

PLATYPUS

PAGE LAYOUT AND TYPESETTING SYSTEM

USER GUIDE

V. 0.2.2 RC1

TABLE OF CONTENTS

What Is Platypus?	4
Why Use Platypus?	5
New or modified in this Release	6
Installation	7
Running Platypus	8
PLATYRUN Convenience File	9
Platypus Command Syntax	10
Comments	10
Units of Measure	11
Page Commands	12
Page Size	12
Page Margins	15
Mirrored Margins	15
New Page	16
Page Footer	16
Columns	16
Paragraph Commands	18
New Paragraph	18
Indentation	18
Alignment	19
End of Line	19
Fonts	21
Typefaces	21
Saving and Restoring the current format	26
Bullet Lists	27

Code and Listings.....	28
User-Defined Strings (Macros)	30
System-Defined Strings	31
Debugging Platypus Results.....	32
Appendix A: Platypus Commands (Alphabetical Order).....	33
Appendix B: Licenses	38

WHAT IS PLATYPUS?

PLATYPUS IS A PAGE LAYOUT and typesetting system built with two goals: ease of use and a comprehensive feature set. At present, it is in early stages of development. Only the features discussed in this guide are currently available. This guide is frequently updated, as features are added. To know which release this document covers, please see the title page.

Page layout and typesetting system.

Presently, Platypus offers many standard layout and typesetting features, including bold, italic, underline, and strike-through text; adjustable page sizes and margins; multiple, customizable columns; use of all accessible fonts at all font sizes; mathematical symbols and special characters; macros; links and URLs; special support for code and listings; and so on—all designed for ease of use.

Many features

Ease of use

Platypus uses a plug-in architecture to generate documents in PDF. HTML will shortly be added on an experimental basis, and RTF—a format for Microsoft Word and other word processors—will come soon. Eventually, other document formats, such as eBooks will be added. All formats can be generated from the same input file.

Several output formats

Platypus is on an aggressive development cycle. Features in upcoming releases are a topic of active discussion and your input is certainly welcome. You can reach the development team at *platypus* (at) *pz.org*.

More features and formats coming

In addition, you can post support questions to the developers by using the mailing lists (<http://platypus.pz.org/maillinglists.html>). To keep abreast of Platypus' progress, please send an e-mail to our announcement list manager. We guarantee to send no more than a few announcements of new releases per *year*. In whatever form, hearing from keeps us motivated to deliver more software goodness.

Let us know what you like.

In the meantime, enjoy creating typeset-quality documents with ease.

—*The Platypus Team*
platypus@pz.org

WHY USE PLATYPUS?

Platypus accepts an input text file containing text and embedded commands; it then generates high-quality PDF files, and soon HTML pages. This approach is similar to TeX and troff, although Platypus focuses on greater ease of use than those packages.

Platypus is not a word processor. It does not use a graphical interface. Its only input is a text file (technically, a UTF-8 file). This approach makes it much easier to write documents—especially long documents—because it enables you to focus on content without having to format the document as you go. Hence, your fingers never leave the keyboard, and your attention is never interrupted by problems such as fixing bullet indent levels, or other tiresome chore.

Another advantage of this design is that you control the fundamental document. You cannot do this with word processors. If you have ever looked inside a document created by Microsoft Word or other word processors, you'll find it is filled with proprietary formatting information that is unintelligible. If something should corrupt the document, you have no way to fix it. In Platypus, the software processes the same file you work on. If the file should somehow be corrupted, you can go and fix it yourself with no difficulty.

Platypus is designed for ease of use. As you'll see in the following pages, most of the commands are intuitive and easy to learn. Moreover, most of them have intelligent defaults. So much so, that you can take most text files and run them through Platypus without modification and generate a useful PDF file (consisting of letter-size pages, with 1" margins and text in 12-point Times Roman font).

A final point: Platypus is open-source software. That is, it is available at no cost from platypus.pz.org. If you want to modify the software, you can download the source code from the same site.

We hope the you'll share our experience: The more you use Platypus, the happier you'll be creating typeset-quality documents.

NEW OR MODIFIED IN THIS RELEASE

The following items in Platypus have changed or been added since version 0.2.0.

Code. `[code]` and `[-code]` Turn code format on and off. This has the effect of switching to the default monospaced font and all end of lines become hard (that is, they are a CR, LF, or CR/LF depending on platform. You can use these commands in the middle of paragraph, in which case just the designated words will be in code font, or for entire blocks of code, such as listings (see next command).

Formatted listings. `[code|lines:1,5]` enables you to print listings, with numbered lines. The values after the lines: parameter determine the number of the first line in the listing (you can continue the numbering from a previous listing) and how many lines to skip between numbers, respectively.

URLs (with and without cover text). You can now embed clickable URLs in your documents. These can be the raw URL or a link hidden by cover text, as when you write “Click here” and the word “here” is a link to a web site.

Bullet Lists You can create bulleted lists with the `[list]` and `[-list]` pair of commands. You can create multi-level lists and you can specify the bullet character for each one.

Save and restore current format. `[savefmt]` and `[restorefmt]` This pair of commands enables you to save the current format (font, text features, leading, etc.) and then restore it at a later time. This is useful when one format is substantially changed. To get back to the previous format, use these commands rather than entering lots of specific commands to undo each of the features of the current format.

Macros. You can access a macro without printing it, by using the `[$` command. For example, `[$macroName]`

More symbols. Numerous new symbols and special characters have been added.

INSTALLATION

Platypus is a Java application that requires Java 6.0 SE (also called Java 1.6) or a later version. Platypus is delivered as a .zip file.

The .zip file should be unzipped into a new directory. This directory becomes the Platypus home directory. On all systems, a system or environment variable, PLATYPUS_HOME, should be set to point to this directory. If you do not know how to set a system variable, then you must use the **platypusrun** file included with the program.

