

YAWP 0.7.1 MANUAL

Yet Another Word Processor

2022-06-25

Carlo Alessandro Verre

carlo.alessandro.verre@gmail.com

I sound my barbaric yawp over the roofs of the world

Walt Whitman

CONTENTS

• 1.	Forewords	3
• 1.1.	What'S Yawp?	3
• 1.2.	Installation	4
• 1.3.	Messages	4
• 1.4.	Glossary	4
• 2.	Usage Modes	6
• 2.1.	Format Mode	6
• 2.2.	No-Format Mode	6
• 2.3.	Undo Mode	6
• 3.	Formatting	7
• 3.1.	Justification	7
• 3.2.	Best Practices	8
• 3.3.	Python Files	9
• 4.	Chapters	12
• 4.1.	Unnamed Chapter	12
• 4.2.	Numbered Chapters	12
• 4.3.	Automatic Chapters	14
• 4.3.1.	Contents Chapter	14
• 4.3.2.	Figures Chapter	15
• 4.3.3.	Index Chapter	16
• 5.	Paging	18
• 6.	Graphics	20
• 7.	Pdf	23
• 7.1.	Pdf Exporting	23
• 7.2.	Pdf Corrections	24
• 8.	Afterwords	28
• 8.1.	Acronyms	28
• 8.2.	Characters	28
• 8.3.	Credits	28
• 8.4.	History	29
• 8.5.	License	29
• 9.	Arguments	31
•	Figures	32
•	Index	33

1. FOREWORDS

1.1. WHAT'S YAWP?

The name "yawp" here means Yet Another Word Processor, and yawp is an automatic (batch, not interactive) word processor for plain text files, with PDF export. If you really need all the features of a full-fledged WYSIWYG word processor as LibreOffice Writer, yawp is not for you. But if you just want to create a simple quick-and-dirty no-frills document, with yawp you can:

- edit a text file by your favorite editor
- run yawp in order to:
 - backup read format and rewrite the text file
 - export the text file into a PDF file
 - view the PDF file for check or print
- possibly go back to the editor and update the text file, or finish

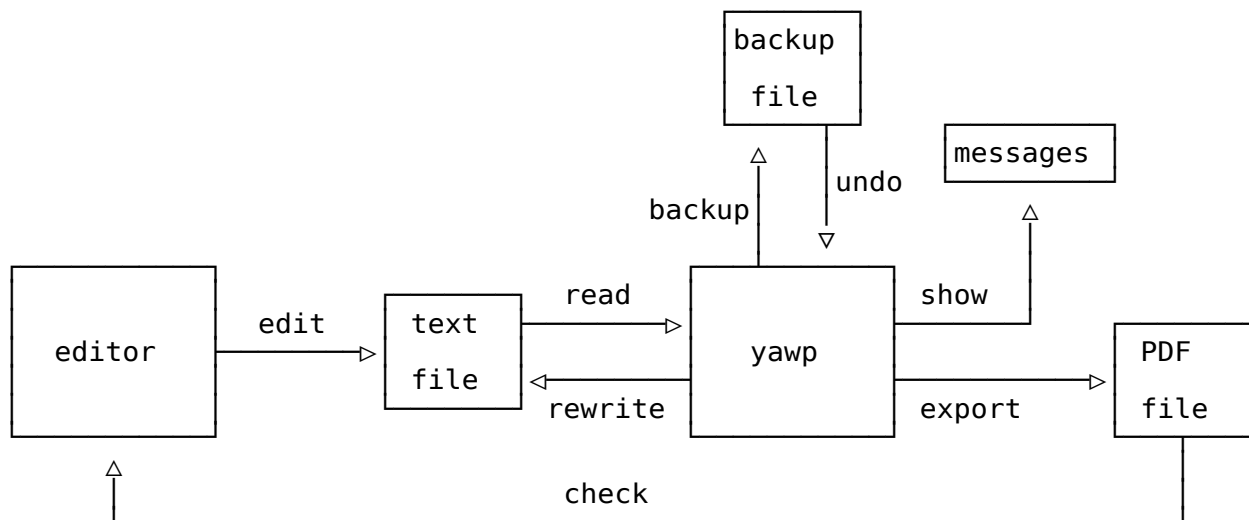


Figure 1.1.a. Usage Flow Diagram

Main features are:

- yawp processes in place a single text file, hereinafter referred to simply as "the file"
- yawp before processing creates a timestamped backup of the file, allowing undo operation (see 2.3.)
- yawp processing is driven by the text in the file and by arguments only, not by commands or tags embedded in text
- yawp justifies (see 3.1.) text at left and right in:
 - unindented paragraphs
 - dot-marked indented paragraphs (as this one)
- yawp accepts unjustified pictures (as schemas, tables and code examples) freely intermixed with text
- yawp adopts an ad hoc policy for Python files, formatting the docstrings but not the Python code (see 3.3.)
- yawp performs automatic multi-level renumbering of chapters (see 4.2.) and inserts an automatic contents chapter in the file (see 4.3.1.)
- yawp performs automatic renumbering of figure captions and inserts an automatic figures chapter in the file (see 4.3.2.)
- yawp recognizes relevant subjects (quoted by "'") and inserts an automatic index chapter in the file (see 4.3.3.)
- yawp cuts the file in pages, by inserting 2-lines page headers (see 5.)
- yawp also has some limited graphic features, you can sketch pictures with segments (by "'") and arrowheads (by '^'), yawp redraws them by suitable graphic characters (as in Figure 1.1.a. above, see 6.)
- yawp exports the resulting lines in PDF format, with control over character size and page layout, and lets you view the generated PDF file, allowing preview and printing (see 7.1.)
- yawp corrects errors made by CUPS-PDF about font size and page margins, you can use default corrections or redefine them by yawp.cfg (see 7.2.)
- yawp is "stable", namely if after a yawp execution you run yawp again on

the same file with the same arguments, the file content doesn't change (except date and time in page headers, see 5.)

Believe it or not, everything has been kept as simple as possible. In order to use yawp, you need to know how it works, but for a quick start you can skip chapters 3.2. and 7.2.

As an example, this "Yawp Manual" has been created as yawp.pdf from yawp.txt by typing:

```
| $ yawp -v -w 80 -J -E 'Yawp 0.7.1 Manual' -X yawp.txt
```

Other examples are scattered below.

1.2. INSTALLATION

CUPS-PDF provides a PDF writer backend to CUPS, and yawp needs it to export the file in PDF format. For example, if your Linux belongs to the Debian family, type:

```
| $ sudo apt-get -y update  
| $ sudo apt-get -y install printer-driver-cups-pdf
```

If you don't have pip, type:

```
| $ sudo apt-get -y install python3-pip
```

If you type at terminal:

```
| $ pip3 install --upgrade yawp
```

this command will:

- install current version of yawp if not present
- upgrade yawp to the current version if already installed

If you see a message like this:

```
| WARNING: The script yawp is installed in ... which is not on PATH.
```

don't worry, a reboot should fix the problem.

1.3. MESSAGES

During execution yawp can write three kinds of messages:

- an "information message" says what's going on
- a "warning message" (starting with 'WARNING:') says what may be wrong, and yawp processing continues
- an "error message" (starting with 'ERROR:') says what's wrong, file backup and rewriting don't take place and yawp processing is terminated

Information messages and warning messages are displayed only if -v argument is on.

All three types of messages are written on stderr.

When applicable, warning and error messages are preceded by position and content of the offending line in the file, or in yawp.cfg configuration file.

The file is read in RAM and then processed several times, so if the file contains more than one error, the displayed error message doesn't necessarily indicate the first error in the file.

1.4. GLOSSARY

Some terms here have a different meaning from that commonly in use.

To "shrink" a character string means:

- to strip away all leading and trailing blanks, and
- to reduce each internal group of consecutive blanks to a single blank

For instance, the string:

```
'  aAa  bBb  "cCc"  '
```

if shrunk, becomes:

```
'aAa bBb "cCc"'
```

To "uppercase" a character string means to uppercase all characters in it, except characters quoted by double quote '"', which remain unchanged.

For instance, the string:

```
'aAa bBb "cCc"'
```

if uppercased, becomes:

```
'AAA BBB "cCc"'
```

To "titlecase" a character string means to uppercase the first character of each word in it, and to lowercase all the others, except the characters quoted by double quote '"', which remain unchanged.

