Platypus

Page layout and typesetting system

User's Guide

v. 0.1.3.0 (Milestone 1)

Note on Platypus Features

Platypus is a page layout and typesetting system is built around two principles: ease of use and a comprehensive feature set. At present, it is in the very 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, see the title page.

Here are some features that are planned for upcoming releases:

- Built-in macro language
- Built-in scripting language
- Template facilities
- Advanced support developers including:
  - Intelligent language-based line breaks in listings, with symbols to show the break clearly
  - Ability to embed markers such as ❶ and ❷ into listings for reference
  - Various schemes to facilitate line numbering
- Extensive support for foreign character sets, including full UTF-8
- Output to PDF, HTML, and RTF (Microsoft Rich Text Format)
- Automatic intelligent numbering of figures, listings, tables, etc.
- Embeddable images
- Fancy typesetting features such as watermarks, multidimensional tables, etc.
- Ability to use any TrueType and OpenType fonts on the system
- Clickable links to bookmarks inside the document and URLs outside
- Debugging features (commands to help you understand why your document is not coming out how you expect)

In addition you'll find all the standard layout and typesetting features including adjustable page sizes, margins (mirrored and regular), text alignment, text formatting (bold, italics, underline, and strikethrough), document styles, headers/footers, hyphenation with support for many languages, color text, color background, subscripts and superscripts, equations, columns (regular and serpentine), tabs, table of contents, multi-level index, and so on.

A projected time line for implementation of many of these features can be found at: http://platypus.pz.org/schedule.pdf The project is able to stay on schedule, in large part because of the good design of the iText library, which renders Platypus documents in PDF.

To be kept abreast of the releases, sign up for our low-volume announcement mailing list by sending an e-mail to announce-subscribe@platypus.tigris.org. And soon, begin creating typeset documents with ease.

Introduction

Platypus accepts an input text file containing text and embedded commands; it then generates high-quality PDF files. This approach is similar to TeX and troff, which are both good programs, although more difficult to use than Platypus. In addition, Platypus, when complete, will contain several important features missing in these other tools.

Platypus is not a word processor. It does not use a graphical interface. Its only input is a text file with embedded commands. 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 indent levels of bullets, or the like. Platypus takes care of formatting and layout for you.

Another advantage of this approach is that you control the fundamental document. You cannot do this with word processors: Have you ever looked inside a document created by Microsoft Word? 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 true open-source software. That is, it is available at no cost from http://platypus.pz.org. You can download the source code from the same site.

We think the more you use Platypus, the happier you'll be creating great documents.

—The Platypus Team
platypus@pz.org

Running Platypus

Platypus is a Java application that requires Java 5.0 SE (also called Java 1.5) or a later version. Platypus is delivered as a Java archive, or JAR file.

Download Platypus[version].zip and unzip the file in a new directory. It should contain: Platypus.jar, iText.jar, readme1st.txt, and additional files.

To run Platypus, go to the directory where you unzipped the files, and enter this command line:

```
java –cp . –jar Platypus.jar input-file output-file [options]
```

where:

`input-file` is any file containing text and Platypus commands

`output-file` is the resulting file. The file extension tells Platypus what format to use. Currently only PDF is supported, although HTML and RTF will be supported later. So, for the time being, output-file must have a .pdf extension or an error results.

`[options]` are command-line switches that enable various options. Although many more switches are planned, only the following ones are currently supported:

`-psize` Page size. Platypus defaults to a page size of 8 ½" x 11", the so-called "letter" size in the United States. You can over-ride this default by using this command line switch or using the `pagesize` command in the document itself. Common values are:

```
-psize A4
-psize LETTER
-psize LEGAL
-psize HALFLETTER
-psize 11X17
```

(For a description of all 50 predefined values, see the section on page size.) If you plan to generate a PDF documents in multiple different sizes, we recommend that you modify the page sizes using this switch, rather than embedding the page size in the document.

In the event, the page size is specified in both at the command line and the document, the page size in the document controls.

`-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.

Platypus Commands

All Platypus commands have the form `[command]`. That is, commands are placed inside a pair of square brackets, without exception.

Here is an example of text containing Platypus commands:

> **`[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.

