [-h] [-o OUTPUT_DIR_BASE]
                         [-f {latex+raw_tex,markdown}]
                         [-t {html5,json,latex,markdown,asciidoc}] [-l {de,en}]
                         [-d] [-i] [-r] [-g]
                         source
```

2.2.1.1 Positional Arguments

source content directory or file

2.2.1.2 Named Arguments

-o, --output-dir-base	output base directory Default: “./innoconv_mintmod_output”
-f, --from	Possible choices: latex+raw_tex, markdown input format Default: “latex+raw_tex”
-t, --to	Possible choices: html5, json, latex, markdown, asciidoc output format Default: “markdown”
-l, --language-code	Possible choices: de, en two-letter language code Default: “de”

-d, --debug	debug mode (output HTML and highlight unknown commands)
	Default: False
-i, --ignore-exercises	don't show logs for unknown exercise commands/envs
	Default: False
-r, --remove-exercises	remove all exercise commands/envs
	Default: False
-g, --generate-innodoc	split sections and generate manifest.yaml
	Default: True

2.3 Converting legacy mintmod content

In this chapter some findings are documented on how to prepare content so it can be read by the *innoconv-mintmod command*.

Note: It's not a complete list and there might be things missing that need to be done in your specific case.

First of all make sure all content is *UTF-8 encoded*. If not, tools like `iconv` can be helpful.

2.3.1 Adjust commands

There are some mintmod commands Pandoc is not able to parse. You need to manually replace them throughout your project.

2.3.1.1 Remove `\ifttm... \else... \fi` commands

`mintmod_ifttm` can get rid of all `\ifttm` commands.

Usage:

```
$ mintmod_ifttm < file_in.tex > file_out.tex
```

Automate on many files:

```
$ find . -name '*.tex' | xargs -I % sh -c 'mintmod_ifttm < % > %_changed && mv %_changed %'
```

Warning: The script cares only about `\ifttm... \else... \fi` with an `\else` command. There may be occurrences of `\ifttm... \fi` (without `\else`). You need to remove them manually!

2.3.1.2 Unwanted LaTeX commands

A couple of commands are superfluous or doesn't make sense in a web-first content publishing platform like innoDoc. So remove any occurrences of the following commands.

- `\input{mintmod.tex}`
- `\input{english.tex}`
- `\begin{document} \begin{document}`
- `\MPragma{MathSkip}`
- `\Mtikzexternalize`
- `\relax`
- `\- (hyphenation)`
- `\pagebreak`
- `\newpage`
- `\MPrintIndex`
- `\relax`

Automate:

```
find . -type f -name '*.tex' -or -name '*.rtex' | xargs perl -i -pe 's/\\"input  
↪{mintmod(.tex|)}\w*\n//igs'
```

2.3.1.3 Including other modules

Pandoc doesn't understand `\IncludeModule`. Change these statements to proper LaTeX commands.

`\IncludeModule{folder}{file.tex}` → `\input{folder/file.tex}`.

2.3.2 Replace strings

There are a couple of special characters you need to replace yourself.

- `\"a` → ä
- `\"o` → ö
- `\"u` → ü
- `\"A` → Ä
- `\"O` → Ö
- `\"U` → Ü
- `\"s` → ß
- `\"s` → ß
- `{\ss}` → ß
- `\ss ``` → ``ß
- `\ss\` → ß
- `\ss{}` → ß

- \ss → ß
- "a → ä
- "o → ö
- "u → ü
- "A → Ä
- "O → Ö
- "U → Ü
- " ` → „
- `` → „
- ' ' → “
- " ' → ”

Automate:

```
find . -type f -name '*.tex' -or -name '*.rtex' | xargs sed -i 's/\\"a/ä/g'
```

2.3.3 Clean up code

Remove unused files from your project and keep track of your changes using a VCS.

2.4 Architecture

This section gives an overview of innoconv-mintmod's architecture.

2.4.1 The command line interface

The entry point is the command line tool `innoconv-mintmod`.

It calls `panzer` with the correct parameters.

Most of the magic happens in the package `MintmodFilterAction`.

It is implemented as a Pandoc filter and provides functions to deal with a number of special LaTeX mintmod commands Pandoc would otherwise just ignore.

All special commands are translated into primitives Pandoc knows already. Additionally information is encoded in attributes that are attached to the resulting elements.

The result of the `MintmodFilterAction` is a regular Pandoc AST that can be further processed by Pandoc output modules, thus be translated to Markdown, LaTeX, HTML and so forth.

The Pandoc JSON output is processed by `generate_innодoc.py`. It's implemented as a post-flight `panzer` script.

2.4.2 panzer

`panzer` is a small wrapper script around Pandoc. It enriches Pandoc with several useful features that just happened to match this project's needs.

First of all it is possible to define profiles (called *styles* in `panzer`) that can already define parameters on how to run Pandoc.

Furthermore it can manage applied filters, run pre- and postprocessors etc.

You can find its configuration in the sub-directory `.panzer`.

2.5 Module overview

2.5.1 innoconv_mintmod.constants

Project constants are defined here.

`innoconv_mintmod.constants.INDEX_LABEL_PREFIX`

Element class for index labels

`innoconv_mintmod.constants.COMMANDS_IRREGULAR`

Math commands with irregular arguments, key=command-name, value=formatstring or value=dict (number of arguments, formatstring)

`innoconv_mintmod.constants.REGEX_PATTERNS`

Regular expressions

`innoconv_mintmod.constants.ELEMENT_CLASSES`

Element classes

`innoconv_mintmod.constants.MINTMOD_SUBJECTS`

Subjects as used in mintmod command \MSetSubject

`innoconv_mintmod.constants.LANGUAGE_CODES`

Supported language codes

`innoconv_mintmod.constants.DEFAULT_LANGUAGE_CODE`

Default language code

`innoconv_mintmod.constants.DEFAULT_OUTPUT_DIR_BASE`

Default innoconv output directory

`innoconv_mintmod.constants.DEFAULT_OUTPUT_FORMAT`

Default innoconv output format

`innoconv_mintmod.constants.OUTPUT_FORMAT_EXT_MAP`

mapping between output formats and file extensions

`innoconv_mintmod.constants.OUTPUT_FORMAT_CHOICES`

Output format choices

`innoconv_mintmod.constants.ROOT_DIR`

project root dir

`innoconv_mintmod.constants.PANZER_SUPPORT_DIR`

panzer support directory

`innoconv_mintmod.constants.ENCODING`

encoding used in this project

2.5.2 innoconv_mintmod.errors

Exceptions are defined here.