For instance, the string:

```
'aAa bBb "cCc"'
```

if titlecased, becomes:

```
'Aaa Bbb "cCc"'
```

Two strings are said to be "equivalent" if, shrunk and uppercased, they become equal. For instance the strings:

```
x = '  aAa  bBb  "cCc"  '
y = '  aAa  bBb  "cCc"  '
```

are equivalent because:

```
uppercase(shrink(x)) == 'AAA BBB "cCc"' == uppercase(shrink(y))
```

Other terms will be explained when they are used.

2. USAGE MODES

General behaviour is controlled by the following arguments:

- "-h, --help": show a help message and exit
- "-H, --manual": view this yawp-generated PDF Yawp Manual and exit
- "-V, --version": show program's version number and exit
- "-v, --verbose": display information and warning messages on stderr

Two arguments, -N and -U:

- "-N, --no-format": run in no-format mode (default: run in format mode)
- "-U, --undo": run in undo mode (default: run in format mode)

define three usage modes:

- format mode (-N off and -U off, see 2.1.)
- no-format mode (-N on, see 2.2.)
- undo mode (-U on, see 2.3.)

To turn on both -U and -N is not allowed.

In all three usage modes, the file can finally be exported and viewed in PDF format (see 7.1.).

2.1. FORMAT MODE

If neither -N nor -U is on, yawp runs in "format mode":

- the file is read, eliminating:
 - page header lines
 - "horizontal tab" '\t' in lines, each replaced by four blanks
 - trailing blanks in lines
- the file content is formatted (see 3. 4. and 5.)
- if -g is on, pictures are redrawn (see 6.)
- the file is backed up in a timestamped copy
- the file is rewritten

A text file is a sequence of lines, separated by line terminators. Both in format and in no-format mode, file reading is in "universal newlines" mode, so three line terminators are accepted:

- "line feed" = '\n' (Unix-Linux-Apple)
- "carriage return" + "line feed" = '\r\n' (MS-Windows)
- "carriage return" = '\r' (Apple old MacOS pre-OSX)

When the file is rewritten, the line terminator is always '\n'.

2.2. NO-FORMAT MODE

If -N is on, yawp runs in "no-format mode":

- the file is read, eliminating:
 - "horizontal tab" '\t' in lines, each replaced by four blanks
 - trailing blanks in lines
- if -g is on, pictures are redrawn (see 6.)
- the file is backed up in a timestamped copy
- the file is rewritten

Input page header lines are kept in output.

2.3. UNDO MODE

If -U is on, yawp runs in "undo mode". The file is recovered from the most recent timestamped backup. The -g argument has no effect.

BEWARE: undo operation can't be undone, the file goes back to the penultimate version and the last one is lost.

3. FORMATTING

3.1. JUSTIFICATION

As said before, in format mode each page "header line" inserted by yawp (see 5.) is removed from the input file. Each remaining "body line" belongs to one of these four kinds:

- a line is an "empty line" if it contains no characters (all trailing blanks in input lines are stripped away, hence every input line containing only blanks becomes an empty line)
- otherwise a line is a "dot line" if it's not empty and starts with:
 - zero or more blanks
 - a "dot character", which can be:
 - a "black small circle" '.', or
 - a "decimal point" '.' (always replaced by '.' on output)
 - a blank
- otherwise a line is an "indented line" if it's not empty and starts with a blank
- otherwise a line is an "unindented line" if starts with a non-blank character

The formatting algorithm, driven by the input lines, oscillates between two states:

- "picture state", where input lines are directly written out as they are
- "text state", where input lines are accumulated into a paragraph buffer for further justification and writing at paragraph end

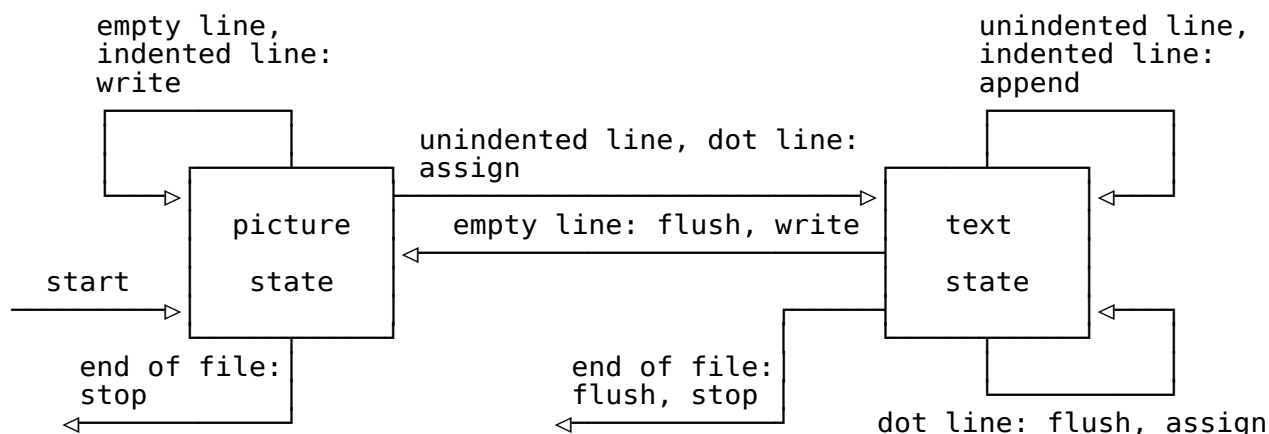


Figure 3.1.a. Formatting State Diagram

The picture state is the initial state. In this state, if the input is:

- an empty line or an indented line: the line is written out as is
- an unindented line: text state is entered, an "unindented paragraph" begins, the line is shrunk and assigned to the paragraph buffer, paragraph left indentation is set to zero
- a dot line: text state is entered, an "indented paragraph" begins, the line is shrunk and assigned to the paragraph buffer, paragraph left indentation is set to the position of initial dot character plus two
- at end of input file: processing is ended

When we are in text state, if the input line is:

- an empty line: the paragraph buffer is flushed (justified, written out and emptied), state goes back to picture state, the empty line is written out
- an indented or unindented line: the line is shrunk and appended to the paragraph buffer
- a dot line: paragraph buffer is flushed, a new paragraph is started, the line is shrunk and assigned to the paragraph buffer, paragraph left indentation is set to the position of initial dot plus two
- at end of input file: paragraph buffer is flushed, processing is ended

Max length of text lines can be controlled by:

- "-w, --chars-per-line": line width in characters per line (default: 0 = automatic)

-w and -W (character width, see 7.1.) can be zero hence "automatic", namely:

- if -w is automatic and -W is not, -w is computed from -W
- if -W is automatic and -w is not, -W is computed from -w
- if both -w and -W are automatic,
 - -w is deduced from max body line length in file (page headers are not considered)
 - -W is computed from -w

Indentation of indented paragraphs can't be greater than half of -w argument, whether set by user or automatically computed.

As an example of -W computed from -w, we impose one character per line only:

```
| $ echo BANNER >banner.txt
|
| $ yawp -v -N -w 1 -X banner.txt
| Correct: '~/yawp/yawp.cfg' not found, default corrections
| Correct: -L 43.325pt = 0.602in = 15.284mm = 1.528cm
| Correct: -R 40.57pt = 0.563in = 14.312mm = 1.431cm
| Correct: -T 53.814pt = 0.747in = 18.985mm = 1.898cm
| Correct: -B 15.991pt = 0.222in = 5.641mm = 0.564cm
| Read: yawp <-- '~/banner.txt'
|       0 header lines, max 0 chars per line, 1 page
|       1 body line, max 6 chars per line
|       1 total line, max 6 chars per line
| Compute: -W 481.89pt = 6.693in = 170.0mm = 17.0cm
| Correct: -W 508.999pt = 7.069in = 179.564mm = 17.956cm
| Compute: char height 803.15pt = 11.155in = 283.333mm = 28.333cm
| Correct: char height 851.173pt = 11.822in = 300.275mm = 30.027cm
| Backup: '~/banner.txt' --> '~/banner-2022.06.25-18.47.48.txt'
| Rewrite: yawp --> '~/banner.txt'
|         0 header lines, max 0 chars per line, 1 page
|         1 body line, max 6 chars per line
|         1 total line, max 6 chars per line
| Export: '~/banner.txt' --> '~/banner.pdf'
|
| $ cat banner.txt
| BANNER
```

Figure 3.1.b. Example With "-N -w 1"

Now banner.txt is not altered because the -N argument, but the exported banner.pdf contains the string 'BANNER', one huge character per page.