Syntactically, command can be grouped into several categories:

- Switches that turn a feature, such as bold, on or off: `[+b]` and `[-b]`
- Commands that specify modify output: `[noindent]`
- Commands that require a parameter: `[align:left]`as seen above. These commands use a colon to separate the command name from the parameter.
- Commands with special syntax, such as `[]`.

The next sections presents Platypus commands, grouped by theme. Appendix A lists the commands in alphabetical order.

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: `/[`not a command`]`.

To print the two characters `/[` (how often will that happen?) 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 interpreted as text. This means that to output [26] to a document requires no special escape character. This feature is designed to enable Platypus to process a greater range of pure text files without unexpected side effects. Some users of Platypus prefer to place a `/` before every `[` that is not a command. This works fine as well.

Comments

Platypus uses both line comments and block comments.

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

Block comments, which can appear on a single line or span many lines, start with `[%%` and end with `%%]`. Platypus does handle 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 `[%%` -based blocks. Likewise, adding more % signs to the beginning and ending blocks will comment out text containing block comments with fewer % signs. (We thank the designers of the Lua language for this excellent solution to embedded comments.)

Units of Measure

Many Platypus commands use numeric values. When these values refer to distance or to size, they can use one of four types of units: centimeters, inches, lines, and points. These are abbreviated 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" (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 some sample examples, all of which communicate exactly the same distance:

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

The line (`li`) unit of measure is not included in this table because the size of a line depends on how it's been set at any given point in the document. 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 place between lines of text. A typical arrangement of text and leading is 12-point text on 14-point leading.

Page Size

Platypus defaults to a pages 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. These sizes are listed in Table 1. In addition, custom sizes can be specified.

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
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
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

Table 1. The predefined page sizes in Platypus.

To specify one of these sizes, use the pagesize command followed by the value in the leftmost 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 output document is opened; that is, when the first text is specified. Consequently, page size should be specified in the document *before* any text. If you plan to print the same document in several different sizes, it is better to specify page size on the command line, rather than hard-coding it into the document.

The `[dump:page]`command will write out data about the page size and the margins to the system console, as a help in debugging.

Page Margins

Platypus use default margins of 1 inch on the top, bottom, and both sides of the page. These margins can be modified using the following commands. `[tmargin: …]` for the top margin, `[bmargin.: …]` bottom margin, `[lmargin: …]` left margin, and `[rmargin: …]` for the right margin. For example, the commands:

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

create top and bottom margins of ½" each.

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 `[+marginmirror]`, 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 mirror must be specified before the first text. If placed after text has been output, the commands are ignored.

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 (generally used for headings)
`[align:right]`	text is aligned on the right margin
`[align:just]`	text is aligned on the left and right margins (space between words can sometimes be uneven with this option).

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

New Paragraph

There are two ways to indicate a new paragraph in Platypus: via the `[]` command or by insertion of a blank line. Hence, these two snippets are equivalent:

```
This is the end of the first paragraph.[]This is the begin-
ning of the second.
```

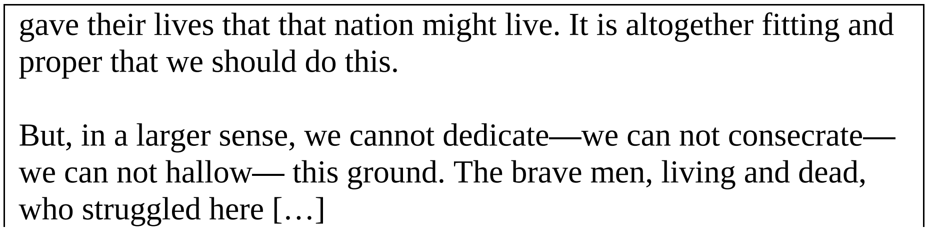
and:

```
This is the end of the first paragraph.

This is the beginning of the second.
```

Two commands determine how a new paragraph appears on the printed page. There are two principal styles for starting a paragraph: block style and traditional.