The directory consists of the following files and subdirectories (directory names are preceded by a / for clarity):

/ platypus home directory

platypus.jar	the main Platypus program
platypusrun.bat	a convenience file for running Platypus on Windows and Vista
UserGuide.pdf	this document

/config

<various configuration files>

/licenses

LICENSE.TXT: information on Platypus license(s)

/plugins

pdf.jar
listing.jar

/lib

<various third-party libraries used by Platypus>
itext.jar

/fonts

<a few free fonts that ship with Platypus>

Once you have set PLATYPUS_HOME and have unzipped the file, run Platypus with the **-fontlist** option. (See next section.) This option examines your current system and makes an inventory of the TrueType and OpenType files on your system, and the name by which they can be referred to when generating PDF files. The resulting file, **fontlist.txt**, is placed in /config (see directories above). If you later add new fonts to your system, rerun this command so that Platypus will detect them.

RUNNING PLATYPUS

To run Platypus as a Java command:

```
java -Xmx256M -jar PLATYPUS_HOME\platypus.jar in-file out-file [options]
```

where:

in-file is any file containing text and Platypus commands. The convention is to use the .typ extension for Platypus file. But any text file with any extension can be an input file.

output-file is the resulting file. The file extension tells Platypus what format to use. If it is omitted, Platypus generates a PDF file by default.

Currently PDF and Platypus listings with color-coded syntax are supported. HTML will be supported soon, and RTF will be supported later. Instructions on generating a listing are explained below.

[options] are command-line switches that enable various options. These include:

-help prints basic usage information including version number and all valid Platypus options.

-fontlist writes an alphabetized list of all fonts found by Platypus on the system. The font names appear in the same format in which they should be specified in a Platypus document (via the `[ff:` command). This list is called **fontlist.txt** and is placed in the /config directory of the Platypus home directory.

-format identifies a format that might be different from the file extension. For example, to generate a listing of a Platypus file with commands and comments highlighted, create an HTML file and use the **-format listing** option to tell Platypus you want a listing: **platyrun infile.plat in-file.html -format listing**. Note Platypus listings can only be generated in HTML.

-verbose Platypus normally prints warnings and errors to the screen. This option increases output by also showing informational messages about internal operations.

-vverbose (for “very verbose”) prints a considerable amount of information about internal operations to the screen. This switch is used primarily for debugging purposes.

Note: to see the current version of Platypus or for a listing of command-line options, simply run Platypus without any arguments or switches or with the **-help** switch.

PLATYRUN CONVENIENCE FILE

Users of Windows and Vista will find the **platyrn.bat** file, which is a convenience file that makes it simple to run Platypus. (If you don't know how to set the environment variable, `PLATYPUS_HOME`, **platyrn.bat** is your only option.) To use the file, please open it in an editor (such as Notepad) and follow the instructions. You will need to specify the directory that is the Platypus home. Save the file. Then anywhere this document states run platypus, enter platyrn (followed by the input-file name, output-file name, and options described in the previous section).

PLATYPUS COMMAND SYNTAX

All Platypus commands are placed inside a pair of square brackets, without exception.

Here is an example of text containing Platypus commands (highlighted):

`[align:center]`Section II. Federal Highways Since 1960

`[noindent][align:left]`The Federal Highway system, which was born in the 1950's, saw remarkable growth during the 1960's. This section details the key events from that period.`[]`Before we begin, however,....

As can be seen, the commands are mostly self-explanatory and should be easy to remember, or even guess. Some commands take parameters. These commands place the parameter after a full colon, such as `[align:left]`

Parameters cannot contain spaces or embedded `[` and `]` characters, so if a parameter needs to include any of these characters, it is placed in curly brackets. For example, this command which specifies a new font face: `[ff:{arial condensed}]`

To use the `[` character in regular text without Platypus interpreting it as a command, precede it with a slash (`/`), which serves as an escape character: `/ [this is not a command]`.

To print the two characters `/ [` use: `[/]/[`. The `[/]` prints a slash. The slash character needs to be written this way only when it precedes a `[` character. At all other times, you can use a `/` and the slash character will print as expected.

Not all `[` characters introduce commands, some are simply text. For example, a `[` followed by either white-space or by a digit is always text. This means that to output `[26]` to a document requires no special escape character. This feature is designed to enable Platypus to process the greatest range of pure text files without unexpected side effects.

COMMENTS

Platypus uses both line comments and block comments.

Line comments use the `%%` sequence, which must be the first two characters in the line.

Block comments, which can appear on a single line or span many lines, start with `[%%` and end with `%%]`.

Platypus handles embedded comments correctly. To comment out a large chunk of text that might already contain block comments, simply increase the number of `%` characters for both the opening and closing markers. So, for example, `[%%% and %%%]` will comment out a block of text that contains comment blocks that use `[%%`. Likewise, adding more `%` signs to the beginning and ending blocks will comment out text containing block comments with fewer `%` signs.

UNITS OF MEASURE

Many Platypus commands sometimes use values that require a unit of measure. When these values refer to distance or to size, they can use one of four types of units: centimeters, inches, lines, and points. These units are expressed in commands as follows:

centimeters:	cm
inches:	in
lines:	li
points:	pt

A point is a unit of measure used in typesetting and defined as 1/72 of an inch (although purists will note that in its original formulation, a point was slightly smaller than this). In many cases, the unit of measure is optional. For most commands, the default unit is points, except for a handful of commands as documented in the next sections.

All units can contain fractional values that are indicated with a decimal point. Here are three examples that all communicate the same distance:

```
[paraindent:0.5in]  
[paraindent:1.27cm]  
[paraindent:36pt]
```

The line (li) unit of measure is not included in these examples because the size of a line depends on how it's been set at any given point in the document. See the [\[paraskip:](#) and [\[leading:](#) commands for more information on its use.

PAGE COMMANDS

Platypus commands are principally broken down into several key categories: those that operate on the whole page (margins, columns, layout, new page, etc.), on paragraphs (indentation, justification, etc.) and those that are applied to individual words (fonts, underlining, strikethrough, special characters). In addition, a collection of miscellaneous formatting commands round out Platypus' capabilities.