Line justification in text is controlled by -l argument:

- "-l, --left-only": justify text lines at left only (default: at left and right)

After formatting:

- groups of consecutive empty lines between text paragraphs are reduced to a single empty line
- groups of consecutive empty lines in contact with a picture are left unchanged

3.2. BEST PRACTICES

Now we can define some "best practices" to follow when editing the file.

Unindented paragraphs should be:

- preceded by an empty line
- started by an unindented line
- continued by indented or unindented lines

- ended by an empty line (better) or by a dot line

Indented paragraphs should be:

- preceded by a line of any kind
- initiated by a dot line (better if starting with 4-8-12-... blanks)
- continued by indented or unindented lines
- ended by an empty line or by another dot line

Pictures should be:

- preceded by an empty line
- initiated and continued by indented lines only
- ended by an empty line (better) or by an unindented line or by a dot line

3.3. PYTHON FILES

Formatting of Python files deserves a special treatment. If the file name ends with '.py' extension, then we suppose the file is a Python source, hence we are interested to format docstrings and not Python code. So the formatting function is alternatively turned on and off by switch lines. A "switch line" is a line containing a `"""` string.

Notice that yawp never formats switch lines, formatting takes place from the line after the start switch line until the line before the next stop switch line.

So your Python file must follow some simple rules:

- docstrings to be formatted must start and ended by `"""` and not `''''''`
- long strings not to be formatted must start and end with `''''''` and not `"""`
- a `"""` inside a string can be coded for instance as `'\'''\''`, so the containing line will not be identified as a switch line

BEWARE: An error in switch lines could format and destroy your Python code. A preliminary check prints an error message and stops execution before file formatting if the total number of switch lines is odd. This should intercept many errors, anyway after yawp processing check the result and if needed go back to previous version by -U. An example:

```

$ cat pycode.py
#!/usr/bin/python3

'''start switch line is not formatted
This is an unindented paragraph.

This is another unindented paragraph.
This is another unindented paragraph.
This is another unindented paragraph.

    This is a picture,
        it remains
    as is.

    • This is an indented paragraph.
This is an indented paragraph.
    This is an indented paragraph.
        • This is another indented paragraph.
    This is another indented paragraph.
This is another indented paragraph.
''' # stop switch line is not formatted

def double(x): # python code is not formatted
    '''start switch line is not formatted
This is another unindented paragraph.
This is another unindented paragraph.
This is another unindented paragraph.
''' # stop switch line is not formatted
    return x + x # python code is not formatted

$ yawp -v -w 32 pycode.py
Correct: '~/.yawp/yawp.cfg' not found, default corrections
Correct: -L 43.325pt = 0.602in = 15.284mm = 1.528cm
Correct: -R 40.57pt = 0.563in = 14.312mm = 1.431cm
Correct: -T 53.814pt = 0.747in = 18.985mm = 1.898cm
Correct: -B 15.991pt = 0.222in = 5.641mm = 0.564cm
Read: yawp <-- '~/pycode.py'
    0 header lines, max 0 chars per line, 1 page
    28 body lines, max 47 chars per line
    28 total lines, max 47 chars per line
Compute: -W 15.059pt = 0.209in = 5.312mm = 0.531cm
Correct: -W 15.906pt = 0.221in = 5.611mm = 0.561cm
Compute: char height 25.098pt = 0.349in = 8.854mm = 0.885cm
Correct: char height 26.599pt = 0.369in = 9.384mm = 0.938cm
Backup: '~/pycode.py' --> '~/pycode-2022.06.25-18.47.57.py'
Rewrite: yawp --> '~/pycode.py'
    0 header lines, max 0 chars per line, 1 page
    34 body lines, max 47 chars per line
    34 total lines, max 47 chars per line

```

```
$ cat pycode.py
#!/usr/bin/python3

'''start switch line is not formatted
This is an unindented paragraph.

This is another unindented
paragraph. This is another
unindented paragraph. This is
another unindented paragraph.

    This is a picture,
        it remains
    as is.

    • This is an indented
      paragraph. This is an
      indented paragraph. This
      is an indented paragraph.
      • This is another
        indented paragraph.
        This is another
        indented paragraph.
        This is another
        indented paragraph.
''' # stop switch line is not formatted

def double(x): # python code is not formatted
    '''start switch line is not formatted
    This is another unindented
    paragraph. This is another
    unindented paragraph. This is
    another unindented paragraph.
    ''' # stop switch line is not formatted
    return x + x # python code is not formatted
```

Figure 3.3.a. Example With A Python File

Inserting page headers in a Python file doesn't make sense, so if the file is a Python file then `-j` and `-J` (see 5.) are silently turned off.

4. CHAPTERS

The file can be partitioned in "chapters" by "chapter lines". A chapter can be:

- the "unnamed chapter", the first one, from first line until first chapter line (see 4.1.)
- a "numbered chapter", as this one (see 4.2)
- an "automatic chapter", automatically emptied and refilled (see 4.3.)
 - the "contents chapter", the list of chapters (see 4.3.1.)
 - the "figures chapter", the list of figure captions (see 4.3.2.)
 - the "index chapter", the list of subjects (see 4.3.3.)

CHAPTER	AUTOMATIC	HOW MANY	WHERE	CHAPT LINE	IN CONTENTS	EJECT IF -J
unnamed	no	max one	file begin	no	no	no
numbered	no	no limit	everywhere	yes	yes	level-1 only
contents	yes	max one	everywhere	yes	no	yes
figures	yes	max one	everywhere	yes	yes	yes
index	yes	max one	everywhere	yes	yes	yes

Figure 4.a. Chapter Kinds

All chapters, except the numbered ones, can appear no more than once in the file.

All chapters, except the unnamed one, may appear everywhere and in any order, and are started by a chapter line.

All chapters, except the unnamed one and the contents one itself, are listed in the contents chapter.

All chapters, except the unnamed one and the numbered ones with level greater than one, trigger a page break if argument -J is turned on (see 5.).

Chapter lines share some properties. Each chapter line:

- is the file first line or is preceded by an empty line
- is the file last line or is followed by an empty line
- is a not empty line and doesn't start with blank

Therefore we can say that a chapter line must be a one-line unindented text paragraph.

4.1. UNNAMED CHAPTER

As said before, the unnamed chapter is the first of the file, and goes from the first line (included) until the first chapter line (excluded). It isn't listed in the contents chapter. As in this Manual, it can be the cover of the document.

4.2. NUMBERED CHAPTERS

Start of a "numbered chapter" is recognized by a numbered chapter line. A chapter line is a "numbered chapter line" if contains one or more blank-separated words, of which:

- the first one is a "chapter label", made by one or more "int-dot couples", each made by:
 - one or more decimal digits, between '0' and '9'
 - a "decimal point" '.'
- the following words, if any, are the chapter title

The "level" of a numbered chapter line is the count of int-dot couples in its label, examples:

- 12345. a level-1 numbered chapter line
- 1.345. a level-2 numbered chapter line
- 0.0.0. a level-3 numbered chapter line

Numbered chapter lines must follow a couple of sequence rules:

- first numbered chapter in the file and any numbered chapter next to an automatic chapter must be a level-1 chapter
- any numbered chapter next to another one can have any level between 1 and the level of the previous one plus 1, but no more

Numbered chapter titles are:

- shrunk and uppercased where they are
- shrunk and titlecased when inserted:
 - into contents chapter
 - into page headers by '%c' (see 5.)

