

osascript man page

This manual page is for Mac OS X Yosemite (version 10.10.5). You can view the documentation available on your local system using the `man` command in a *Terminal* window.

OSASCRIP(1) BSD General Commands Manual OSASCRIP(1)

NAME

osascript — execute OSA scripts (AppleScript, JavaScript, etc.)

SYNOPSIS

```
osascript [-l language] [-i] [-s flags]  
[-e statement | programfile] [argument ...]
```

DESCRIPTION

osascript executes the given OSA script, which may be plain text or a compiled script (.sct) created by Script Editor or `osacompile(1)`. By default, **osascript** treats plain text as AppleScript, but you can change this using the `-l` option. To get a list of the OSA languages installed on your system, use `osalang(1)`.

osascript will look for the script in one of the following three places:

1. Specified line by line using `-e` switches on the command line.
2. Contained in the file specified by the first filename on the command line. This file may be plain text or a compiled script.
3. Passed in using standard input. This works only if there are no filename arguments; to pass arguments to a STDIN-read script, you must explicitly specify “-” for the script name.

Any arguments following the script will be passed as a list of strings to the direct parameter of the “run” handler. For example, in AppleScript:

```
a.sct:
```

```
on run argv
return "hello, " & item 1 of argv & "."
end run
```

```
% osascript a.sct world
hello, world.
```

The options are as follows:

-e *statement*

Enter one line of a script. If **-e** is given, **osascript** will not look for a filename in the argument list. Multiple **-e** options may be given to build up a multi-line script. Because most scripts use characters that are special to many shell programs (for example, AppleScript uses single and double quote marks, “(”, “)”, and “*”), the statement will have to be correctly quoted and escaped to get it past the shell intact.

-i

Interactive mode: **osascript** will prompt for one line at a time, and print the result, if applicable, after each line. Any script supplied as a command argument using **-e** or *programfile* will be loaded, but not executed, before starting the interactive prompt.

-l *language*

Override the language for any plain text files. Normally, plain text files are compiled as AppleScript.

-s *flags*

Modify the output style. The *flags* argument is a string consisting of any of the modifier characters **e**, **h**, **o**, and **s**. Multiple modifiers can be concatenated in the same string, and multiple **-s** options can be specified. The modifiers come in exclusive pairs; if conflicting modifiers are specified, the last one takes precedence. The meanings of the modifier characters are as follows:

h

Print values in human-readable form (default).

s

Print values in recompilable source form.

osascript normally prints its results in human-readable form: strings do not have quotes around them, characters are not escaped, braces for lists and records are omitted, etc. This is generally more useful, but can introduce ambiguities. For example, the lists ‘{"foo", "bar"}’ and ‘{{"foo", {"bar"}}}’ would both be displayed as ‘foo, bar’. To see the results in an unambiguous form that could be recompiled into the same value, use the **s** modifier.

e

Print script errors to stderr (default).

o

Print script errors to stdout.

osascript normally prints script errors to stderr, so downstream clients only see valid results. When running automated tests, however, using the **o** modifier lets you distinguish script errors, which you care about matching, from other diagnostic output, which you don't.

SEE ALSO

osacompile(1), osalang(1), *AppleScript Language Guide*

HISTORY

osascript in Mac OS X 10.0 would translate ‘\r’ characters in the output to ‘\n’ and provided **c** and **r** modifiers for the **-s** option to change this. **osascript** now always leaves the output alone; pipe through `tr(1)` if necessary.

Prior to Mac OS X 10.4, **osascript** did not allow passing arguments to the script.

Mac OS X April 24, 2014 Mac OS X