```
exception innoconv_mintmod.errors.ParseError
    Raised when a mintmod command cannot be parsed.
```

2.5.3 innoconv_mintmod_mintmod.mintmod_filter

This module handles mintmod LaTeX commands.

Commands and environments are defined in the classes Commands and Environments.

2.5.3.1 innoconv_mintmod.mintmod_filter.commands

Handle mintmod LaTeX commands.

Note: Provide a `handle_CMDNAME` function for handling CMDNAME command. You need to `slugify` the command name.

Example: `handle_msection` method will receive the command `\MSection`.

```
class innoconv_mintmod.mintmod_filter.commands.Commands
```

Handlers for commands are defined here.

Given the command:

```
\MSection{Foo}
```

The handler method `handle_msection` receives the following arguments:

- `cmd_args: ['Foo']`
- `elem: panflute.base.Element`

```
handle_bigskip(cmd_args, elem)
    Handle \bigskip command.
```

```
handle_clearpage(cmd_args, elem)
    Handle \clearpage command.
```

```
handle_glqq(cmd_args, elem)
    Handle \glqq command.
```

```
handle_grqq(cmd_args, elem)
    Handle \grqq command.
```

```
handle_highlight(cmd_args, elem)
    Handle highlight command.
```

This seems to be some sort of formatting command. There's no documentation and it does nothing in the mintmod code. We just keep the information here.