Numbers in input don't matter, yawp replaces them by the right ones, only the level matters. So you are free to create destroy or swap chapters as you like, they will be renumbered accordingly. Example:

```
$ cat chapters.txt
0. aAa aAa aAa

32.33. bBb bBb 'bBb'

0.0. cCc "cCc" cCc

0. 'dDd dDd' dDd

0.0. eEe 'eEe' "eEe"

9.9.9. 'fFf fFf fFf'

0.0. "gGg gGg gGg"

$ yawp -v -w 32 chapters.txt
Correct: '~/yawp/yawp.cfg' not found, default corrections
Correct: -L 43.325pt = 0.602in = 15.284mm = 1.528cm
Correct: -R 40.57pt = 0.563in = 14.312mm = 1.431cm
Correct: -T 53.814pt = 0.747in = 18.985mm = 1.898cm
Correct: -B 15.991pt = 0.222in = 5.641mm = 0.564cm
Read: yawp <-- '~/chapters.txt'
      0 header lines, max 0 chars per line, 1 page
      13 body lines, max 20 chars per line
      13 total lines, max 20 chars per line
Compute: -W 15.059pt = 0.209in = 5.312mm = 0.531cm
Correct: -W 15.906pt = 0.221in = 5.611mm = 0.561cm
Compute: char height 25.098pt = 0.349in = 8.854mm = 0.885cm
Correct: char height 26.599pt = 0.369in = 9.384mm = 0.938cm
Backup: '~/chapters.txt' --> '~/chapters-2022.06.25-18.47.57.txt'
Rewrite: yawp --> '~/chapters.txt'
      0 header lines, max 0 chars per line, 1 page
      13 body lines, max 20 chars per line
      13 total lines, max 20 chars per line

$ cat chapters.txt
1. AAA AAA AAA

1.1. BBB BBB 'BBB'

1.2. CCC "cCc" CCC

2. 'DDD DDD' DDD

2.1. EEE 'EEE' "eEe"

2.1.1. 'FFF FFF FFF'

2.2. "gGg gGg gGg"
```

Figure 4.2.a. Example Of Chapter Renumbering

4.3. AUTOMATIC CHAPTERS

Contents index and figures chapters are "automatic chapters", this means:

- the file can't contain more than one automatic chapter for each of the three kinds
- all input lines after the automatic chapter line until and the next chapter line (or until end of file) are supposed to be the old chapter content and are deleted
- the new chapter content is automatically inserted after the chapter line

If `-j` and `-J` (see 6.) are off, pages are not numbered. So, if an automatic chapter is requested, it will still appear, but without page numbers.

BEWARE: an error in the next chapter line could erase a piece of your file, so after yawp processing check the result and if needed go back to previous version by `-U`.

Automatic chapters are controlled by `-c -f -F` and `-i` arguments, which must all be non-null strings. Furthermore `-c -f` and `-i` must all have distinct values, and `-F` can't contain blanks.

4.3.1. CONTENTS CHAPTER

For "contents chapter" we mean an automatic chapter containing the list of chapters, possibly with the number of the page they begin on. Namely, the contents chapter will list all numbered chapters, the figures chapter, and the index chapter, but it won't list either the nameless chapter or the contents chapter itself.

Contents chapter line starting the contents chapter is defined by `-c` argument:

- `"-c, --contents-title":` title of contents chapter (default: 'contents'), example:

```
| $ yawp -c 'list of chapters' ...
```

A chapter line is a contents chapter line if it's equivalent to `-c`. Such a line is:

- shrunk and uppercased where it is
- shrunk and titlecased when inserted into page headers by `'%c'` (see 5.)

Example:

```
$ cat contents.txt
contents

    (any previous content
    of contents chapter
    will be deleted)

0. aAa aAa aAa
32.33. bBb bBb 'bBb'
0.0. cCc "cCc" cCc
0. 'dDd dDd' dDd
0.0. eEe 'eEe' "eEe"
9.9.9. 'fFf fFf fFf'
0.0. "gGg gGg gGg"
```

```

$ yawp -v -w 32 contents.txt
Correct: '~/yawp/yawp.cfg' not found, default corrections
Correct: -L 43.325pt = 0.602in = 15.284mm = 1.528cm
Correct: -R 40.57pt = 0.563in = 14.312mm = 1.431cm
Correct: -T 53.814pt = 0.747in = 18.985mm = 1.898cm
Correct: -B 15.991pt = 0.222in = 5.641mm = 0.564cm
Read: yawp <-- '~/contents.txt'
      0 header lines, max 0 chars per line, 1 page
      19 body lines, max 25 chars per line
      19 total lines, max 25 chars per line
Compute: -W 15.059pt = 0.209in = 5.312mm = 0.531cm
Correct: -W 15.906pt = 0.221in = 5.611mm = 0.561cm
Compute: char height 25.098pt = 0.349in = 8.854mm = 0.885cm
Correct: char height 26.599pt = 0.369in = 9.384mm = 0.938cm
Backup: '~/contents.txt' --> '~/contents-2022.06.25-18.47.57.txt'
Rewrite: yawp --> '~/contents.txt'
      0 header lines, max 0 chars per line, 1 page
      23 body lines, max 28 chars per line
      23 total lines, max 28 chars per line

$ cat contents.txt
CONTENTS

    • 1.      Aaa Aaa Aaa
    • 1.1.    Bbb Bbb 'Bbb'
    • 1.2.    Ccc "cCc" Ccc
    • 2.      'Ddd Ddd' Ddd
    • 2.1.    Eee 'Eee' "eEe"
    • 2.1.1.  'Fff Fff Fff'
    • 2.2.    "gGg gGg gGg"

1. AAA AAA AAA
1.1. BBB BBB 'BBB'
1.2. CCC "cCc" CCC
2. 'DDD DDD' DDD
2.1. EEE 'EEE' "eEe"
2.1.1. 'FFF FFF FFF'
2.2. "gGg gGg gGg"

```

Figure 4.3.1.a. Example Of Contents Chapter

4.3.2. FIGURES CHAPTER

For "figures chapter" we mean an automatic chapter containing a list of figure captions, possibly showing which page a "caption" is on. Not all pictures need to be followed by a caption, and a caption without a picture doesn't make sense, so we can say a "figure" is a picture followed by a caption.

Figures chapter is controlled by `-f` and `-F` arguments. Figures chapter line starting the figures chapter is defined by `-f`:

- `"-f, --figures-title":` title of figures chapter (default: 'figures'), example:

```
| $ yawp -f 'list of figures' ...
```

A chapter line is a figure chapter line if it's equivalent to `-f`. Such a line is:

- shrunk and uppercased where it is
- shrunk and titlecased when inserted:
 - into the contents chapter
 - into page headers by `'%c'` (see 5.)

Yawp recognizes a figure caption by -F:

- "-F, --caption-prefix": first word of figure captions (default: 'figure')

-F must be a single word, in other words it can't contain blanks.

A line is a "figure caption line" if:

- is the first line in file or it's preceded by an empty line
- is the last line in file or it's followed by an empty line
- is a not empty line and starts with blank
- contains two or more blank-separated words, of which:
 - the first one is equivalent to -F
 - the second one is a "caption label" containing:
 - the label of the containing chapter (label of unnamed chapter is null string '')
 - a progressive lowercase letter between 'a' and 'z', meaning caption position in the containing chapter
 - a decimal point '.'
 - the following words, if any, are the figure title

Therefore we can say a caption line must be a one-line picture.

Examples of correct caption labels are:

- 'a.' (the first caption in the unnamed chapter)
- '7.c.' (the third caption in the '7.' chapter)
- '7.9.b.' (the second caption in the '7.9.' chapter)
- '7.9.8.z.' (the 26th caption in the '7.9.8.' chapter)

A single chapter can't contain more than 26 caption lines, one for each letter of english alphabet.

The input caption label can be any, of any level, it's automatically updated on output. Caption lines are:

- shrunk titlecased and centered where they are
- shrunk and titlecased when listed in the figures chapter

The label letter always remains lowercase.

Notice the difference between the figures chapter line (one, starting the figures chapter) and the figure caption lines (many, listed in the figures chapter).

We suppose a figure caption is logically connected with previous picture, so, when applicable, figure caption lines are logically pasted with previous picture, in order to avoid figure and caption split across two pages.

As an example, see in this manual the scattered figure captions and the figures chapter at end.