PAGE SIZE

Platypus defaults to a page size of 8½" wide by 11" high, the so-called 'letter' size in the United States. Fifty predefined page sizes are available for use with the `[pagesize:` command (Table 1). In addition, custom page sizes can be specified.

Table 1. The predefined page sizes in Platypus.

Platypus Value	Description	Metric (mm)		Inches		Points	
		W	H	W	H	W	H
4A0	DIN476	1682	2378	66.2	93.6	4768	6741
2A0	DIN476	1189	1682	46.8	66.2	3370	4768
A0	ISO	841	1189	33.1	46.8	2384	3370
A1	ISO	594	841	23.4	33.1	1684	2384
A2	ISO	420	594	16.5	23.4	1191	1684
A3	ISO	297	420	11.7	16.5	842	1191
A4	ISO	210	297	8.3	11.7	595	842
A5	ISO	148	210	5.8	8.3	420	595
A6	ISO	105	148	4.1	5.8	297	420
A7	ISO	74	105	2.9	4.1	210	297
A8	ISO	52	74	2.0	2.9	148	210

Platypus Value	Description	Metric (mm)		Inches		Points	
		W	H	W	H	W	H
A9	ISO	37	52	1.5	2.0	105	148
A10	ISO	26	37	1.0	1.5	73	105
B0	ISO	1000	1414	39.4	55.7	2834	4008
B1	ISO	707	1000	27.8	39.4	2004	2834
B2	ISO	500	707	19.7	27.8	1417	2004
B3	ISO	353	500	13.9	19.7	1000	1417
B4	ISO	250	353	9.8	13.9	708	1000
B5	ISO	176	250	6.9	9.8	498	708
B6	ISO	125	176	4.9	6.9	354	498
B7	ISO	88	125	3.5	4.9	249	354
B8	ISO	62	88	2.4	3.5	175	249
B9	ISO	44	62	1.7	2.4	124	175
B10	ISO	31	44	1.2	1.7	87	124
ANSI-A	ANSI	216	279	8.5	11	612	792
ANSI-B	ANSI	279	432	11	17	792	1227
ANSI-C	ANSI	432	539	17	22	1224	1584
ANSI-D	ANSI	539	864	22	34	1584	2448
ANSI-E	ANSI	864	1118	34	44	2448	3168
ARCH-A	Architectural A	229	305	9	12	648	864

Platypus Value	Description	Metric (mm)		Inches		Points	
		W	H	W	H	W	H
ARCH-B	Architectural B	305	457	12	18	864	1296
ARCH-C	Architectural C	457	610	18	24	1296	1728
ARCH-D	Architectural D	610	914	24	36	1728	2592
ARCH-E	Architectural E	914	1219	36	48	2592	3456
LEDGER		432	279	17	11	1224	792
LEGAL		216	356	8.5	14	612	1008
LETTER		216	279	8.5	11	612	792
NOTE		191	254	7.5	10	540	720
11X17		279	432	11	17	792	1224
Cards and Card Stock							
BUSCARD-INTL	Bus. card (Intl)	85.6	54	3.4	2.125	243	153
BUSCARD-US	Business card (US)	89	51	3.5	2	252	144
ID-1	ISO 7810	85.6	54	3.4	2.125	243	153
ID-2	ISO 7810	105	74	4.1	2.9	297	210
ID-3	ISO 7810	125	88	4.9	3.5	354	249
INDEX3X5	Index card	127	76	5	3	360	216
INDEX4X6	Index card	152	102	6	4	432	288
INDEX5X8	Index card	203	127	8	5	576	360

To specify one of these sizes, use the `pagesize` command followed by the value in the left column. So A4 size, would be specified with: `[pagesize:A4]` and legal paper in the US would be specified with `[pagesize:LEGAL]`.

In the rare event that none of the sizes in Table 1 meets your needs, you can specify your own page size with the `[pagewidth:]` and `[pageheight:]` commands. The following commands define a page of legal-size paper:

```
[pagewidth:8.5in][pageheight:14in]
```

Platypus uses the page size that is in effect when the page begins. Consequently, page size should be specified in the Platypus file *before* any text in the document, otherwise the first page will have the default size (8 ½"x11", so-called "letter" size in North America).

You can change the page size of the next page at any point in the document. Note however, that issuing a new page-size command, results in an immediate start of a new page in the specific size.

PAGE MARGINS

Margins are the (generally) blank areas that form the edges of a printed document. They can be set or modified using the following commands.

```
[tmargin: for the top margin,  
[bmargin: bottom margin,  
[lmargin: left margin, and  
[rmargin: right margin.
```

For example, the commands:

```
[tmargin:36pt] [bmargin:1.27cm]
```

create top and bottom margins of ½" each.

If no margins are specified, Platypus documents default to margins of 1" (2.54 cm) on all four sides.

PDF In PDF files, when margins are specified, they take effect at the *next* page. Consequently, if you want to set the margins for an entire document, you need to place the margin commands before any text is encountered.

To see what size the margins are currently set to, use the `[dump:page]` command. When Platypus encounters this command, it prints various data items, including margin sizes, to the console or monitor.

MIRRORED MARGINS

When printing a document that will have a binding, it is frequently desirable to have left and right margins of unequal width. Generally on a right page, the left margin will be somewhat

wider to allow for the document binding. The left page then mirrors this margin set-up: the right margin, which is closer to the binding, is wider. To get this in Platypus, define the margins for the right page and then use the command `[marginsmirrored]`, which turns on the margin mirroring. (Note: once turned on, it cannot be turned off within the same document. The default is off, meaning all pages have the same margins.)

This command also changes the way page headers and footers are printed, as it forces those mechanisms to take into account whether a page is right or left to determine where to place text and page numbers.