```
handle_hspace(cmd_args, elem)
    Handle \hspace and \hspace* command.
```

```
handle_input(cmd_args, elem)
    Handle \input command.
```

handle_jhtmlhinweiseingabefunktionen (*cmd_args, elem*)

Handle \jHTMLHinweisEingabeFunktionen command.

handle_jhtmlhinweiseingabefunktionenexp (*cmd_args, elem*)

Handle \jHTMLHinweisEingabeFunktionenExp command.

handle_mblank (*cmd_args, elem*)

Handle \MBlank command.

handle_mcontenttable (*cmd_args, elem*)

Handle \MContentTable command.

handle_mcopyrightcollection (*cmd_args, elem*)

Handle \MCopyrightCollection command.

handle_mcopyrightlabel (*cmd_args, elem*)

Handle \MCopyrightLabel command.

handle_mdeclaresiteuxid (*cmd_args, elem*)

Handle \MDeclareSiteUXID command.

The command can occur in an environment that is parsed by a subprocess. In this case there's no last header element. The process can't set the ID because it can't access the doc tree. Instead it replaces the \MDeclareSiteUXID by an element that is found by the parent process using function `innoconv.utils.extract_identifier()`.

handle_mdirectrouletteexercises (*cmd_args, elem*)

Handle \MDirectRouletteExercises command.

Remember points for next question.

handle_medskip (*cmd_args, elem*)

Handle \medskip command.

handle_mentry (*cmd_args, elem*)

Handle \MEntry command.

This command creates an entry for the index.

handle_mequationitem (*cmd_args, elem*)

Handle \MEquationItem command.

handle_mextlink (*cmd_args, elem*)

Handle \MExtLink command.

This command inserts an external link.

handle_mformelzoomhint (*cmd_args, elem*)

Handle \MFormelZoomHint command.

handle_mglobalchaptertag (*cmd_args, elem*)

Handle \MGlobalChapterTag command.

handle_mglobalconftag (*cmd_args, elem*)

Handle \MGlobalConfTag command.

handle_mglobaldatatag (*cmd_args, elem*)

Handle \MGlobalDataTag command.

handle_mglobalfavotag (*cmd_args, elem*)

Handle \MGlobalFavoTag command.

handle_mgloballocationtag (*cmd_args, elem*)

Handle \MGlobalLocationTag command.

handle_mgloballogintag (*cmd_args, elem*)
Handle \MGlobalLoginTag command.

handle_mgloballogouttag (*cmd_args, elem*)
Handle \MGlobalLogoutTag command.

handle_mglobalsearchtag (*cmd_args, elem*)
Handle \MGlobalSearchTag command.

handle_mglobalstart (*cmd_args, elem*)
Handle \MGlobalStart command.

handle_mglobalstesttag (*cmd_args, elem*)
Handle \MGlobalSTestTag command.

handle_mgraphics (*cmd_args, elem, add_desc=True*)
Handle \MGraphics.

Embed an image with title.

Example: MGraphics{img.png}{scale=1}{title}

handle_mgraphicssolo (*cmd_args, elem*)
Handle \MGraphicsSolo.

Embed an image without title. Uses filename as image title.

handle_mgroupbutton (*cmd_args, elem*)
Handle \MGroupButton command.

Render empty as this button is displayed automatically in clients.

handle_mindex (*cmd_args, elem*)
Handle \MIndex command.

This command creates an invisible entry for the index.

handle_minputhint (*cmd_args, elem*)
Handle \MInputHint command.

handle_mllabel (*cmd_args, elem*)
Handle \MLLabel command.

Will search for the previous header element and update its ID to the ID defined in the command. Otherwise proceed like \MDeclareSiteUXID.

Hides identifier in fake element like (*innoconv_mintmod.mintmod_filter.commands.Commands.handle_mdeclaresiteuxid()*).

handle_mlcheckbox (*cmd_args, elem*)
Handle questions defined by \MLCheckbox command

handle_mlfunctionquestion (*cmd_args, elem*)
Handle questions defined by \MLFunctionQuestion command

handle_mlintervalquestion (*cmd_args, elem*)
Handle questions defined by \MLIntervalQuestion command

handle_mlparsedquestion (*cmd_args, elem*)
Handle questions defined by \MLParsedQuestion command

handle_mlquestion (*cmd_args, elem*)
Handle questions defined by \MLQuestion command

handle_mlsimplifyquestion (*cmd_args, elem*)
Handle questions defined by \MLSimplifyQuestion command

handle_mlsspecialquestion (*cmd_args, elem*)
Handle questions defined by \MLSpecialquestion command

handle_mmodstartbox (*cmd_args, elem*)
Handle \MModStartBox command.

This command displays a table of content for the current chapter. This is handled elsewhere and becomes a no-op.

handle_mnref (*cmd_args, elem*)
Handle \MNRef command.

This command inserts a section link.

handle_modsemph (*cmd_args, elem*)
Handle modsemph command.

handle_modstextbf (*cmd_args, elem*)
Handle modstextbf command.

handle_mppragma (*cmd_args, elem*)
Handle \MPragma command.

This command was used to embed build time flags for mintmod. It becomes a no-op.

handle_mprintindex (*cmd_args, elem*)
Handle \MPrintIndex command.

Index will be printed automatically. It becomes a no-op.

handle_mpullsite (*cmd_args, elem*)
Handle \MPullSite command.

handle_mref (*cmd_args, elem*)
Handle \MRef command.

This command translates to \vref.

handle_msection (*cmd_args, elem*)
Handle \MSection command.

handle_msetpoints (*cmd_args, elem*)
Handle \MSetPoints command.

Remember points for next question.

handle_msetsectionid (*cmd_args, elem*)
Handle \MSetSectionID command.

Will search for the previous header element and update its ID to the ID defined in the command.

handle_msetsubject (*cmd_args, elem*)
Handle \MSetSubject{} command.

Command defines the category.

handle_msref (*cmd_args, elem*)
Handle \MSRef command.

This command inserts a fragment-style link.

handle_msubject (*cmd_args, elem*)
Handle \MSubject{title} command.
Command defines the document title.

handle_msubsection (*cmd_args, elem*)
Handle \MSubsection

handle_msubsubsection (*cmd_args, elem*)
Handle \MSubsubsection

handle_msubsubsectionx (*cmd_args, elem*)
Handle \MSubsubsectionx command. Which will generate a level 4 header.

handle_msubsubsubsectionx (*cmd_args, elem*)
Handle \MSubsubsubsectionx command. Which will generate a level 4 header.
From logical point of view this should be level 5. But from looking at the sources, level 4 is correct.

handle_mtikzauto (*cmd_args, elem*)
Handle \MTikzAuto command.
Create a CodeBlock with TikZ code.

handle_mttitle (*cmd_args, elem*)
Handle \MTTitle command.
This is an equivalent to \subsubsection

handle_mugraphics (*cmd_args, elem*)
Handle \MUGraphics.
Embed an image with title.

handle_mgraphicssolo (*cmd_args, elem*)
Handle \MUGraphicsSolo.
Embed an image without title.

handle_mvvideo (*cmd_args, elem*)
Handle \MVvideo.
Just return a Link Element.

handle_mwatermarksettings (*cmd_args, elem*)
Handle \MWatermarkSettings command.

handle_myoutubevideo (*cmd_args, elem*)
Handle \MYoutubeVideo.
Just return a Link Element.

handle_mzahl (*cmd_args, elem*)
Handle \MZahl command.
This is a math command but in fact occurs also in text.

handle_mxyzhltrennzeichen (*cmd_args, elem*)
Handle \MXYZhltrennzeichen command.
It is transformed to a \decmarker command and later substituted by MathJax. This is already in math substitutions but as it occurs outside of math environments it's defined here too.

handle_newline (*cmd_args, elem*)
Handle newline command.

```
handle_newpage(cmd_args, elem)
    Handle \newpage command.

    A display related command. It becomes a no-op.

handle_noindent(cmd_args, elem)
    Handle \noindent command.

handle_quad(cmd_args, elem)
    Handle \quad command.

handle_smallskip(cmd_args, elem)
    Handle \smallskip command.

handle_special(cmd_args, elem)
    Handle \special command.

    This command is used to embed HTML in LaTeX source.

handle_vspace(cmd_args, elem)
    Handle \vspace command.

    A display related command. It becomes a no-op.
```

2.5.3.2 innoconv_mintmod.mintmod_filter.elements

Convenience functions and classes for creating common elements.

```
class innoconv_mintmod.mintmod_filter.elements.Question(*args, **kwargs)
    Wrapper/Factory class that inherits from pf.Element and will return pf.Code instances, with special classes and attributes, depending on the given mintmod class.
```

```
static parse_args(cmd_args, *names)
    Parse exercise arguments.
```

Receive a list of argument names and a list of values and return a pandoc conformant argument array containing element's arguments. In other words: take a list of arguments and make them named arguments for easier referencing.

```
innoconv_mintmod.mintmod_filter.elements.create_content_box(elem_content,
    elem_classes, lang)
```

Create a content box.

Convenience function for creating content boxes that only differ by having different content and classes.

```
innoconv_mintmod.mintmod_filter.elements.create_header(title_str, doc, level=0,
    parse_text=False, identifier='', short_title=None)
```

Create a header element.

Because headers need to be referenced by later elements, references to the last found header is remembered.