4.3.3. INDEX CHAPTER

For "index chapter" we mean an automatic chapter containing an alphabetical list of subjects, possibly showing which pages each "subject" is on. Subjects are sought in text lines only, therefore neither in pictures nor in captions nor in chapter lines. A subject can appear:

- as a "quoted subject" if it's preceded and followed by a "double quote" '"' (but double quotes preceded or followed by a "single quote" "'" or by another double quote '"' are not taken into account, in order to allow things like '"' in text)
- an "unquoted subject" otherwise

Unquoted subjects are recognized everywhere if the file contains at least one corresponding quoted subject somewhere. Hence in index chapter the line relating to a given subject will contain:

- quoted (by '"') page numbers for pages containing one or more quoted

- instances of the subject (and zero or more unquoted instances)
- unquoted page numbers for pages containing one or more unquoted instance of the subject and zero quoted instances

Subject length can't be greater than half of -w argument, whether set by user or automatically computed.

Index chapter line starting the index chapter is defined by -i argument:

- "-i, --index-title": title of index chapter (default: 'index'), example:
| \$ yawp -i 'list of subjects' ...

A chapter line is an index chapter line if it's equivalent to -i. Such a line is:

- shrunk and uppercased where it is
- shrunk and titlecased when inserted:
 - into the contents chapter
 - into page headers by '%c' (see 5.)

For technical reasons, a subject cannot occupy more than one line in the index chapter. Hence if a subject is referenced in many pages and the resulting line is longer than -w argument, then it's truncated and terminated by '...' with a warning message.

As an example, see the index chapter at end of this manual.

5. PAGING

Normally page headers in input are deleted and the file in output is not split in pages by page headers. Insertion of page headers is controlled by `-j` and `-J` arguments:

- `"-j, --eject"`: insert a page header on full page

If `-f` is turned on, a page header is inserted into the file when current line doesn't fit into current page.

- `"-J, --eject-pict-chap"`: insert page headers on full page, on broken picture, and before level-1/contents/figures/index chapters

If `-J` is turned on (or both `-j` and `-J` are), a page header is inserted into the file when:

- current line doesn't fit into current page
- current picture doesn't fit into current page
- current line is a numbered level-1 or contents or figures or index chapter line

If a picture is too long to fit into the current page, a page eject is forced. But if the picture is higher than one page, such a page eject is useless and doesn't take place, see for example the configuration file in 7.2.

Each page, except the first one, is prefixed by a page header. Therefore the total number of pages is equal to the number of page headers plus 1. Each page header is made up of two header lines:

- first header line, starting with a "form feed" `'\f'`, containing data such as file name, chapter title, date, time or page number, under control of `-e -E -o -O` and `-a` arguments
- second header line, a dashed separation line of "macron" `'-'` characters, which form a kind of underline under the first one

The form feed character causes a page break but is a non-printable character, therefore it's not considered when measuring the length of lines. Depending on your editor, it can appear as a quarter musical note, or as a small empty rectangle.

When the file is initially read, if `-U` and `-N` are turned off, all header lines are eliminated. So, if you want to eliminate page headers from your file, just run yawp without `-j` or `-J`.

BEWARE: never start a text line with a form feed or a macron character, after a yawp run such a line would disappear.

Inserting page headers in a Python file doesn't make sense, so if the file name ends with `'.py'`, `-j` and `-J` are silently turned off.

The content of first header line is controlled by `-e -E -o -O` and `-a` arguments:

- `"-e, --even-left"`: headers of even pages, left (default: `'%n/%N'`)
- `"-E, --even-right"`: headers of even pages, right (default: `'%f.%e %Y-%m-%d %H:%M:%S'`)
- `"-o, --odd-left"`: headers of odd pages, left (default: `'%c'`)
- `"-O, --odd-right"`: headers of odd pages, right (default: `'%n/%N'`)

<code>-e</code>	<code>-E</code>	<code>-o</code>	<code>-O</code>
-----	-----	-----	-----
even page		odd page	

Figure 5.a. Page Header When `"-a"` Is Off

Each "percent variable" is evaluated as follows, no other percent variable is allowed.

VAR	VALUE
'%P'	file "long path", with no ending '/'
'%p'	file "short path", with no ending '/'
'%f'	file name, with no extension
'%e'	file extension, with no separator '.'
'%Y'	current year, 4 digits
'%m'	current month, 2 digits
'%d'	current day, 2 digits
'%H'	current hour, 2 digits
'%M'	current minute, 2 digits
'%S'	current second, 2 digits
'%n'	current page number
'%N'	total number of pages
'%c'	current level-1/contents/figures/index chapter
'%%'	'%'

Figure 5.b. Percent Variables

What do "long path" and "short path" mean? If for instance the file is '/home/xxxx/yyy/zzz.www.txt' and the user is 'xxxx' then we get:

- '%P' --> '/home/xxxx/yyy'
- '%p' --> '~/yyy'
- '%f' --> 'zzz.www'
- '%e' --> 'txt'
- '%P/%f.%e' --> '/home/xxxx/yyy/zzz.www.txt'
- '%p/%f.%e' --> '~/yyy/zzz.www.txt'

If you don't need front-back printing, set:

- "-a, --all-pages-E-e": put in all page headers -E at left and -e at right

and so you will get the same header format on even and odd pages.

-E-----e	-E-----e
even page	odd page

Figure 5.c. Page Header When "-a" Is On

Notice the swap between -e and -E.

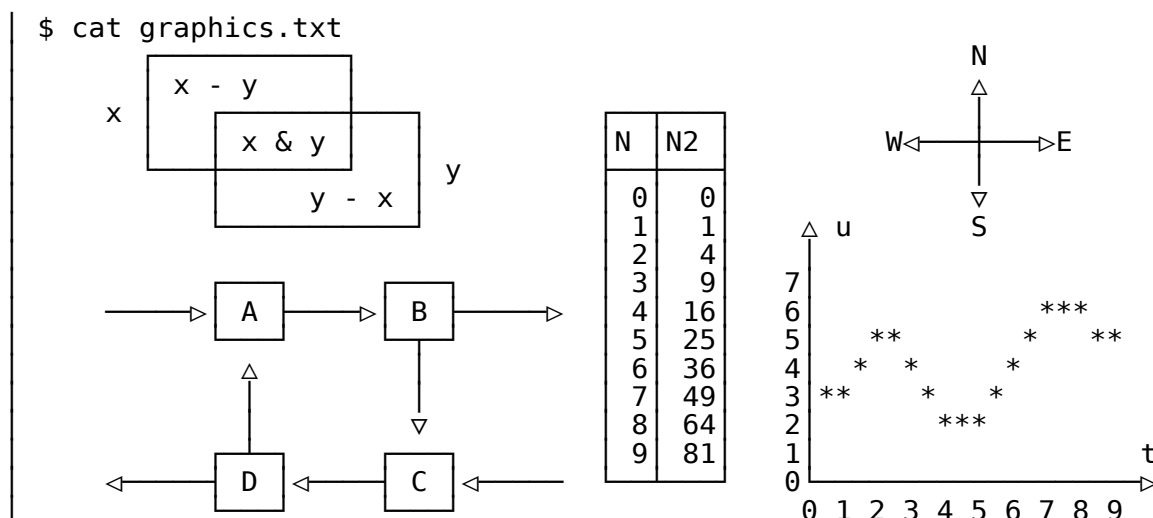


Figure 6.a. Example With "-g"

Argument -g works with -N too, so you can redraw picture files without the need for a blank at beginning of the lines, and without yawp trying to format the file. We can say that when -N is turned on, the whole file is a single picture. As another example, we want to print a chessboard.

```
$ cat chessboard.txt
```

```

$ yawp -v -N -g -A 1 -X -L 1cm chessboard.txt
Correct: '~/.yawp/yawp.cfg' not found, default corrections
WARNING: -L 1cm < 2cm, you may get unexpected results
Correct: -L 13.737pt = 0.191in = 4.846mm = 0.485cm
WARNING: -R 1cm < 2cm, you may get unexpected results
Correct: -R 10.456pt = 0.145in = 3.689mm = 0.369cm
Correct: -T 53.814pt = 0.747in = 18.985mm = 1.898cm
Correct: -B 15.991pt = 0.222in = 5.641mm = 0.564cm
Read: yawp <-- '~/.chessboard.txt'
    0 header lines, max 0 chars per line, 1 page
    17 body lines, max 17 chars per line
    17 total lines, max 17 chars per line
Compute: -w 17
Compute: -W 31.681pt = 0.44in = 11.176mm = 1.118cm
Correct: -W 33.464pt = 0.465in = 11.805mm = 1.181cm
Compute: char height 31.681pt = 0.44in = 11.176mm = 1.118cm
Correct: char height 33.576pt = 0.466in = 11.845mm = 1.184cm
Backup: '~/.chessboard.txt' --> '~/.chessboard-2022.06.25-18.47.57.txt'
Rewrite: yawp --> '~/.chessboard.txt'
    0 header lines, max 0 chars per line, 1 page
    17 body lines, max 17 chars per line
    17 total lines, max 17 chars per line
Export: '~/.chessboard.txt' --> '~/.chessboard.pdf'