Page margins and margin mirroring must be specified *before* the first text. If placed after text has been output, the commands are ignored.

NEW PAGE

A new page can be forced by the `[pg]` command. Any text following this command will start at the top of the next page. To insert a blank page, use `[pg]` twice.

There is no need to put a new-page command at the end of the document. PDF files automatically make the last page a full page.

PAGE FOOTER

Platypus can print a footer at the bottom of any and all pages. By default, a page footer consisting of the page number in the following format is printed on all pages of a document except the first:

— 9 —

where 9 stands for the page number.

To disable printing of the footer, use the `[-footer]` command. To start or restart printing the footer later, use `[+footer]`.

☒ **Tip:** To print the footer on every page, including page 1, put the `[+footer]` command before the end of the text on page 1.

COLUMNS

Platypus documents can contain columns of text. To change the number of columns use the `[columns: command. So to change from the default single column to 2 columns, enter: [columns:2]. Platypus will automatically calculate the size of the columns and the gutter (the space between the columns).`

When going from 1 column to more columns, Platypus places the multicolumn text below the existing single-column text on the same page. Likewise, when going from multiple columns to 1 column, Platypus puts the single column text directly below the multiple columns.

However, when going from multiple columns to some other number of multiple columns, Platypus starts a new page. So, in this situation, changing columns has the effect of ejecting a new page.

COLUMN WIDTH

If you want to override the column width computed by Platypus, use the `[columnwidth:` command. This command should immediately precede the `[columns:` command.

To turn off custom column width, set columnwidth to 0: `[columnwidth:0]`

PARAGRAPH COMMANDS

NEW PARAGRAPH

A new paragraph is created by a blank line in the text.

This is the end of the first paragraph.

This is the beginning of the second.

A pair of commands determine how a new paragraph is started, reflecting two principal styles: block style and traditional.

Block style is frequently used in letters and memoranda. In this style (used in this guide), a line is skipped before a new paragraph, and the first line of the new paragraph is not indented:

gave their lives that that nation might live. It is altogether fitting and proper that we should do this today on this battlefield.

But, in a larger sense, we cannot dedicate—we cannot consecrate—we cannot hallow— this ground. The brave men, living and dead, who struggled here [...]

Traditional style, by contrast, does not skip a line and it does indent the first line of a new paragraph. It is the preferred format for books, magazines, and other documents consisting of more than a few pages. Here's how it looks:

gave their lives that that nation might live. It is altogether fitting and proper that we should do this today on this battlefield.

But, in a larger sense, we cannot dedicate—we cannot consecrate—we cannot hallow— this ground. The brave men, living and dead, who struggled here [...]

[\[paraskip:1li\]](#) This command tells Platypus to skip one line before the next paragraph. For block style, it should be set to 1, for traditional style it should be set to 0. Note: if no units are specified, the default of lines is used.

Typically, this command and [\[indent:](#) (which controls the first-line indent; see next section) are set once at the beginning of a document and not changed afterwards. However, you can change them at any point. If multiple instances of these commands appear in the same paragraph, Platypus uses the last one.

INDENTATION

There are four paragraph indentation commands: indent the first line only, suppress first-line indent for the following paragraph, and indent the entire paragraph from either the right or left side.

`[indent:36pt]` This command tells Platypus to indent the first line of all following paragraphs by 36 points. Of course, you can specify the distance in inches, centimeters, or points. If no unit is specified, the default is points. To turn first-line indentation off, this value should be set to zero: `[indent:0]` Note that for PDF files, Platypus defaults to an indent of zero; so there is no need to specify this value unless you want something other than zero.

Sometimes you have set up a first-line indent, but you don't want the first line of a specific paragraph indented. (This frequently occurs at the first paragraph after a section heading or in a new chapter.) In such a case, use the `[noindent]` command, which suppresses indentation for the next paragraph *only*. Note: this command must appear before new text in the paragraph.

To indent the entire paragraph from the left margin, use the `[paraindent:]` command. To indent a paragraph from the right margin, use the `[paraindentR:]` command. Note this second command has an upper-case R at the end of the command name (referring to the right margin).

To see what indentation parameters have been set for a given paragraph, use the `[dump:paragraph]` command, which displays various data items about the current paragraph, including indentation, on the system console.

ALIGNMENT

To specify how text should be aligned on the page, use one of the following options:

`[align:left]` text is aligned on the left margin (as in this document)

`[align:center]` text is aligned in the center (often used for headings)

`[align:right]` text is aligned on the right margin

`[align:just]` text is aligned on both the left and right margins

`[align:justified]`

Alignment applies to an entire paragraph. If several alignment commands appear in the same paragraph, Platypus implements the last one.

To see the current alignment setting, use the `[dump:paragraph]` command. It displays several data items on the console regarding the current paragraph, including alignment.

END OF LINE

By default, Platypus wraps lines of input text into paragraphs. Sometimes, however, you need to have a hard end of line to avoid this wrapping. The `[]` command provides this. Here is an example:

```
Shopping List
Lettuce[]
Carrots[]
Apples[]
Cherries[]
Lots of Cookies
```

This produces the expected result of a list, with one entry per line and blank lines (due to end of paragraph) after “Shopping List” and at the end, after “Lots of Cookies”.

To insert a blank line without triggering a new paragraph, enter `[]` on a line by itself. This command can be particularly useful when you want to skip multiple lines, as in:

```
[]
[]
[]
```

This inserts three blank lines in the current paragraph.

☒ **Tip:** If you insert a blank line using the `[]` command, it’s important to remember that the text prior to and immediately after the blank line are part of the *same* paragraph. This means that if after the blank line, you change alignment, leading, or any other feature that affects the current paragraph, the text *before and after* the blank line will be affected. As a result, it’s generally better to use separate paragraphs than to use on `[]` to insert a blank line between blocks of text.