```
innoconv_mintmod.mintmod_filter.elements.create_image(filename, descr, elem,
    add_descr=True, block=True)
```

Create an image element.

2.5.3.3 innoconv_mintmod.mintmod_filter.environments

Handle mintmod LaTeX environments.

Note: Provide a `handle_ENVNAME` function for handling ENVNAME environment. You need to `slugify` the environment name.

Example: `handle_mxcontent` method will receive the `\begin{MXContent}... \end{MXContent}` environment.

class innoconv_mintmod.mintmod_filter.environments.**Environments**

Handlers for environments are defined here.

Given the environment:

```
\begin{MXContent} {Foo title long} {Foo title} {STD}
    Foo content
\end{MXContent}
```

The handler method `handle_mxcontent` receives the following arguments:

- `elem_content`: 'Foo content'
- `cmd_args`: ['Foo title long', 'Foo title', 'STD']
- `elem`: `panflute.elements.RawBlock`

handle_html (`elem_content, env_args, elem`)

Handle `\html` environment.

handle_itemize (`elem_content, env_args, elem`)

Handle itemize environments, that were not correctly recognized by pandoc. This e.g. happens if there are `\MExerciseItems` environments contained in the items.

handle_mcontent (`elem_content, env_args, elem`)

Handle `\MContent` environment.

handle_mcoshzusatz (`elem_content, env_args, elem`)

Handle `\MCOSHZusatz` environment.

handle_mexample (`elem_content, env_args, elem`)

Handle `\MExample` command.

handle_mexercise (`elem_content, env_args, elem`)

Handle `\MExercise` environment.

handle_mexercisecollection (`elem_content, env_args, elem`)

Handle `\MExerciseCollection` environment.

handle_mexerciseitems (`elem_content, env_args, elem`)

Handle `\MExerciseItems` environments by returning an ordered list containing the `\item`s defined in the environment. This is needed on top of `handle_itemize` as there are also `mexerciseitems` environments outside itemize environments.

handle_mexercises (`elem_content, env_args, elem`)

Handle `\MExercises` environment.

handle_mexperiment (`elem_content, env_args, elem`)

Handle `\MExperiment` environment.

handle_mhint (`elem_content, env_args, elem`)

Handle `\MHint` command.

handle_minfo (*elem_content, env_args, elem*)

Handle \MInfo environment.

handle_mintro (*elem_content, env_args, elem*)

Handle \MIntro environment.

handle_mquestiongroup (*elem_content, env_args, elem*)

Handle \MQuestionGroup environments.

In mintmod used to group checkboxes together. We just return the content as questions are grouped by exercises anyway.

handle_msectionstart (*elem_content, env_args, elem*)

Handle \MSectionStart environment.

handle_mtest (*elem_content, env_args, elem*)

Handle \MTest environment.

handle_mxcontent (*elem_content, env_args, elem*)

Handle \MXContent environment.

handle_mxinfo (*elem_content, env_args, elem*)

Handle \MXInfo environment.

2.5.3.4 innoconv_mintmod.mintmod_filter.filter_action

Pandoc filter that transforms mintmod commands.

class innoconv_mintmod.mintmod_filter.filter_action.**MintmodFilterAction** (*debug=False*)

The Pandoc filter is defined in this class.

filter (*elem, doc*)

Receive document elements.

This method receives document elements from Pandoc and delegates handling of simple substitutions, mintmod commands and environments.

Parameters

- **elem** (`panflute.base.Element`) – Element to handle
- **doc** (`panflute.elements.Doc`) – Document

2.5.3.5 innoconv_mintmod.mintmod_filter.math

Handle mintmod math commands.

innoconv_mintmod.mintmod_filter.math.**handle_math** (*elem*)

Handle mintmod text substitutions and some commands with irregular arguments.

2.5.4 innoconv_mintmod.runner

Runner module

```
class innoconv_mintmod.runner.InnoconvRunner(source,          output_dir_base,      lan-
                                              guage_code,        ignore_exercises=False,
                                              remove_exercises=False,      gen-
                                              erate_innodoc=False,      in-
                                              put_format='latex+raw_tex',      out-
                                              put_format='markdown',      gener-
                                              ate_innodoc_markdown=False,      de-
                                              bug=False)
```

innoConv (mintmod) runner that spawns a panzer instance.

run()

Setup paths and options and run the panzer command.

Return type `str`

Returns output filename

2.5.5 innoconv_mintmod.utils

Utility module

`innoconv_mintmod.utils.block_wrap(elem, orig_elem)`

Wraps an element in a block if necessary.

If the original element was block panflute expects the return value to be also block. In many places we need to detect this and wrap an inline.

Parameters

- `elem` (`panflute.base.Element`) – Element to be wrapped
- `orig_elem` (`panflute.base.Element`) – Original element

Return type `panflute.base.Element`

Returns elem or elem wrapped in `panflute.elements.Plain`

`innoconv_mintmod.utils.convert_simplification_code(code)`

Convert binary flags to string flags.

`innoconv_mintmod.utils.destringify(string)`

Takes a string and transforms it into list of Str and Space objects.

This function breaks down strings with whitespace. It could be done by calling `parse_fragment()` but doesn't have the overhead invoked.

Example