```

```
$ cat chessboard.txt
```

	X		X		X		X
X		X		X		X	
	X		X		X		X
X		X		X		X	
	X		X		X		X
X		X		X		X	
	X		X		X		X
X		X		X		X	

Figure 6.b. Example With "-N -g -A 1"

Some remarks:

- '-N' prevents formatting the file as text, but the backup is still performed
- defaults '-w 0' and '-W 0' make the picture take up all available space between left and right margins
- '-A 1' transforms rectangles into squares in chessboard.pdf, so by printing it on an A4 sheet you get a square chessboard of about 18x18cm, 7x7in

7. PDF

7.1. PDF EXPORTING

Normally the file isn't exported to a PDF file. But if you set the argument:

- `"-X, --export-view-pdf"`: at end export and view PDF file

then a PDF file is exported and viewed by a PDF viewer (as atril or evince) for check preview and print.

To define the PDF viewer, set the argument:

- `"-Y, --view-pdf-by"`: viewer for the exported PDF file (default: `'xdg-open'`)

The default value `'xdg-open'` causes you view the PDF file thru the system default PDF viewer.

To define path and name of the exported PDF file, set the argument:

- `"-P, --file-pdf"`: exported PDF file (default: `'%P/%f.pdf'`)

Each "percent variable" in `-P` is evaluated as explained in chapter 6., but `'%n'`, `'%N'` and `'%c'` are not applicable and not allowed. `'%P'` and `'%p'` are both allowed but give the same result. The resulting file name must end with lowercase `'.pdf'`.

Character size is defined by `-W` and `-A` arguments:

- `"-W, --char-width"`: character width (default: `'0'` = automatic, unit: `pt/in/mm/cm`)

Value is an integer or float literal followed by a lowercase unit suffix:

- `'pt'` for points (1 inch = 72 points)
- `'in'` for inches
- `'mm'` for millimeters
- `'cm'` for centimeters

so for instance these are all equivalent, giving a value of one inch:

- `-W 72pt`
- `-W 1.0in`
- `-W 25.4mm`
- `-W 2.54cm`

There is no default unit, the suffix is mandatory. Only a zero value (as `'0'` or `'0.0'`) can lack the suffix, but a zero value for `-W` means "automatic".

Both `-W` and `-w` (line width in chars per line, see 3.1.) can be zero hence automatic, namely:

- if `-W` is automatic and `-w` is not, `-W` is computed from `-w`
- if `-w` is automatic and `-W` is not, `-w` is computed from `-W`
- if both `-w` and `-W` are automatic,
 - `-w` is deduced from max body line length in file (page headers are not considered)
 - `-W` is computed from `-w`

Given the character width by `-W`, the character height is derived from `-A`:

- `"-A, --char-aspect"`: character aspect ratio = char width / char height (default: `'3/5'`, `'1'` = square grid)

`-A` is a ratio, so it can be:

- an integer or float literal (as `'1'` or `'0.6'`)
- two integer or float literals, separated by a "slash" `'/'` (as `'3/5'`)

Dimensions of the paper sheet, expressed in points inches centimeters or millimeters (as explained for -W before) are defined by -S:

- "-S, --paper-size": portrait paper size width x height (default: 'A4' = '210x297mm', unit: pt/in/mm/cm)

Format is portrait, in other words the width cannot be greater than the height.

Values can be symbolic names too, see the following table. These names are case-insensitive ('A4' or 'a4' is the same), while the 'x' and the unit suffix in the explicit value must be lowercase.

NAME	VALUE
HALF LETTER	5.5x8.5in
LETTER	8.5x11.0in
LEGAL	8.5x14.0in
JUNIOR LEGAL	5.0x8.0in
LEDGER	11.0x17.0in
TABLOID	11.0x17.0in
A0	841x1189mm
A1	594x841mm
A2	420x594mm
A3	297x420mm
A4	210x297mm
A5	148x210mm
A6	105x148mm
A7	74x105mm
A8	52x74mm
A9	37x52mm
A10	26x37mm
B0	1000x1414mm
B1	707x1000mm
B1+	720x1020mm
B2	500x707mm
B2+	520x720mm
B3	353x500mm
B4	250x353mm
B5	176x250mm
B6	125x176mm
B7	88x125mm
B8	62x88mm
B9	44x62mm
B10	31x44mm

Figure 7.1.a. Symbolic Names For "-S"

Paper width and height are exchanged with each other by -Z:

- "-Z, --landscape": turn page by 90 degrees (default: portrait)

Unprintable margins around the paper sheet are controlled by:

- "-L, --left-margin": left margin (default: '2cm', unit: pt/in/mm/cm)
- "-R, --right-margin": right margin (default: '-L', unit: pt/in/mm/cm)
- "-T, --top-margin": top margin (default: '2cm', unit: pt/in/mm/cm)
- "-B, --bottom-margin": bottom margin (default: '-T', unit: pt/in/mm/cm)

Margins are expressed in points inches centimeters or millimeters as explained above for -W, but:

- if -R has the default value '-L' then it's forced to the value of -L
- if -B has the default value '-T' then it's forced to the value of -T

BEWARE: margins of less than 2cm are allowed but not guaranteed, and may give unexpected results.

7.2. PDF CORRECTIONS

Export of the PDF file is performed by CUPS via the "lp" Unix command. Geometry is controlled by yawp passing to lp various arguments:

- -o cpi=N (number of characters per inch, default=10)
- -o lpi=N (number of lines per inch, default=6)
- -o page-left=N (left page margin, value in points)
- -o page-right=N (right page margin, value in points)
- -o page-top=N (top page margin, value in points)
- -o page-bottom=N (bottom page margin, value in points)

These options are undocumented in the lp man page, but you can find them for instance in:

<https://www.computerhope.com/unix/ulp.htm>

Unfortunately, by printing the PDF file on paper the above options are not respected, but are affected by not negligible errors. With "lpr" command the same thing happens. Namely:

- page margins are enlarged, so they need to be reduced
- character width and height are reduced, so they need to be enlarged
- in portrait and landscape orientation errors are different, and so corrections must be

A mechanism in yawp tries to correct such errors. If -v is on, such corrections are signaled by information messages starting with 'Correct:'. This device has been tuned for the default paper size 'A4' = '210x297mm', both portrait and landscape, on a Hewlett-Packard Officejet 2620 printer. With another format or another printer you may get different results, and so you may need different correctors.