☒ **Tip:** As a rule, the `[]` command should not be placed at the end of a paragraph as this will add an extra blank line to the paragraph break. However, this blank line can occasionally be desirable; this occurs occasionally when `[paraskip:01i]` is specified but a blank line is desired at a paragraph break.)

FONTS

In common parlance, the term font refers to a typeface. We're fond of saying "the Arial font." But in typesetting, a font refers to the specific instance of a typeface, including data about the size of the letters and formatting attributes, such as bold or italic. A font, as referred to by a typesetter, might be: 10-point Arial Extra Bold. Platypus uses this expanded typesetter's definition when referring to fonts. By default, Platypus uses 12-point Times Roman medium. (Medium means that there are no special formatting characteristics: it's neither bold nor italic.)

TYPEFACES

Platypus supports all TrueType and OpenType fonts on the system (these are the prevalent fonts for Windows, Mac, and Linux) plus five typefaces built into all PDF readers: Courier, Helvetica, Times Roman, Symbol, and Dingbats.

To change the current typeface, use the `[ff:` command. (`ff` stands for "font face." All font commands begin with the letter `f`.) To change to the Courier typeface, for example, enter: `[ff:COURIER]`. The typeface will change to Courier and remain Courier, until another `[ff:` command is encountered.

THE BASE 14 FONTS

Every PDF reader or rendering device is required to support the Basic 14 fonts. These fonts include Courier, Helvetica, and Times Roman in medium, italic, bold, and italic bold, plus two image/icon fonts called Symbol and Dingbat. These fonts are specified using the following commands. (Note that all these typefaces are in capitals.)

```
[ff:COURIER]
[ff:HELVETICA]
[ff:TIMES_ROMAN]
[ff:SYMBOL]
[ff:DINGBATS]
```

These fonts, which are illustrated on the following page, are known in Adobe's language, as the "Base 14 fonts."

The use of other fonts is discussed in the next section.

The 14 Fonts Built Into All PDF Readers

Courier

The Quick Brown Fox Jumped...1234567890

Courier Italic

The Quick Brown Fox Jumped...1234567890

Courier Bold

The Quick Brown Fox Jumped...1234567890

Courier Bold Italic

The Quick Brown Fox Jumped...1234567890

Helvetica

The Quick Brown Fox Jumped...1234567890

Helvetica Italic

The Quick Brown Fox Jumped...1234567890

Helvetica Bold

The Quick Brown Fox Jumped...1234567890

Helvetica Bold Italic

The Quick Brown Fox Jumped...1234567890

Times Roman

The Quick Brown Fox Jumped...1234567890

Times Roman Italic

The Quick Brown Fox Jumped...1234567890

Times Roman Bold

The Quick Brown Fox Jumped...1234567890

Times Roman Bold Italic

The Quick Brown Fox Jumped...1234567890

Symbol

τηε θυιγκ βρωων φοξ φυμπεδ . 12345

Dingbat

▼※※ □◆※※※ ※□□■※□ ※◆○□※※ ✎ ✍ ✎ ✓✓✕

OTHER FONTS

Platypus maintains a list of TrueType font files (.ttf), TrueType font collections (.ttc), and OpenType fonts (.otf) in **fontlist.txt** in the PLATYPUS_HOME\config directory. When creating or updating this list (by running Platypus with the **-fontlist** option), it searches for fonts in the following directories.

On Microsoft Windows: c:\windows\fonts\
d:\windows\fonts\
c:\winnt\fonts\
d:\winnt\fonts\

On UNIX and Linux-based systems:

/usr/X/lib/X11/fonts/TrueType
/usr/openwin/lib/X11/fonts/TrueType
/usr/share/fonts/default/TrueType
/usr/X11R6/lib/X11/fonts/ttf

On all systems:

(java runtime directory)/lib/fonts

Any fonts found in these directories are eligible for use. Platypus embeds any characters used into the PDF file if the fonts permit it (the vast majority do permit this). This renders the document accurately even if the reader's system does not have the exact same fonts. Some fonts do not allow embedding. In this case, the PDF reader software will attempt to find the closest matching font.

Platypus uses the name of the font as it's specified in the font file. If the font fact you specified is not appearing in the output document, consult **fontlist.txt** in the PLATYPUS_HOME\config directory for the right name of the font. This is an alphabetical list (by font name) of all the fonts Platypus found on your system. Here is a sample extract:

```
Arial Unicode MS=c:/windows/fonts/ARIALUNI.TTF
Batang=d:/windows/fonts/batang.ttc
BatangChe=d:/windows/fonts/batang.ttc
Bell MT=c:/windows/fonts/BELL.TTF
```

This shows that "Arial Unicode MS", "Batang", "BatangChe", and "Bell MT" are valid font names with corresponding font files on this user's system.

If the font name contains spaces, it must be placed between curly braces: {}. For example, the Bell MT font in the previous list of fonts would be specified by: `[ff:{Bell MT}]`

Note: in the event that you specify an invalid typeface, Platypus ignores the request, issues a warning on the console, and continues using the current font. So, if you don't see the font

you're expecting, check for error messages and that you have correctly specified the name of the font.

FONT ATTRIBUTES (BOLD AND ITALIC)

Bold is turned on via `[+b]` and turned off via `[-b]`. Likewise, italics uses `[+i]` and `[-i]`. These attributes remain in effect across a font change. So, if you have italics turned on and you insert a character in another typeface, it too will be italicized. Only the corresponding switch turns off bold or italics. These switches need not be paired. For example, you can italicize an entire document by placing a `[+i]` at the beginning.

Moreover, these attributes can overlap, which is something that's difficult to do in other page-layout languages. For example,

This `[+i]` is a sample of `[+b]` overlapping `[-i]` font `[-b]` attributes.

→ This *is a sample of **overlapping font*** attributes.

To turn on both bold and italics at the same time, use both switches: `[+b][+i]` or `[+i][+b]`.