```
>>> destringify('foo  bar      baz')
[Str(foo), Space, Str(bar), Space, Str(baz)]
```

Parameters `string(str)` – String to transform

Return type `list`

Returns list of `panflute.Str` and `panflute.Space`

innoconv_mintmod.utils.**extract_identifier**(*content*)

Extract identifier from content and remove annotation element.

\MLabel/MDeclareSiteUXID commands that occur within environments are parsed in a child process (e.g. innoconv_mintmod.mintmod_filter.commands.handle_mlabel()). The id attribute can't be set directly as they can't access the whole doc tree. As a workaround they create a fake element and add the identifier.

Parameters **content** (*list*) – List of elements

Return type *str*

Returns identifier (might be None)

innoconv_mintmod.utils.**get_panzer_bin**()

Get path of panzer binary.

innoconv_mintmod.utils.**get_remembered**(*doc*, *key*, *keep=False*)

Retrieve remembered element from the document and forget it.

To remember elements use *remember()*.

Parameters

- **doc** (*panflute.elements.Doc*) – Document where the element is stored
- **key** (*str*) – Key under which element is stored
- **keep** (*bool*) – If value should be kept after retrieving (default=False)

Return type *panflute.base.Element*

Returns The remembered element or *None*

innoconv_mintmod.utils.**log**(*msg_string*, *level='INFO'*)

Log messages when running as a panzer filter.

Parameters

- **msg_string** (*str*) – Message that is logged
- **level** (*str*) – Log level (INFO, WARNING, ERROR OR CRITICAL)

innoconv_mintmod.utils.**parse_cmd**(*text*)

Parse a LaTeX command using regular expressions.

Parses a command like: \foo{bar} {baz}

Parameters **text** (*str*) – String to parse

Return type (*str, list*)

Returns command name and list of command arguments

innoconv_mintmod.utils.**parse_fragment**(*parse_string*, *lang*, *as_doc=False*,
from_format='latex+raw_tex')

Parse a source fragment using panzer.

Parameters

- **parse_string** (*str*) – Source fragment
- **lang** (*str*) – Language code
- **as_doc** (*bool*) – Return elements as *panflute.elements.Doc*
- **from_format** (*str*) – Source format

Return type list of *panflute.base.Element* or *panflute.elements.Doc*

Returns parsed elements

Raises

- `OSError` – if panzer executable is not found
- `RuntimeError` – if panzer recursion depth is exceeded
- `RuntimeError` – if panzer output could not be parsed

`innoconv_mintmod.utils.parse_nested_args(to_parse)`

Parse LaTeX command arguments that can have nested commands. Returns arguments and rest string.

Parses strings like: {bar}{baz{}}rest into [['bar', 'baz{}'], 'rest'].

Parameters `to_parse (str)` – String to parse

Return type (list, str)

Returns parsed arguments and rest string

`innoconv_mintmod.utils.remember(doc, key, elem)`

Rememember an element in the document for later.

To retrieve remembered elements use `get_remembered()`.

Parameters

- `doc` (`panflute.elements.Doc`) – Document where to store the memory
- `key (str)` – Key under which element is stored
- `elem` (`panflute.base.Element`) – Element to remember

`innoconv_mintmod.utils.remove_annotations(doc)`

Remove left-over annotation elements from document.

Parameters `doc (panflute.elements.Doc)` – Document

`innoconv_mintmod.utils.remove_empty_paragraphs(doc)`

Remove empty paragraphs from document.

Parameters `doc (panflute.elements.Doc)` – Document

`innoconv_mintmod.utils.to_inline(elem, classes=[], attributes={})`

Convert any given pandoc element to inline element(s). Some information may be lost.

2.5.6 generate_innodoc

This is the final step to generate innoDoc content from Mintmod input.

- Load pandoc output from single JSON file.
- Generate a section tree from headings.
- Create a mapping between mintmod section IDs and section paths. (a)
- Create a mapping between element IDs and section paths. (b)
- Rewrite all links by using (a) and (b).
- Save individual sections to innoDoc-specific directory structure.
- Generate a `manifest.yml`.
- Removes single JSON file.

```
class generate_innodoc.CreateMapOfIds (sections)
    Create a mapping between link IDs and section path.

    create()
        Create map.

class generate_innodoc.CreateMapOfSectionIds (sections)
    Create mapping between mintmod section id and section path.

    get_map()
        Generate section map.

class generate_innodoc.ExtractSectionTree (nodes, level)
    Generate section tree from a flat document structure.

    create_section()
        Save a section.

    get_tree()
        Generate and return tree.

class generate_innodoc.GenerateInnodoc (debug=False)
    Main class for generate_innodoc postflight filter.

    LANGKEY = 'languages'
        Languages key in manifest.yml

    main()
        Post-flight script entry point.

    update_manifest (title, outdir)
        Update manifest.yml file.

        If it doesn't exist it will be created.

generate_innodoc.MAX_LEVELS = 3
    Max. depth of headers to consider when splitting sections

class generate_innodoc.PostprocessLinks (sections, section_map, id_map)
    Postprocess all links to work with new section structure.

    process()
        Rewrite links.

class generate_innodoc.WriteSections (sections, outdir_base, output_format)
    Write sections to individual files and remove content from TOC tree.