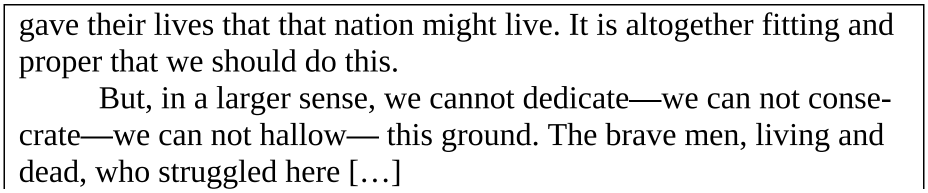
Block style is frequently used in letters and memoranda. In this style (used in this guide), a line is skipped before a new paragraph, and the first line of the new paragraph is not indented:



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

But, in a larger sense, we cannot dedicate—we can not consecrate—we can not 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 commonly found in books, magazines, and other documents consisting of more than a few pages. At one time, it was the standard for letters and memoranda, but that use has fallen off considerably. Here's how it looks:



gave their lives that that nation might live. It is altogether fitting and proper that we should do this.
        But, in a larger sense, we cannot dedicate—we can not consecrate—we can not hallow— this ground. The brave men, living and dead, who struggled here [...]

These ability to indent and to skip a line for a new paragraph are controlled in Platypus by two commands: `paraindent` (which indicates how much to indent the first line) and `paraskip` (which specifies how much space to skip between paragraphs).

`[paraindent:36pt]` This command tells Platypus to indent the first line of all following paragraphs by 36 points (or half an inch). As usual, you can specify the distance in inches, centimeters, or points. If no unit is specified, the default is points. To use block-style new paragraphs, this value should be set to zero: `[paraindent:0]`

Sometimes you have indent set up, but you don't want the first line of a specific paragraph indented. This frequently occurs in the first paragraph right after a section heading or in a new chapter. In this case, use `[noindent]` command, which skips indenting the first line of only next paragraph. Note: this command must appear before new text in the paragraph. If it appears after some text has appeared it affects the following paragraph, which is unlikely to be correct.

`[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, these two commands 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.

New Page

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

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

Fonts

In common parlance, the term font generally refers to a typeface. We're fond of saying "the Arial font." But in typesetting, the term font refers to the specific instance of a typeface. It includes data about the size of the letters and certain formatting attributes, such as bold or italic. A font, as referred to by a typesetter, might be: 12-point Arial Extra Bold. Platypus uses this expanded typesetter's definition when referring to fonts. When Platypus uses the term typeface, it is referring to Arial, Times Roman, or Courier—what is commonly thought of as font families.

By default, Platypus uses 12-point Times Roman medium. (Medium means that there are no special formatting characteristics: it's neither bold nor italic.)

Changing Typefaces

At present, Platypus supports the following typefaces: 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 following typeface commands are valid. (Note that all the typefaces are in capitals.)

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

The first three typefaces are alphabetic and have bold, italic, and bold-italic variants (for a total of 12 alphabetic fonts). The last two typefaces are symbols, which don't support bolding or italics. In total, therefore, Platypus has 14 fonts currently available. (These are illustrated on the following page.) These fourteen fonts, known in Adobe's language, as the "Base 14 fonts" are guaranteed to be available in all PDF readers. Any other fonts might or might not be available, depending on several factors. Support for fonts beyond the Base 14 is scheduled for an upcoming milestone of Platypus.

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 also that you have correctly specified the name of the font.

The Basic 14 Fonts Currently Available in Platypus

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 θυιχκ βροων φοξ φυμπεδ . 12345

Dingbat
▼❈❅ ❑◆❉❃❋ ✲❐❒◗■ ❆❑❙ ❊◆❍❐❅❅ ✎ ✐✒✓✔✗

Changing the 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.

When the font size is changed, the leading (or space between lines) is changed as well. It is set to 1.2 times the size of the font. So, changing a font size to 15 points causes leading to change to 18 points.

Changing the Leading (The Spacing Between Lines)
As explained in the previous section, Platypus automatically adjusts leading to fit the font you're using. However, if you want to override this setting, you can do so with the leading command. This is useful if you want to double space text, which would use this command:

> [leading:2li]

To revert back to single-spaced text later, use [leading:3li]. The 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.