FONT SIZE

To change the size of the current font, use the `[fsize:` command. For example, to change the current font to 14 point, enter: `[fsize:14pt]`. The output text remains at this new size until another command changing font size is encountered. If this command is part of a series of font commands, it can use the multiple-command syntax: `[font|size:15pt|...]`



A Look Inside: Font Size and Leading.

When the font size is changed, the leading (or space between lines) is changed as well. The rules for this are that the leading is generally set at 1.2 times the size of the *largest* font in the paragraph (see exception below.) This means that if you change to a smaller font in mid-paragraph, the leading will remain the same.

If you make the font only slightly larger (up to 3 points larger), the leading will not change. This feature enables you to change font size to adjust for a change in typefaces without modifying the spacing of the paragraph. Anything size increase larger than 3 points results in computing the leading at 1.2 times the size of the largest font.

You can over-ride these settings by explicitly setting the leading with the `[leading:` command.

LEADING (SPACE BETWEEN LINES)

The height of a line (from the base line of one line of characters to the baseline of the line below it) is known as *leading* (pronounced “led-ding”), as a reference to the days when pages were typeset by hand and strips made of lead were placed between lines of text. A typical arrangement of text and leading is 12-point text on 14-point leading.

Platypus automatically adjusts leading to fit the font you’re using. However, if you want to override this setting, you use the `leading` command. This is useful if you want to double space text, which would use this command: `[leading:21i]`

To revert back to single-spaced text later, use `[leading:11i]`. The `leading` command also accepts all the usual units of measure, so you can use it to expand or tighten the space between lines as you wish.

Note: the `leading` command affects the entire current paragraph. So if it appears at the end of the paragraph the preceding text in the paragraph will have the same leading as the end. If two or more `leading` commands appear in the same paragraph (invariably, an error), the last one is used.

To view the leading at any point in the document, use the `[dump:paragraph]` command. See the entry for that command for further details.

UNDERLINE

Underlining is turned on using `[+u]` and turned off using `[-u]`. This command results in text being underlined with a single line.

STRIKETHROUGH

Strikethrough text has a line going through it, like this: ~~This is Strikethrough~~. It is generally used in documents to show material that has been changed or deleted. Platypus supports the use of strikethrough with the following commands:

`[+st]` begins strikethrough, and `[-st]` marks the end of the strikethrough text. For example:

The committee voted `[+st]`against`[-st]` in favor of the resolution.

→ The committee voted ~~against~~ in favor of the resolution.

When the font size changes, the position of the strikethrough line is automatically changed by Platypus, so that it always falls correctly through the text.

Saving and Restoring the current format

Sometimes you want to make a few changes to the text format (perhaps changing fonts for a word or two) and you'd like to save the current format and restore it, without having to re-specify all the features of the current format. You can do this with `[savefmt]`, which saves the current format and `[restorefmt]`, which restores the format to the last saved format.

For example:

Sometimes, I prefer a `[savefmt][ff:script][fsize:11pt][+i]`cursive`[restorefmt]` font for my personal letters.

→ Sometimes I prefer a *cursive* font for my personal letters.

The purpose of this pair of commands is simply a convenience that saves a lot of keystrokes.

Note: if the `[restorefmt]` command is encountered without a preceding `[savefmt]` command, the default Platypus format (Times Roman; 12 pt type on 14 pt leading; no bold, italics, strikethrough or underline) is used.

BULLET LISTS

Platypus allows you to create bullet lists easily. These lists can be nested, and the bullet character(s) can be specified by the user.

The `[list]` command begins a bulleted list and the `[-list]` command marks the end of the list. The bulleted items between these commands are separated using the hard end-of-line, `[]`. The default bullet character is the standard black bullet: •

The states of New England:	→	The states of New England:
<code>[list]Connecticut[]</code>		• Connecticut
<code>Rhode Island[]</code>		• Rhode Island
<code>Massachussets[]</code>		• Massachusetts
<code>Vermont[]</code>		• Vermont
<code>New Hampshire[]</code>		• New Hampshire
<code>Maine[][-list]</code>		• Maine

Nested Lists. Bullet lists can nest, with no limit on the number of levels except for that imposed by the width of the page or the column. To embed one list in another, simply use additional `[list]` and `[-list]` commands:

Main Parts of the Book:	→	Main Parts of the Book:
<code>[list]Front Matter[]</code>		• Front Matter
<code>[list]Cover[]</code>		• Cover
<code>Copyright notice[]</code>		• Copyright Notice
<code>Title page[]</code>		• Title Page
<code>Preface[][-list]</code>		• Preface
<code>Chapters 1-16[]</code>		• Chapters 1-16
<code>Bibliography[][-list]</code>		• Bibliography

Bullet Character. You can specify a different bullet character from the default, by specifying it in the `[list]` command with the `bullet` option. For example, `[list|bullet:>]` generates a list that uses the > character as a bullet. Note that you can use more than one character for a bullet. For example, `[list|bullet:>>]` will create bullets consisting of >>.

You can also specify bullet characters by their Platypus name, by putting the name in braces. For example `[list|bullet:{rarrow}]` will create bullets consisting of a right arrow: →. A complete list of symbol names are presented elsewhere in this document. Platypus has several built-in bullets. These are:

<code>{bullet}</code> (the default)	•
<code>{dot}</code>	•
<code>{ring}</code>	◦
<code>{asterisk}</code>	*
<code>{bkstar}</code>	★
<code>{bksquare}</code>	■
<code>{bksmrttriangle}</code>	►
<code>{bkrttriangle}</code>	▶

Indent. The amount of indent for items in a bullet list is fixed at: 0.25" (0.635 cm or 18 pts).

CODE AND LISTINGS

Platypus is particularly good at printing source code. Code can be included in running text or formatted into elegant, easy-to-read listings.

To insert code in running text, simply add the `[code]` command before the code and `[-code]` at the end. The effect of this command is to change the current font to monospace of the appropriate size.

The variables `[code]ohms[-code]` and `[code]farads[-code]` should be set to 0.