    PANDOC_TIMEOUT = 120
        Timeout for pandoc process

    write_sections()
        Write all sections.
```

**CHAPTER
THREE**

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

g

generate_innodoc, 19

i

innoconv_mintmod, 1
innoconv_mintmod.constants, 8
innoconv_mintmod.errors, 9
innoconv_mintmod.mintmod_filter, 9
innoconv_mintmod.mintmod_filter.commands,
 9
innoconv_mintmod.mintmod_filter.elements,
 14
innoconv_mintmod.mintmod_filter.environments,
 15
innoconv_mintmod.mintmod_filter.filter_action,
 16
innoconv_mintmod.mintmod_filter.math,
 16
innoconv_mintmod.runner, 17
innoconv_mintmod.utils, 17

INDEX

B

block_wrap() (in module `inoconv_mintmod.utils`),
17

C

Commands (class in `inoconv_mintmod.mintmod_filter.commands`),
9

COMMANDS_IRREGULAR (in module `inoconv_mintmod.constants`), 8

convert_simplification_code() (in module `inoconv_mintmod.utils`), 17

create() (generate_innodoc.CreateMapOfIds
method), 20

create_content_box() (in module `inoconv_mintmod.mintmod_filter.elements`),
14

create_header() (in module `inoconv_mintmod.mintmod_filter.elements`),
14

create_image() (in module `inoconv_mintmod.mintmod_filter.elements`),
14

create_section() (generate_innodoc.ExtractSectionTree
method), 20

CreateMapOfIds (class in generate_innodoc), 19

CreateMapOfSectionIds (class in generate_innodoc), 20

D

DEFAULT_LANGUAGE_CODE (in module `inoconv_mintmod.constants`), 8

DEFAULT_OUTPUT_DIR_BASE (in module `inoconv_mintmod.constants`), 8

DEFAULT_OUTPUT_FORMAT (in module `inoconv_mintmod.constants`), 8

destringify() (in module `inoconv_mintmod.utils`),
17

E

ELEMENT_CLASSES (in module `inoconv_mintmod.constants`), 8

`conv_mintmod.constants`), 8

ENCODING (in module `inoconv_mintmod.constants`), 8

Environments (class in `inoconv_mintmod.mintmod_filter.environments`),
15

extract_identifier() (in module `inoconv_mintmod.utils`), 17

ExtractSectionTree (class in `generate_innodoc`),
20

F

filter() (`inoconv_mintmod.mintmod_filter.filter_action.MintmodFilter`
method), 16

G

generate_innodoc
module, 19

GenerateInnodoc (class in `generate_innodoc`), 20

get_map() (generate_innodoc.CreateMapOfSectionIds
method), 20

get_panzer_bin() (in module `inoconv_mintmod.utils`), 18

get_remembered() (in module `inoconv_mintmod.utils`), 18

get_tree() (generate_innodoc.ExtractSectionTree
method), 20

H

handle_bigskip() (inoconv_mintmod.mintmod_filter.commands.Commands
method), 9

handle_clearpage() (inoconv_mintmod.mintmod_filter.commands.Commands
method), 9

handle_glqq() (inoconv_mintmod.mintmod_filter.commands.Commands
method), 9

handle_grqq() (inoconv_mintmod.mintmod_filter.commands.Commands
method), 9

handle_highlight() (inoconv_mintmod.mintmod_filter.commands.Commands
method), 9

