NAME

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

SYNOPSIS

```
platypus [-vh] [-O profile] [-FASDBR]] [-ydlH] [-bgn] [-KYL] [-P profile]
         [-a appName] [-c scriptPath] [-o outputType] [-i icon]
         [-p interpreter] [-V version] [-u author] [-s signature]
         [-I identifier] [-f bundledFile] [-X suffixes] [-T filetypes]
         [-G interpreterArgs] destinationPath
```

DESCRIPTION

platypus is the command line counterpart to the Platypus Mac OS X application. It is a tool that creates a Mac OS X application bundles which executes a script.

platypus defaults to creating application bundles, but can also create Platypus profiles that can subsequently be loaded by either the command line program or the Platypus graphical application.

platypus supports the following flags (parsed in the order in which they are passed):

-O Create profile file as output instead of actually creating an application bundle. When you select this option, the "destinationPath" paramater (i.e. the final parameter to the program) must have a .platypus suffix. Please note that in order to create reusable profiles, this option requires that all paths to bundled files, icons etc. be absolute. Profiles may or may not work in older/newer versions of the program.

-P *[profile]*
Loads all settings from a Platypus profile file. You will still need to specify a destination path for the application. Individual settings in the profile can be overridden with subsequent parameters.

-a *[appName]*
Specifies the name of the application. 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.

-c *[scriptPath]*
Path to the script that is to be wrapped into an application.

-o *[outputType]*
Specifies the script application's output type, which can be any of the following:

'None' The application has no graphical output.

'Progress Bar' The application displays a progress bar while running.

'Text Window' The application displays a window with a text field with all script output.

'Web View' The application displays a window with a web view which renders output as HTML.

'Status Menu' The application displays a status menu item in the menubar which runs script and displays output when selected.

'Droplet' The application displays a droplet window for dropping files on for processing.

-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 files. The image will be scaled to the standard Mac OS X icon size.

Please note that specifying a non-icns image will not produce icons that look good in smaller sizes. For best results, use professional Mac OS X icon-editing software and specify a carefully crafted .icns file. If this option is left unspecified, the default Platypus app icon is used.

−p *[interpreter]*
Sets the interpreter for your script (e.g. /usr/bin/perl). If no interpreter is specified, the default is the standard shell (/bin/sh).

−V *[version]*
Sets the application bundle's version. This is displayed 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, it defaults 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 is 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 that you will 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.

−f *[bundledFile]*
Specifies a file to be bundled with the script application. This file will be copied over the Resources folder of the application bundle, which is the same folder in which your script will eventually reside. You can specify any number of files to be bundled, but you must use the -f flag each time, since the flag only accepts one file path after the -f flag.

−I *[identifier]*
Sets the application's bundle identifier. An application identifier is a reverse DNS name (e.g. com.apple.iTunes) that uniquely identifies your application. If this option is left empty, it will default to an identifier of the format "org.username.appname" (e.g. org.sveinbjorn.Platypus).

−F If set, your script will receive the path to the application's Resources folder (e.g. "/Applications/MyPlatypusApp.app/Contents/Resources") as the first argument ($ARGV[0], $1 etc., depending on your scripting language of choice).

−A This flag makes the application request administrator privileges via Apple's Security Framework (i.e. prompt for a password) and then executes the script with those privileges. For details on the nature of these privileges, see the Apple documentation for the AuthorizationExecuteWithPrivileges() in Security.framework.

−S Secure bundled script. This encrypts the bundled script, to fend off prying eyes. Please note that this will NOT make the script 'secure' from anyone knowledgable enough to get a clear text version, but it just might serve to dissuade the less determined.

−D Makes your script application droppable, i.e. capable of receiving drag and 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.

−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 done by registering the application with the Finder as a user interface element (LSUIElement). For this reason, the application's windows will not move to the front when it is launched.

−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. the Quit menu item).

−X *[suffixes]*
Only appropriate if you are using the -D option. This flag allows you to specify the file suffixes (e.g. .txt, .wav) your application can open. This should be a |-separated string (e.g. "txt|wav|jpg").

−T *[filetypes]*
Only appropriate if you are using the -D option. This flag allows you to specify the file type codes (e.g. ’8BIM’, ’TEXT’) your application can open. This should be a |-separated string of 4-character ASCII strings (e.g. "8BIM|TEXT|R*ch").

−G *[arguments]*
Arguments for the script interpreter. These should be specified as a |-separated string (e.g. ’-w|-s|-l’).

−b *[hexColor]*
For Text Window output mode only. Set background color of text output (e.g. #ffffff).

−g *[hexColor]*
For Text Window output mode only. Set foreground color of text output (e.g. #000000).

−n *[fontName]*
For Text Window output mode only. Set font and fontsize for text output field (e.g. ’Monaco 10’).

−E *[encodingNumber]*
For Text Window output mode only. Set text encoding for script output. Must be one of the numbers defined in the NSString class reference (e.g. 4=UTF-8, 1=ASCII, etc.). Default is UTF8.

−K *[kind]*
For Status Menu output mode only. Set display kind for Status Menu output mode. This can be "Text", "Icon and text" or "Icon".

−Y *[title]*
For Status Menu output mode only. Set the display title for the status item in Status Menu output mode.

−L *[imagePath]*
For Status Menu output mode only. Set the icon image for the status item in Status Menu output mode. This must be a 16x16 pixel image.

−d Development mode. A symlink to the original script is created inside the application bundle instead of an actual copy being placed therein. Symlinks are also created to bundled files. This option is incompatible with the -S option.

−l Optimize application. Strips and compiles the bundled nib file to reduce application size, making the nib uneditable. This option only works if you have Apple’s Developer Tools installed, since it uses the program /Developer/usr/bin/ibtool.

−y Force mode. With this flag set, the program will happily overwrite any previous files and folders in destination path. Use with caution.

−H *[nibPath]*
Specify an alternate nib to copy to application bundle. This allows you to integrate a customised Platypus application nib into your build process.

−v Prints the version of the **platypus** command line utility

-h Prints help and usage string

The **platypus** utility exits 0 on success, and >0 if an error occurs.

EXAMPLES

```
platypus -P myProfile.platypus myApplication.app

platypus -a MyApp -c script.pl -p /usr/bin/perl -o ’Text Window’ myApplication.app

platypus -D -a MyDroplet -c droplet.sh -o ’Web View’ myApplication.app
```

FILES

`/usr/local/bin/platypus`	program binary
`/usr/local/share/platypus/ScriptExec`	executable binary
`/usr/local/share/platypus/MainMenu.nib`	Nib file for app
`/usr/local/share/platypus/PlatypusDefault.icns`	Default icon

AUTHORS

This manual page was written by Sveinbjorn Thordarson. To support Platypus development, please visit http://sveinbjorn.org/donations.