→ The variables `ohms` and `farads` should both be set to 0.

The `[code]` command saves the current format and restores it automatically when the `[-code]` command is encountered.

Code listings are defined by the `code` command with options that define line numbering. The command syntax is: `[code|lines:1,5]` The two numbers after the `lines:` parameter define, respectively, the number of the first line of code in the listing, and the line multiple on which line numbers should appear. For example, `[code|lines:1,1]` will print a listing in which every line is numbered, starting with the first line numbered as 1. `[code|lines:1,2]` will print numbers on every other line. Listings end when the `[-code]` command is encountered.

```
[code|lines:1,5]
int binarySearch( int a, int value )
{
    int low = 0;
    int high = a.length - 1;
    while( low <= high ) {
        int mid = low + (high - low)/2;
        int midValue = a - mid;
        if( midValue ) {
            low = mid + 1;
        }
        else {
            return mid;
        }
    }
}
[-code][]
Listing of [+i]incorrect[-i] binary search.
```

generates the following result:

```

1. int binarySearch( int a, int value )
. {
.     int low = 0;
.     int high = a.length - 1;
5.     while( low <= high ) {
.         int mid = low + (high - low)/2;
.         int midValue = a[mid];
.         if( midValue == value ) {
.             low = mid + 1;
10.        }
.         else {
.             return mid;
.         }
.     }
15. }

```

Listing of *incorrect* binary search.

Notice that that when numbered lines are specified, the first line always receives a number. Lines without numbers are marked by a small dot, to facilitate counting. If you are print several listings and you want Platypus to continue the line numbers across the listings, the first parameter to `lines:` should be 0. Then, if the previous listing ended at line 41, for example, the current listing will start with line 42.

USER-DEFINED STRINGS (MACROS)

Platypus offers a string substitution capability, similar in concept to macros in a variety of programming languages. You can define a variable and assign a string to it. Later, you can print that string by simply referencing the variable. These user-defined strings are called macros in Platypus.

Macro names must start with a letter (either upper-case or lower-case). Thereafter, any sequence of letters, digits, or the `_` character is permitted. Macro names are case-sensitive. The minimum length is two characters.

Valid names: `PageSize`, `TaxesDue`, `iq`

Invalid names:

`_mine` (first character must be a letter or digit)

`1/2nelson` (may only contain a letters, digits, or `_`)

`4Seasons` (may not start with a digit)

`My Macro` (no embedded spaces allowed)

Here is how to define a macro:

```
[def:SampleMacro{This is sample macro text.}]
```

Note that the value of the variable is enclosed in curly braces `{ }` and follows immediately after the name. It can contain any characters except `}`.

To print the variable contents, use an asterisk. Using the previous macro `[*SampleMacro]` results in:

```
This is sample macro text.
```

being printed to the document.

Another example: The font list on page 22 was generated in part using this feature:

...

```
[def: FontText{The Quick Brown Fox Jumped...1234567890}][[]]
```

...

```
[ff:TIMES_ROMAN]Helvetica[]  
[ff:HELVETICA][* FontText][[]]  
[]  
[ff:TIMES_ROMAN]Helvetica Italic[]  
[ff:HELVETICA][+i][*FontText][-i][[]]
```

```
[  
[ff:TIMES_ROMAN]Helvetica Bold[  
[ff:HELVETICA][+b][*FontText][-b][
```

In some places, it might be desirable to access the macro without printing it. This is done with the `[$` command. For example, in URLs that have cover text, the cover text can be specified using a macro. This is done this way: `[$macroName]`

Note: this feature is scheduled to be greatly expanded in upcoming releases.

SYSTEM-DEFINED STRINGS

Platypus internally uses several system values that can be printed to a document. Like user-defined strings, these values can be printed to a document with the `[*` command. So, for example, to print the current version number of Platypus in a document, use: `[*_version]`

The printable system values are:

- `_version` The version of Platypus being run. This is the same value that appears on the console when Platypus is run.
- `_format` The format of the output document. Current possibilities are: `PDF` or `LISTING`
- `_commandLine` The command-line arguments for this run of Platypus. This value consists of all the text typed at the command line after the command (the underlined portion of the following example).

```
java -jar Platypus.jar input.txt output.pdf -verbose
```

(Note: Backslashes will appear doubled. So `\doc\input.txt` becomes `\\doc\\input.txt`. This effect is due to how Java handles backslashes.)

DEBUGGING PLATYPUS RESULTS

Sometimes, when you're not getting the output you expect, it's useful to have commands that let you peer into the Platypus operations. The following commands print data to the console (display screen) about various aspects of Platypus operation at the point where the command was encountered.

`[dump:all]` displays all values that can be shown including all those below. Be careful with this command, it can potentially output lots of data.

`[dump:commandLine]` displays Platypus version number, the command line used to invoke this run of Platypus, and the output format.

`[dump:page]` displays page size, width of the margins, number of columns, etc. for the current page

`[dump:paragraph]` displays text alignment, left and right indent, first-line indent, etc. for the current paragraph

`[dump:userStrings]` displays all user-defined strings (macros) and their respective values.

Dump commands can be placed anywhere in the document where a command is legal.

Most displayed data shows not only an item's value, but the file and line number where it was last changed. For example, this sample output from the `[dump:page]` command:

```
page width: 595.0 points last changed at line 212
page height: 842.0 points last changed at line 212
margins:
  top: 72.0 points last changed at line 0
  bottom: 72.0 points last changed at line 0
  left: 72.0 points last changed at line 0
  right: 72.0 points last changed at line 0
```

line 0 indicates that the value was set at program start up (i.e., it's the default value), or it was set by the user on the command line.

APPENDIX A: PLATYPUS COMMANDS (ALPHABETICAL ORDER)

(Note: The list of symbols and foreign characters appears after these commands.)

