

## NAME

platypus - create Mac OS X application wrapper around a script.

## SYNOPSIS

```
platypus [-vhASDBR] [-a name] [-t type] [-o type] [-i icon]  
          [-p interpreter] [-V version] [-s signature] [-I identifier]  
          script destination
```

## DESCRIPTION

platypus is the command line counterpart of the Platypus Mac OS X application, and serves the function of creating Mac OS X application (.app) bundles around scripts.

platypus supports the following flags:

-a [name]

Specifies the name of the application to be created. This is can be different from the name of the .app bundle itself, and is displayed in the application's menus, about window and Info.plist property list.

-t [name]

Specifies the script type. platypus is quite clever about figuring this out on its own. It first tries to parse the shebang line (!) of the script to detect this. If this fails, a guess is made based on the script's suffix (e.g. .pl or .sh). If your script has neither a shebang line or a suffix, this option will allow you to specify the script type. The following values (case insensitive) are supported:

shell	Shell scripts, run with interpreter /bin/sh
python	Python scripts, run with interpreter /usr/bin/python
applescript	AppleScripts, run with interpreter /usr/bin/osascript
expect	Expect scripts, run with /usr/bin/expect
perl	Perl scripts, run with interpreter /usr/bin/perl
ruby	Ruby scripts, run with /usr/bin/ruby
tcl	Tcl scripts, run with /usr/bin/tclsh
php	PHP scripts, run with /usr/bin/php

-o [type]

Specifies the script application's output type. Three different kinds of (case insensitive) values are supported as arguments:

None The application has no graphical output. Script text output

goes to STDOUT.

ProgressBar The application displays an indeterminate progress bar. Script output goes to STDOUT.

TextWindow The application displays a window with a text field which receives all script output.

-i [icon]

Specifies a file to use as icon for the application bundle. The icon file can be a regular Mac OS X .icns file, or an image file of any kind supported by the Mac OS X Cocoa APIs. As of writing, this includes JPEG, GIF, PNG, TIFF, PSD, BMP, PICT, TGA, SGI, PNTG PDF, EPS and PostScript (.ps). The image will be scaled to 128x128 pixel size, which is standard for Mac OS X icons.

platypus has its own set of default icons for different script types (stored in /usr/share/platypus/icons) which it will use if this option is ignored.

-p [interpreter]

This option allows you to specify an arbitrary interpreter for your script. This option will override the -t option, making it (in theory) possible to bundle any interpreted script.

-V [version]

Sets the application bundle's version. This is reflected in the application bundle's Info.plist property list and about window.

-u [author]

Sets the name of the application author (e.g. "Apple Computer" or "John Smith"). If not specified, platypus will default to the current user's full user name.

-s [signature]

Mostly for backwards compatibility and heritage reasons. In Mac OS 9 and earlier (and, indeed, in Mac OS X), each application was assigned a four character ASCII signature known as an OSType. It is possible to register application signatures in a database on the Apple Developer Website. Please note that Apple has reserved all lowercase-only combinations.

Chances are, though, that you'll have no need of this option. If it is left empty, the application will get the signature string '????', which is ignored by Mac OS X. This value is stored within the

application bundle in the Info.plist application property list and PkgInfo files.

-I [identifier]

Sets the application's bundle identifier. An application identifier is a kind of reverse DNS name (e.g. com.apple.iTunes). If this option is left empty, platypus will default to an identifier of the format "org.username.appname" (e.g. org.sveinbjornrnt.Platypus).

-A This flag makes the application request Administrator privileges via Apple's Security Framework (i.e. prompt for a password) and then execute the script with those privileges.

-S Secure bundled script. Securing a bundled script will make the platypus-generated application verify the script file's checksum to ensure its integrity when launching. Furthermore, the bundled script is hidden by prepending a '.' to the filename and subsequently encrypted using a simple encryption algorithm (specifically, a variation of the 'rot13' algorithm).

Please note that none of these precautions will make the script inaccessible to an experienced hacker, but should serve to dissuade the less determined.

-D Makes your script application "droppable", i.e. capable of receiving drag'n dropped files as arguments to the script. The application bundle's property list is modified so that it can receive dropped files in the Dock and Finder. These files are then passed on to the script as arguments via @ARGV. However, the first argument to the script (\$ARGV[1], \$1 etc., depending on your scripting language of choice) is always the path to the application bundle (e.g. "/Applications/MyPlatypusApp.app").

-B This option causes the application to run in the background. The application will not appear in the Dock or bring its windows to the front when launched. This is achieved by registering the application with the Finder as a user interface element (LSUIElement).

-R This option sets the application so that it remains running after the script has been executed. It can then later be quit by the user via regular means (i.e. Quit menu item).

-v Prints the version of the platypus command line utility

-h Prints help and usage string

Please direct any comments, questions or queries to Sveinbjorn Thordarson <sveinbt@hi.is>.

#### FILES

/usr/local/bin/platypus	program binary
/usr/local/share/platypus/exec	executable for script app
/usr/local/share/MainMenu.nib	Nib file for script app
/usr/local/share/icons/	Default icons

#### AUTHORS

This manual page was written by Sveinbjorn Thordarson <sveinbt@hi.is>.

Darwin

May 6, 2005

Darwin