Corrections are controlled by -C argument:

- "-C, --calibration": don't correct character size and page margins

and by the configuration file '~/.yawp/yawp.cfg', if exists:

- if -C argument is turned on, no correction takes place, and we can make experiments in order to define content of '~/.yawp/yawp.cfg'
- otherwise if '~/.yawp/yawp.cfg' doesn't exist, the default corrections are applied
- otherwise all corrector coefficients are read from '~/.yawp/yawp.cfg'

An example for '~/.yawp/yawp.cfg' follows, corresponding to the default corrections:

```
#----- yawp configuration file -----

plm 10mm 16mm # portrait left margin
plm 20mm 24mm
plm 30mm 35mm
plm 40mm 43mm
plm 50mm 52mm
plm 60mm 62mm
plm 70mm 72.5mm
plm 80mm 83mm
plm 90mm 92mm
plm 100mm 101mm

llm 10mm 20.5mm # landscape left margin
llm 20mm 29.5mm
llm 30mm 39mm
llm 40mm 48mm
llm 50mm 57mm
llm 60mm 66.5mm
llm 70mm 75.5mm
llm 80mm 84.5mm
llm 90mm 94mm
llm 100mm 104mm
```

prm 10mm 15mm # portrait right margin
prm 20mm 25.5mm
prm 30mm 34.5mm
prm 40mm 44.5mm
prm 50mm 54.5mm
prm 60mm 64.5mm
prm 70mm 72mm
prm 80mm 81mm
prm 90mm 91.5mm
prm 100mm 100mm

lrm 10mm 22mm # landscape right margin
lrm 20mm 30mm
lrm 30mm 40mm
lrm 40mm 49.5mm
lrm 50mm 58mm
lrm 60mm 68mm
lrm 70mm 77.5mm
lrm 80mm 86mm
lrm 90mm 96mm
lrm 100mm 104mm

ptm 10mm 11.5mm # portrait top margin
ptm 20mm 21mm
ptm 30mm 30.5mm
ptm 40mm 39.5mm
ptm 50mm 49mm
ptm 60mm 59mm
ptm 70mm 68mm
ptm 80mm 77.5mm
ptm 90mm 87mm
ptm 100mm 96mm

ltm 10mm 11mm # landscape top margin
ltm 20mm 20.5mm
ltm 30mm 30mm
ltm 40mm 39mm
ltm 50mm 48.5mm
ltm 60mm 57.5mm
ltm 70mm 67mm
ltm 80mm 76mm
ltm 90mm 85mm
ltm 100mm 95mm

pbm 10mm 24mm # portrait bottom margin
pbm 20mm 34mm
pbm 30mm 43mm
pbm 40mm 52.5mm
pbm 50mm 62mm
pbm 60mm 71mm
pbm 70mm 81mm
pbm 80mm 90mm
pbm 90mm 100mm
pbm 100mm 109.5mm

lbm 10mm 24mm # landscape bottom margin
lbm 20mm 32mm
lbm 30mm 42mm
lbm 40mm 52mm
lbm 50mm 60mm
lbm 60mm 70mm
lbm 70mm 80mm
lbm 80mm 88mm
lbm 90mm 99mm
lbm 100mm 107mm

pcw 100mm 94.674mm # portrait character width

lcw 100mm 92.200mm # landscape character width

```
pch 100mm 94.358mm # portrait character height
lch 100mm 92.647mm # landscape character height
```

Figure 7.2.a. Example Of "yawp.cfg" Configuration File

As you can see, standard Unix '#'-comments and empty lines are accepted. The file contains a set of correctors, in any order. Each "corrector" has the form 'k y x', where:

- k is a 3-letters lowercase "key" among 12 allowed values:
 - 'plm' portrait left margin
 - 'prm' portrait right margin
 - 'ptm' portrait top margin
 - 'pbm' portrait bottom margin
 - 'pcw' portrait character width
 - 'pch' portrait character height
 - 'llm' landscape left margin
 - 'lrm' landscape right margin
 - 'ltm' landscape top margin
 - 'lbm' landscape bottom margin
 - 'lcw' landscape character width
 - 'lch' landscape character height
- y is a "wanted value" in points/inches/millimeters/centimeters
- x is an "obtained value" in points/inches/millimeters/centimeters

For instance, take the first corrector, 'plm 10mm 16mm'. This means that, by trying to print a page with -Z off, -C on, and -L 10mm (y = 10mm), we have empirically obtained an actual measured left margin of 16mm on paper (x = 16mm). Therefore if we wish an actual measured left margin of 16mm on paper (x = 16mm), we should give to lp command an argument '-o page-left' equivalent to 10mm (y = 10mm). So, given a wanted measure x and a set of corresponding correctors, which is the function $y = f(x)$ telling us what argument y provide to lp?

For each key we can have zero, one, two or more correctors, namely:

- zero correctors: no correction, $y = f(x) = x$, the straight line passing by (0, 0) and (1, 1)
- one corrector, say (y0, x0): $y = f(x) = (y0 / x0) * x$, the straight line passing by (0, 0) and (x0, y0)
- two correctors, say [(y0, x0), (y1, x1)], $y = f(x) = y0 + (x - x0) * (y1 - y0) / (x1 - x0)$, the straight line passing by (x0, y0) and (x1, y1)
- three or more correctors, say [(y0, x0), (y1, x1), (y2, x2), ...], $y = f(x)$ is approximated by the least-squares straight line defined by (x0, y0), (x1, y1), (x2, y2), ...

Finally, if the result is less than zero, it's forced to zero.

8. AFTERWORDS

8.1. ACRONYMS

ACRONYM	MEANING
CUPS	Common Unix Printing System
IDE	Integrated Development Environment
IDLE	Integrated Development and Learning Environment
MS	MicroSoft
PDF	Page Description Format
PyPI	Python Package Index
RAM	Random Access Memory
WYSIWYG	What You See Is What You Get
YAWP	Yet Another Word Processor

Figure 8.1.a. Acronyms

8.2. CHARACTERS

UNICODE CHARACTER NAME	CHAR	HEX	DEC
horizontal tab	'\t'	0009	9
line feed	'\n'	000a	10
form feed	'\f'	000c	12
carriage return	'\r'	000d	13
blank	' '	0020	32
double quote	'\"'	0022	34
number sign	'#'	0023	35
percent sign	'%'	0025	37
single quote	'\"'	0027	39
decimal point	'.'	002e	46
slash	'/'	002f	47
digit zero	'0'	0030	48
digit nine	'9'	0039	57
circumflex accent	'^'	005e	94
back quote	'`'	0060	96
latin small letter a	'a'	0061	97
latin small letter x	'x'	0078	120
latin small letter z	'z'	007a	122
macron	'-'	00af	175
black small circle	'•'	2022	8226
box drawings light horizontal	'─'	2500	9472
box drawings light vertical	' '	2502	9474
box drawings light down and right	'└'	250c	9484
box drawings light down and left	'└'	2510	9488
box drawings light up and right	'└'	2514	9492
box drawings light up and left	'└'	2518	9496
box drawings light vertical and right	'└'	251c	9500
box drawings light vertical and left	'└'	2524	9508
box drawings light down and horizontal	'└'	252c	9516
box drawings light up and horizontal	'└'	2534	9524
box drawings light vertical and horizontal	'└'	253c	9532
white up-pointing triangle	'△'	25b3	9651
white right-pointing triangle	'▷'	25b7	9655
white down-pointing triangle	'▽'	25bd	9661
white left-pointing triangle	'◁'	25c1	9665

Figure 8.2.a. Characters

8.3. CREDITS

A somewhat analogous (but very different) program is the Unix command "fmt", for details type:

```
| $ man fmt
```

We are not aware of any other program of this kind.

Yawp has been developed under Xubuntu 21.10:

<http://www.xubuntu.org>

Standard PDF viewer in Xubuntu 21.10 is atril 1.26.0:

<https://wiki.mate-desktop.org/mate-desktop/applications/atril>

by Python 3.9.7 and IDLE, the Python's IDE:

<https://www.python.org>

and published on PyPI by flit 3.7.1:

<https://pypi.org/project/flit>

8.4. HISTORY

- version 0.7.1 - 2022-06-25 - Development Status :: 4 - Beta
 - '-f, --form-feed', renamed '-j, --eject'
 - '-F, --form-feed-chap', renamed '-J, --eject-pict-chap'
 - '-X, --export-pdf', added
 - '-Y, --view-pdf-by', added
 - -j allows pictures be spanned across pages (while -J doesn't)
 - '-f, --figures-title', '-F, --caption-prefix', figures chapter, added
 - characters quoted by '"' in chapter titles and figure captions are neither uppercased nor titlecased
 - index subjects are sought in text only
 - '-s, --echo-shell', suppressed
 - '-p, --print-file', suppressed
- version 0.6.1 - 2022-05-06 - Development Status :: 4 - Beta
 - '-k, --calibration', renamed '-C, --calibration'
 - max subject lenght and max length of chapter titles are -w / 2
 - corrections, made parametric by configuration file yawp.cfg
 - in index chapter:
 - page numbers for quoted subjects, quoted
 - unquoted page numbers for unquoted subjects, added
 - '-m, --max-subject', suppressed
 - '-Q, --print-quality', suppressed
- version 0.5.4 - 2022-04-04 - Development Status :: 4 - Beta
 - just another fix to the Yawp Manual
- version 0.5.3 - 2022-04-02 - Development Status :: 4 - Beta
 - just a fix to the installation instructions in Yawp Manual
- version 0.5.2 - 2022-04-02 - Development Status :: 4 - Beta
 - correction of page margins and char size, new algorithm
 - page margins, no more bounded by 2cm..8cm interval (but below 2cm you may get unexpected results)
 - '-g, --graphics', now works with -N too
 - information messages about corrected values, added
 - index subjects, taken not only from text but from pictures too
 - '-b, --dump-buffer', useful for debug only, suppressed
- version 0.5.1 - 2022-03-19 - Development Status :: 4 - Beta
 - completely redefined and rewritten
- version 0.4.2 - 2022-01-04 - obsolete and deprecated
- version 0.4.1 - 2022-01-03 - obsolete and deprecated
 - first version on pypi.org

8.5. LICENSE

This program is free software, you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3, 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 General Public License for more details.