<code>[]</code>	New paragraph. (A blank line in text is equivalent.)
<code>[\$MacroName]</code>	Prints the text contained in the macro <code>MacroName</code> to the document.
<code>[+b][-b]</code>	Turn on bold on or off
<code>[+footer] [-footer]</code>	Turn the footer on or off
<code>[+i] [-i]</code>	Turn on italics on or off
<code>[+marginmirror]</code>	Mirror the margins of the right page on the left page. Default: off; that is, all pages have the same margins.
<code>[+st] [-st]</code>	Turn strikethrough text on or off
<code>[+u] [-u]</code>	Turn on underlining on or off
<code>[align:center]</code>	Align text so that it's centered
<code>[align:just]</code>	Justified text: paragraph is aligned with both the left and right margins
<code>[align:left]</code>	Align text to the left margin
<code>[align:right]</code>	Align text to the right margin
<code>[bmargin:</code>	Specify the bottom margin of a page. (Default: 1 inch; 2.54 cm)
<code>[code]</code>	Start code/listing format
<code>[-code]</code>	End of code/listing format. Revert to previous style.
<code>[def:\$MacroName{Hi!}]</code>	Defines a user-defined string or macro named <code>\$MacroName</code> containing the string <code>Hi!</code>
<code>[dump:</code>	Write the value of a variable or group of variables to the console. Used for debugging Platypus documents.
<code>[ff:typeface-name]</code>	Change to the specified typeface
<code>[fsize:</code>	Change the size of the current font to the specified size.
<code>[indent:</code>	Indent the first line of the following paragraph by the specified amount.

<code>[leading:</code>	Specify the leading (space between lines).
<code>[lmargin:</code>	Specify the left margin of a page. (Default: 1 inch; 2.54 cm)
<code>[noindent]</code>	Do not indent the first line of the current paragraph (if the command appears before any text has been written to the paragraph). If text has been written to the current paragraph, this command affects the indentation of the <i>next</i> paragraph, which is unlikely to be correct.
<code>[pageheight:</code>	Specify the page height (for custom page size).
<code>[pagesize:</code>	Page size. Applies to the next page to be rendered.
<code>[pagewidth:</code>	Specify the page width (for custom page size).
<code>[paraindent:</code>	Indent paragraph from the left margin by ... units
<code>[paraindentR:</code>	Indent paragraph from the right margin by ... units
<code>[paraskip:</code>	Skip ... lines before start of next paragraph.
<code>[pg]</code>	Insert new page and skip to the top of the top of it.
<code>[restorefmt]</code>	Restore a previously saved format. See <code>[savefmt]</code>
<code>[rmargin:</code>	Specify the right margin of a page. (Default: 1 inch; 2.54 cm.)
<code>[savefmt]</code>	Save the current format (for later retrieval).
<code>[tmargin:</code>	Specify the top margin of a page. (Default: 1 inch; 2.54 cm.)

Symbols and International Characters Currently Implemented

	á		â		à		ä		Á		Â		À		Ä
	[a']		[a^]		[a`]		[a"]		[A']		[A^]		[A`]		[A"]

	ç		Ç	
	[c_]		[C_]	

	é		ê		è		ë		É		Ê		È		Ë
	[e']		[e^]		[e`]		[e"]		[E']		[E^]		[E`]		[E"]

	í		î		ì		ï		Í		Î		Ì		Ï
	[i']		[i^]		[i`]		[i"]		[I']		[I^]		[I`]		[I"]

	ñ		Ñ	
	[n~]		[N~]	

	ó		ô		ò		ö		Ó		Ô		Ò		Ö
	[o']		[o^]		[o`]		[o"]		[O']		[O^]		[O`]		[O"]

	ú		û		ù		ü		Ú		Û		Ù		Ü
	[u']		[u^]		[u`]		[u"]		[U']		[U^]		[U`]		[U"]

	ı		¿		ß
	[!^]		[?^]		[ssharp]

Greek letters

α[alpha] β[beta] γ[gamma] δ[delta] ... ψ[psi] ω[omega]
ς[sigmavar] ϑ[thetavar]
Α[Alpha] Β[Beta] Γ[Gamma] Δ[Delta] ... Ψ[Psi] Ω[Omega]

Dashes and Quotation Marks

	-		-		"		"		`		'	
	[--]		[---]		[dlq]		[drq]		[slq]		[srq]	

Commerce

™[trademark] ®[registered]
©[copyright]

Currency

\$ \$,
¢ [cents]
¤ [currency]
€ [euro]
£ [sterling]
¥ [yen]

Ligatures

ae: æ[ae] - Æ[Æ]
fi: fi[fi]
fl: fl[fl]
oe: œ[oe] - Œ[Œ]

Bullets

•[bullet]
·[dot]
◦[ring]
*[asterisk]
★[bkstar]
▸[bksmrttriangle]

►[bkrttriangle]
▪[bksquare]

Arrows

←[larrow] →[rarrow] ↔ [lrrarrow]
⇐[lArrow] ⇒[rArrow] ⇔[lrArrow]
⇐[larrrows] ⇒[rarrrows] ⇔[lrrarrrows]

↑[uarrow] ↓[darrow] ⇕[udarrow]
↑[uArrow] ↓[dArrow] ⇕[udArrow]
↑↑[uarrrows] ↓↓[darrrows] ⇕⇕[udarrrows]

↗[nearrow] ↘[searrow] ↖[nwarrow] ↙[swarrow]
↖[ularrow] ↗[urarrow] ↘[dlarrow] ↙[drarrow]

Miscellaneous

... [dots]

APPENDIX B: LICENSES

Platypus is a copyrighted work. It is not in the public domain. However, it can be used freely as long as you adhere to the terms of its license.

Platypus uses the Apache 2.0 license, which is one of the most liberal open-source licenses available. The text of the Apache 2.0 license can be found at:

<http://www.apache.org/licenses/LICENSE-2.0.html>

Platypus uses several open-source packages, most notably, the iText library for generating PDF files. The licenses and download information for all these packages are available in the `/licenses` directory of the installation.