```

handle_hspace()                               (inno-          conv_mintmod.mintmod_filter.environments.Environments
                                              conv_mintmod.mintmod_filter.commands.Commands   method), 15
                                              method), 9                                     handle_mexercise()                           (inno-
                                              (inno-          conv_mintmod.mintmod_filter.environments.Environments
                                              conv_mintmod.mintmod_filter.environments.Environments   method), 15
                                              method), 15                                     handle_mexercisecollection()                (inno-
                                              (inno-          conv_mintmod.mintmod_filter.environments.Environments
                                              conv_mintmod.mintmod_filter.commands.Commands   method), 15
                                              method), 9                                     handle_mexerciseitems()                  (inno-
                                              (inno-          conv_mintmod.mintmod_filter.environments.Environments
                                              conv_mintmod.mintmod_filter.environments.Environments   method), 15
                                              method), 15                                     handle_mexercises()                      (inno-
                                              (inno-          conv_mintmod.mintmod_filter.environments.Environments
                                              conv_mintmod.mintmod_filter.commands.Commands   method), 15
                                              method), 9                                     handle_mexperiment()                     (inno-
                                              (inno-          conv_mintmod.mintmod_filter.environments.Environments
                                              conv_mintmod.mintmod_filter.commands.Commands   method), 15
                                              method), 10                                     handle_mextlink()                         (inno-
                                              (in       module      inno-          conv_mintmod.mintmod_filter.commands.Commands
                                              conv_mintmod.mintmod_filter.math), 16           method), 10
                                              method), 10                                     handle_mformelzoomhint()                 (inno-
                                              (inno-          conv_mintmod.mintmod_filter.commands.Commands   method), 10
                                              conv_mintmod.mintmod_filter.commands.Commands
                                              method), 10                                     handle_mglobalchaptertag()                (inno-
                                              (inno-          conv_mintmod.mintmod_filter.environments.Environments
                                              conv_mintmod.mintmod_filter.commands.Commands   method), 15
                                              method), 15                                     handle_mglobalconftag()                  (inno-
                                              (inno-          conv_mintmod.mintmod_filter.commands.Commands   method), 10
                                              conv_mintmod.mintmod_filter.commands.Commands
                                              method), 10                                     handle_mglobaldatatag()                  (inno-
                                              (inno-          conv_mintmod.mintmod_filter.commands.Commands   method), 10
                                              conv_mintmod.mintmod_filter.commands.Commands
                                              method), 10                                     handle_mglobalfavotag()                  (inno-
                                              (inno-          conv_mintmod.mintmod_filter.commands.Commands   method), 10
                                              conv_mintmod.mintmod_filter.commands.Commands
                                              method), 10                                     handle_mgloballocationtag()              (inno-
                                              (inno-          conv_mintmod.mintmod_filter.environments.Environments
                                              conv_mintmod.mintmod_filter.commands.Commands   method), 15
                                              method), 15                                     handle_mgloballogintag()                  (inno-
                                              (inno-          conv_mintmod.mintmod_filter.commands.Commands   method), 10
                                              conv_mintmod.mintmod_filter.commands.Commands
                                              method), 10                                     handle_mgloballogouttag()                (inno-
                                              (inno-          conv_mintmod.mintmod_filter.commands.Commands   method), 10
                                              conv_mintmod.mintmod_filter.commands.Commands
                                              method), 10                                     handle_mglobalsearchtag()                (inno-
                                              (inno-          conv_mintmod.mintmod_filter.commands.Commands   method), 11
                                              conv_mintmod.mintmod_filter.commands.Commands
                                              method), 11                                     handle_mglobalstart()                   (inno-
                                              (inno-          conv_mintmod.mintmod_filter.commands.Commands   method), 10
                                              conv_mintmod.mintmod_filter.commands.Commands
                                              method), 10                                     handle_mglobalstesttag()                (inno-
                                              (inno-          conv_mintmod.mintmod_filter.commands.Commands   method), 11
                                              conv_mintmod.mintmod_filter.commands.Commands
                                              method), 10                                     handle_mexample()                      (inno-
                                              (inno-          conv_mintmod.mintmod_filter.commands.Commands   method), 11
                                              conv_mintmod.mintmod_filter.commands.Commands
                                              method), 11

```

<i>conv_mintmod.mintmod_filter.commands.Commands method), 11</i>	<i>(inno- handle_modstextbf () conv_mintmod.mintmod_filter.commands.Commands method), 12</i>	<i>(inno- conv_mintmod.mintmod_filter.commands.Commands method), 12</i>
<i>handle_mgraphicssolo () conv_mintmod.mintmod_filter.commands.Commands method), 11</i>	<i>(inno- handle_mppragma () conv_mintmod.mintmod_filter.commands.Commands method), 12</i>	<i>(inno- conv_mintmod.mintmod_filter.commands.Commands method), 12</i>
<i>handle_mgroupbutton () conv_mintmod.mintmod_filter.commands.Commands method), 11</i>	<i>(inno- handle_mprintindex () conv_mintmod.mintmod_filter.environments.Environments method), 15</i>	<i>(inno- conv_mintmod.mintmod_filter.commands.Commands method), 12</i>
<i>handle_mhint () conv_mintmod.mintmod_filter.environments.Environments method), 15</i>	<i>(inno- handle_mpullsite () conv_mintmod.mintmod_filter.commands.Commands method), 11</i>	<i>(inno- conv_mintmod.mintmod_filter.commands.Commands method), 12</i>
<i>handle_mindex () conv_mintmod.mintmod_filter.commands.Commands method), 11</i>	<i>(inno- handle_mquestiongroup () conv_mintmod.mintmod_filter.environments.Environments method), 15</i>	<i>(inno- conv_mintmod.mintmod_filter.commands.Commands method), 16</i>
<i>handle_minfo () conv_mintmod.mintmod_filter.environments.Environments method), 15</i>	<i>(inno- handle_mref () conv_mintmod.mintmod_filter.commands.Commands method), 11</i>	<i>(inno- conv_mintmod.mintmod_filter.commands.Commands method), 12</i>
<i>handle_minputhint () conv_mintmod.mintmod_filter.commands.Commands method), 11</i>	<i>(inno- handle_msection () conv_mintmod.mintmod_filter.environments.Environments method), 16</i>	<i>(inno- conv_mintmod.mintmod_filter.commands.Commands method), 12</i>
<i>handle_mintro () conv_mintmod.mintmod_filter.environments.Environments method), 16</i>	<i>(inno- handle_msectionstart () conv_mintmod.mintmod_filter.commands.Commands method), 11</i>	<i>(inno- conv_mintmod.mintmod_filter.environments.Environments method), 16</i>
<i>handle_mlcheckbox () conv_mintmod.mintmod_filter.commands.Commands method), 11</i>	<i>(inno- handle_msetpoints () conv_mintmod.mintmod_filter.commands.Commands method), 12</i>	<i>(inno- conv_mintmod.mintmod_filter.commands.Commands method), 12</i>
<i>handle_mlfunctionquestion () conv_mintmod.mintmod_filter.commands.Commands method), 11</i>	<i>(inno- handle_msetsectionid () conv_mintmod.mintmod_filter.commands.Commands method), 11</i>	<i>(inno- conv_mintmod.mintmod_filter.commands.Commands method), 12</i>
<i>handle_mlintervalquestion () conv_mintmod.mintmod_filter.commands.Commands method), 11</i>	<i>(inno- handle_msetssubject () conv_mintmod.mintmod_filter.commands.Commands method), 11</i>	<i>(inno- conv_mintmod.mintmod_filter.commands.Commands method), 12</i>
<i>handle_mlparsedquestion () conv_mintmod.mintmod_filter.commands.Commands method), 11</i>	<i>(inno- handle_msref () conv_mintmod.mintmod_filter.commands.Commands method), 11</i>	<i>(inno- conv_mintmod.mintmod_filter.commands.Commands method), 12</i>
<i>handle_mlquestion () conv_mintmod.mintmod_filter.commands.Commands method), 11</i>	<i>(inno- handle_msubject () conv_mintmod.mintmod_filter.commands.Commands method), 11</i>	<i>(inno- conv_mintmod.mintmod_filter.commands.Commands method), 12</i>
<i>handle_ml simplifyquestion () conv_mintmod.mintmod_filter.commands.Commands method), 11</i>	<i>(inno- handle_msubsection () conv_mintmod.mintmod_filter.commands.Commands method), 11</i>	<i>(inno- conv_mintmod.mintmod_filter.commands.Commands method), 13</i>
<i>handle_ml specialquestion () conv_mintmod.mintmod_filter.commands.Commands method), 12</i>	<i>(inno- handle_msubsubsection () conv_mintmod.mintmod_filter.commands.Commands method), 12</i>	<i>(inno- conv_mintmod.mintmod_filter.commands.Commands method), 13</i>
<i>handle_mmmodstartbox () conv_mintmod.mintmod_filter.commands.Commands method), 12</i>	<i>(inno- handle_msubsubsectionx () conv_mintmod.mintmod_filter.commands.Commands method), 13</i>	<i>(inno- conv_mintmod.mintmod_filter.commands.Commands method), 13</i>
<i>handle_mnref () conv_mintmod.mintmod_filter.commands.Commands method), 12</i>	<i>(inno- handle_msubsubsubsectionx () conv_mintmod.mintmod_filter.commands.Commands method), 13</i>	<i>(inno- conv_mintmod.mintmod_filter.commands.Commands method), 13</i>
<i>handle_modsemph ()</i>	<i>(inno- handle_mtest ()</i>	<i>(inno-</i>