You should have received a copy of the GNU General Public License along with this program (see the file LICENSE), if not, see:

<https://www.gnu.org/licenses/>

or contact Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02111-1301 USA.

Please send bugreports, fixes, enhancements, criticism and (possibly) praise and thanks to carlo.alessandro.verre@gmail.com.

9. ARGUMENTS

Usage arguments:

- -h, --help: show a help message and exit
- -H, --manual: view this yawp-generated PDF Yawp Manual and exit
- -V, --version: show program's version number and exit
- -v, --verbose: display information and warning messages on stderr
- -N, --no-format: run in no-format mode (default: run in format mode)
- -U, --undo: run in undo mode (default: run in format mode)
- -g, --graphics: redraw '-' segments and '^'-arrowheads

Formatting arguments:

- -w, --chars-per-line: line width in characters per line (default: '0' = automatic)
- -l, --left-only: justify text lines at left only (default: at left and right)
- -c, --contents-title: title of contents chapter (default: 'contents')
- -f, --figures-title: title of figures chapter (default: 'figures')
- -F, --caption-prefix: first word of figure captions (default: 'figure')
- -i, --index-title: title of index chapter (default: 'index')

Paging arguments:

- -j, --eject: insert page headers on full page
- -J, --eject-pict-chap: insert page headers on full page, on broken picture, and before level-1/contents/figures/index chapters
- -e, --even-left: headers of even pages, left (default: '%n/%N')
- -E, --even-right: headers of even pages, right (default: '%f.%e %Y-%m-%d %H:%M:%S')
- -o, --odd-left: headers of odd pages, left (default: '%c')
- -O, --odd-right: headers of odd pages, right (default: '%n/%N')
- -a, --all-pages-E-e: put in all page headers -E at left and -e at right

PDF arguments:

- -X, --export-view-pdf: at end export and view PDF file
- -Y, --view-pdf-by: viewer for the exported PDF file (default: 'xdg-open')
- -P, --file-pdf: exported PDF file (default: '%P/%f.pdf')
- -W, --char-width: character width (default: '0' = automatic, unit: pt/in/mm/cm)
- -A, --char-aspect: character aspect ratio = char width / char height (default: '3/5', '1' = square grid)
- -S, --paper-size: portrait paper size width x height (default: 'A4' = '210x297mm', unit: pt/in/mm/cm)
- -Z, --landscape: turn page by 90 degrees (default: portrait)
- -L, --left-margin: left margin (default: '2cm', unit: pt/in/mm/cm)
- -R, --right-margin: right margin (default: '-L', unit: pt/in/mm/cm)
- -T, --top-margin: top margin (default: '2cm', unit: pt/in/mm/cm)
- -B, --bottom-margin: bottom margin (default: '-T', unit: pt/in/mm/cm)
- -C, --calibration: don't correct character size and page margins

Positional argument:

- file: text file to be processed

FIGURES

• 1.1.a.	Usage Flow Diagram	3
• 3.1.a.	Formatting State Diagram	7
• 3.1.b.	Example With "-N -w 1"	8
• 3.3.a.	Example With A Python File	11
• 4.a.	Chapter Kinds	12
• 4.2.a.	Example Of Chapter Renumbering	13
• 4.3.1.a.	Example Of Contents Chapter	15
• 5.a.	Page Header When "-a" Is Off	18
• 5.b.	Percent Variables	19
• 5.c.	Page Header When "-a" Is On	19
• 6.a.	Example With "-g"	21
• 6.b.	Example With "-N -g -A 1"	22
• 7.1.a.	Symbolic Names For "-S"	24
• 7.2.a.	Example Of "yawp.cfg" Configuration File	27
• 8.1.a.	Acronyms	28
• 8.2.a.	Characters	28

INDEX

- -A, --char-aspect "23", 31
- -B, --bottom-margin "24", 31
- -C, --calibration "25", 29, 31
- -E, --even-right "18", 31
- -F, --caption-prefix "16", 29, 31
- -H, --manual "6", 31
- -J, --eject-pict-chap "18", 29, 31
- -L, --left-margin "24", 31
- -N, --no-format "6", 31
- -O, --odd-right "18", 31
- -P, --file-pdf "23", 31
- -R, --right-margin "24", 31
- -S, --paper-size "24", 31
- -T, --top-margin "24", 31
- -U, --undo "6", 31
- -V, --version "6", 31
- -W, --char-width "23", 31
- -X, --export-view-pdf "23", 31
- -Y, --view-pdf-by "23", 31
- -Z, --landscape "24", 31
- -a, --all-pages-E-e "19", 31
- -c, --contents-title "14", 31
- -e, --even-left "18", 31
- -f, --figures-title "15", 29, 31
- -g, --graphics "20", 29, 31
- -h, --help "6", 31
- -i, --index-title "17", 31
- -j, --eject "18", 29, 31
- -l, --left-only "8", 31
- -o, --odd-left "18", 31
- -v, --verbose "6", 31
- -w, --chars-per-line "8", 31
- Yawp Manual "4", 6, 29, 31
- arrowheads 3, "20"
- automatic 3, "8", 12, 13, 14, 15, 16, "23", 31
- automatic chapter "12", 13, 14, 15, 16
- automatic chapters "14"
- back quote "20"
- best practices "8"
- black small circle "7"
- body line "7", 8, 23
- caption "15", 16
- caption label "16"
- carriage return "6"
- chapter label "12", 16
- chapter lines "12", 13, 16
- chapters 3, 4, "12", 13, 14, 18, 31
- circumflex accent "20"
- contents chapter 3, "12", 13, "14", 15, 17, 31
- corrector 25, "27"
- decimal point "7", "12", 16
- dot character "7"
- dot line "7", 9
- double quote 5, "16"
- draw characters "20"
- empty line "7", 8, 9, 12, 16
- equivalent "5", 14, 15, 16, 17, 27
- error message "4", 9
- figure 3, 12, "15", 16, 29, 31
- figure caption line "16"
- figures chapter 3, "12", "15", 16, 31
- fmt "28"
- form feed "18"
- format mode "6", 7, 20, 31
- header line "7", 18
- horizontal tab "6"
- indented line "7"
- indented paragraph "7"

• index chapter	3, "12", "16", 17, 18, 29, 31
• information message	"4"
• int-dot couples	"12"
• key	"27"
• level	"12", 13
• line feed	"6"
• long path	"19"
• lp	"25", 27
• lpr	"25"
• macron	"18"
• no-format mode	"6", 20, 31
• numbered chapter	"12", 13
• numbered chapter line	"12"
• obtained value	"27"
• percent variable	"19", "23"
• picture state	"7"
• quoted subject	"16"
• segments	3, "20"
• short path	"19"
• shrink	"4"
• single quote	"16"
• slash	"23"
• stable	"3"
• subject	"16", 17, 29
• switch line	"9"
• text state	"7"
• the file	"3", 4, 6, 8, 9, 11, 12, 13, 14, 16, 18, 19, 20, ...
• titlecase	"5"
• undo mode	"6", 20, 31
• unindented line	"7", 8, 9
• unindented paragraph	"7"
• universal newlines	"6"
• unnamed chapter	"12", 16
• unquoted subject	"16"
• uppercase	"5"
• wanted value	"27"
• warning message	"4", 17
• yawp	"3", 4, 6, 7, 9, 13, 14, 18, 20, 21, 25, 29