Changing 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 TeX and other page-layout languages. For example,

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

generates:

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

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

Underline

Underlining is turned on using `[+u]` and turned off using `[-u]`. This command result 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.
```

Results in:

> 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.

User-Defined Strings (Macros)

Platypus offers a limited 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.

Variable names must start with a capital letter followed by one or more letters, digits or underscore characters. Names are case-sensitive and all letters are meaningful.

Valid names: `PageSize, TaxesDue2007, IQ`

Invalid names:

`height` (must capital letter)
`A1/2nelson` (may only contain letters, digits, or _)
`My Macro` (no embedded spaces allowed)

The substitute string is enclosed in curly braces `{ }` and follows immediately after the name. This string can contain any characters except `}`. At present, if the string contains `[` characters, these are processed as text by Platypus. (This will change in future releases so that `[ ]` translation conforms with that currently used for commands. So at present, it's best not to embed `[ ]` in macro text.)

Here is how to define a variable:

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

To print the variable contents, use an asterisk: `[*SampleMacro]` results in: `This is sample macro text.` being printed to the document.

The font list on page 14 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][]
```

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

To list all the user-defined strings on the console, use the following command at any point in the document:

```
[dump:userStrings]
```

This will print all the user-defined strings, with the variable name and the replacement string together. For example, when run on the previous example, this command outputs:

```
User-defined Strings
     $FontText: The Quick Brown Fox Jumped...1234567890
```

on the console.

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. Currently, there are three commands that let you print the values of key variables to the console.

These are:

`[dump:page]` which prints out the variables relating to the size of the page and the margins.

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

`[dump:all]` which dumps all values that can be dumped including those for pages and strings, as well as a few others. Be careful with this command, it can potentially output lots of data. The `–vverbose` command-line switch forces a `[dump:all]` at the end of a Platypus run.

Dump commands can be placed anywhere in the document where a command is legal. Eventually, all user-settable variables will be dumpable.

Appendix A: Implemented Platypus Commands

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

[]	New paragraph. (A blank line in text is equivalent.)
[*$MacroName]	Prints the text contained in the macro MacroName to the document.
[+b]	Turn on bold
[+i]	Turn on italics
[+marginmirror]	Mirror the margins of the right page on the left page. Default: off; that is, all pages have the same margins.
[+st]	Turn on strikethrough
[+u]	Turn on underlining
[-b]	Turn off bold
[-i]	Turn off italics
[-st]	Turn off strikethrough
[-u]	Turn off underlining
[align:center]	Align text so that it's centered
[align:just]	Justified text: paragraph is aligned with both the left and right margins
[align:left]	Align text to the left margin
[align:right]	Align text to the right margin
[bmargin: …]	Specify the bottom margin of a page. (Default: 1 inch)
[def:MacroName{Hi!}]	Defines a user-defined string or macro named MacroName containing the string Hi!
[dump: …]	Write the value of a variable or group of variables to the console. Used for debugging Platypus documens.
[ff:typeface-name]	Change to the specified typeface
[fsize:14pt].	Change the size of the current font to the specified size (here, 14 point).
[leading: …]	Specify the leading (space between lines).
[lmargin: …]	Specify the left margin of a page. (Default: 1 inch)
[noindent]	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 *next* paragraph, which is unlikely to be correct.
[pageheight: …]	Specify the page height (for custom page size).
[pagesize: …]	Page size. Must be specified before any text. Also, see command line switch -psize for controlling page size.

`[pagewidth: …]`	Specify the page width (for custom page size).
`[paraindent: …]`	Indent paragraph’s first line by … units
`[paraskip: …]`	Skip … lines before start of next paragraph.
`[pg]`	Insert new page and skip to the top of the top of it.
`[rmargin: …]`	Specify the right margin of a page. Default: 1 inch.
`[tmargin: …]`	Specify the top margin of a page. Default: 1 inch.

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"]

|  ¡   |  ¿   |    ß
| [!^] | [?^] | [ssharp]

© [copyright]
® [registered]
```

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

In addition, the iText library, which does a lot of the magic that Platypus gets credit for is licensed under the Mozilla license 1.1. You are required to follow the terms of this license as well as the license for Platypus proper. Text of the Mozilla 1.1 license is located at: http://www.lowagie.com/iText/MPL-1.1.txt

Fortunately, both licenses are liberal. They encourage use for all legal purposes; they require you to include and publish the licenses and copyrights with any software you distribute that includes Platypus or iText code or binaries; and they forbid you from taking credit for having written these products.

The licenses also explicitly remove any liability from the authors: in other words, you use this software at your own risk.

Finally, because the source code for Platypus and iText is copyrighted, you cannot use it in your programs without specifically including the copyright notice from the original source code.

Before relying on this synopsis of your rights and responsibilities, please consult the actual text of the licenses. In cases of conflicts between the licenses and this summary, the terms of the license control.