```

    conv_mintmod.mintmod_filter.environments.Environments conv_mintmod.mintmod_filter.commands.Commands
    method), 16 method), 14

handle_mtikzauto() (inno-
    conv_mintmod.mintmod_filter.commands.Commands
    method), 13 INDEX_LABEL_PREFIX (in module inno-
handle_mtitle() (inno- conv_mintmod.constants), 8
    conv_mintmod.mintmod_filter.commands.Commands innoconv_mintmod
    method), 13 module, 1
handle_mugraphics() (inno- innoconv_mintmod.constants
    conv_mintmod.mintmod_filter.commands.Commands module, 8
    method), 13 innoconv_mintmod.errors
handle_mugraphicssolo() (inno- module, 9
    conv_mintmod.mintmod_filter.commands.Commands innoconv_mintmod.mintmod_filter
    method), 13 module, 9
handle_mvvideo() (inno- innoconv_mintmod.mintmod_filter.commands
    conv_mintmod.mintmod_filter.commands.Commands module, 9
    method), 13 innoconv_mintmod.mintmod_filter.elements
handle_mwatermarksettings() (inno- module, 14
    conv_mintmod.mintmod_filter.commands.Commands innoconv_mintmod.mintmod_filter.environments
    method), 13 module, 15
handle_mxcontent() (inno- innoconv_mintmod.mintmod_filter.filter_action
    conv_mintmod.mintmod_filter.environments.Environments module, 16
    method), 16 innoconv_mintmod.mintmod_filter.math
handle_mxinfo() (inno- module, 16
    conv_mintmod.mintmod_filter.environments.Environments innoconv_mintmod.runner
    method), 16 module, 17
handle_myoutubevideo() (inno- innoconv_mintmod.utils
    conv_mintmod.mintmod_filter.commands.Commands module, 17
    method), 13 InnoconvRunner (class in inno-
handle_mzahl() (inno- conv_mintmod.runner), 17
    conv_mintmod.mintmod_filter.commands.Commands
    method), 13 L
handle_mzzxyzhltrennzeichen() (inno- LANGKEY (generate_innodoc.GenerateInnodoc at-
    conv_mintmod.mintmod_filter.commands.Commands tribute), 20
    method), 13 LANGUAGE_CODES (in module inno-
handle_newline() (inno- conv_mintmod.constants), 8
    conv_mintmod.mintmod_filter.commands.Commands log() (in module innoconv_mintmod.utils), 18
    method), 13
handle_newpage() (inno- M
    conv_mintmod.mintmod_filter.commands.Commands main() (generate_innodoc.GenerateInnodoc method),
    method), 13 20
handle_noindent() (inno- MAX_LEVELS (in module generate_innodoc), 20
    conv_mintmod.mintmod_filter.commands.Commands MINTMOD_SUBJECTS (in module inno-
    method), 14 conv_mintmod.constants), 8
handle_quad() (inno- MintmodFilterAction (class in inno-
    conv_mintmod.mintmod_filter.commands.Commands conv_mintmod.mintmod_filter.filter_action),
    method), 14 16
handle_smallskip() (inno- module
    conv_mintmod.mintmod_filter.commands.Commands generate_innodoc, 19
    method), 14 innoconv_mintmod, 1
handle_special() (inno- innoconv_mintmod.constants, 8
    conv_mintmod.mintmod_filter.commands.Commands innoconv_mintmod.errors, 9
    method), 14 innoconv_mintmod.mintmod_filter, 9
handle_vspace() (inno-

```

innoconv_mintmod.mintmod_filter.commands, **T**
 9 to_inline() (in module innoconv_mintmod.utils), 19
innoconv_mintmod.mintmod_filter.elements, **U**
 14
innoconv_mintmod.mintmod_filter.environments, **U**
 15 update_manifest()
innoconv_mintmod.mintmod_filter.filter_action, **U**
 16 ate_innодoc.GenerateInnодoc
 (gener-
 method),
innoconv_mintmod.mintmod_filter.math, **W**
 16 write_sections()
innoconv_mintmod.runner, **W**
 17 ate_innодoc.WriteSections method), 20
innoconv_mintmod.utils, **W**
 17 WriteSections (class in generate_innодoc), 20

O

OUTPUT_FORMAT_CHOICES (in module innoconv_mintmod.constants), 8
OUTPUT_FORMAT_EXT_MAP (in module innoconv_mintmod.constants), 8

P

PANDOC_TIMEOUT (generate_innодoc.WriteSections attribute), 20
PANZER_SUPPORT_DIR (in module innoconv_mintmod.constants), 8
parse_args() (innoconv_mintmod.mintmod_filter.elements.Question static method), 14
parse_cmd() (in module innoconv_mintmod.utils), 18
parse_fragment() (in module innoconv_mintmod.utils), 18
parse_nested_args() (in module innoconv_mintmod.utils), 19
ParseError, 9
PostprocessLinks (class in generate_innодoc), 20
process() (generate_innодoc.PostprocessLinks method), 20

Q

Question (class in innoconv_mintmod.mintmod_filter.elements), 14

R

REGEX_PATTERNS (in module innoconv_mintmod.constants), 8
remember() (in module innoconv_mintmod.utils), 19
remove_annotations() (in module innoconv_mintmod.utils), 19
remove_empty_paragraphs() (in module innoconv_mintmod.utils), 19
ROOT_DIR (in module innoconv_mintmod.constants), 8
run() (innoconv_mintmod.runner.InnoconvRunner method), 17